



Universidad de Jaén
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

Desarrollo de una Aplicación Basada en Visión por Computador para Asistencia a Invidentes

Alumno: Nombre Alumno

Tutor: Prof. D^a. Silvia María Satorres Martínez

Dpto: Ingeniería Electrónica y Automática

Junio, 2021



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Ingeniería Electrónica y Automática

D^a SILVIA MARÍA SATORRES MARTÍNEZ , tutor del Trabajo Fin de Grado titulado: DESARROLLO DE UNA APLICACIÓN BASADA EN VISIÓN POR COMPUTADOR PARA ASISTENCIA A INVIDENTES, que presenta ADRIÁN ORTIZ MARTÍNEZ, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, JUNIO de 2021

El alumno:

Adrián Ortiz Martínez

ADRIÁN ORTIZ MARTÍNEZ

Los tutores:

SILVIA MARÍA
SATORRES MARTÍNEZ

A mi padre, impulsor de esta idea de proyecto, y a mi madre. Por su confianza ciega en mí.

RESUMEN

En este proyecto se podrá visualizar el desarrollo de una aplicación orientada para ayudar a las personas invidentes en su día a día. Dicha aplicación estará basada en la programación de un ordenador de placa reducida con la finalidad de que pueda detectar y hacer un seguimiento de marcas fiduciales en el entorno, informando en todo momento al usuario de la situación de estas marcas. En cuanto el software, la programación se hará con Python utilizando las librerías que ofrece OpenCV.

ABSTRACT

This project will visualise the development of an application aimed at helping blind people in their daily lives. This application will be based on the programming of a computer with a reduced board so that it can detect and monitor the fiducial marks in the environment, informing the user of the situation of these marks at all times. As for the software, the programming will be done with Python using the libraries offered by OpenCV.

AGRADECIMIENTOS

Quiero hacer llegar mi más sentido agradecimiento a todas las personas que han hecho posible que yo haya podido llegar hasta aquí. Desde mi familia, mis amigos, mi pareja, hasta todos los profesores que me han prestado su ayuda en algún momento. Nunca imaginé que podría emprender un camino como este y eso me transmite un sentimiento de agradecimiento y motivación para seguir con lo que me proponga en mi futuro.

Adrián Ortiz Martínez

Contenido

RESUMEN	ii
ABSTRACT	iii
AGRADECIMIENTOS	iv
1 - INTRODUCCIÓN	1
1.2 - Motivación	1
1.3 - Objetivos.....	2
1.4 - Estructura de la memoria	3
2 - ESTADO DE LA TECNOLOGÍA Y DEL ARTE	5
2.1 - Sistemas de guía para personas invidentes.....	6
2.1.1 OrCam MyEye.....	6
2.1.2 - weWalk	7
2.1.3 - Sunu Band	7
2.1.4 - Perro guía robótico.....	8
2.1.5 - Sistema de guiado en desarrollo por Massachusetts of Technology	9
2.2 - Procesamiento de marcas fiduciales.....	10
2.2.1 - Marcadores fiduciales y realidad aumentada.....	10
2.2.1 - Sistema de navegación para Pepper Robot	11
3 - DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO.....	12
3.1 - Diseño hardware.....	12
3.2 - Construcción del prototipo	20
4 - SOFTWARE DEL PROTOTIPO	21
4.1 - Módulo ArUco.....	22
4.1.1 - Creación de un marcador ArUco	22
4.1.2 - Detectar un marcador ArUco.....	26
4.1.3 - Estimar posición de un marcador ArUco	28
4.2 - Diseño y programación de una aplicación.....	30
4.2.1 - Calibración de la cámara.....	30
4.2.2 - Interfaz de usuario.....	32
4.2.3 - Detección, clasificación y estimación de posición cámara-marcaador	35
5 - RESULTADOS.....	39
5.1 - Distancia.....	39
5.1.2 - Comparación de medida de distancia entre marcadores de distinto tamaño	41
5.2 - Modo Día y Modo Noche	49

6 - PRESUPUESTO	65
7 - CONCLUSIONES Y TRABAJOS FUTUROS.....	66
8 - BIBLIOGRAFÍA	67

ÍNDICE DE FIGURAS

Figura 1. Gafa OrCam MyEye	6
Figura 2. Bastón weWALK	7
Figura 3. Pulseras Sunu Band.....	8
Figura 4. Prototipo perro guía.....	8
Figura 5. Sistema de guiado para invidentes.....	9
Figura 6. Marcadores fiduciales y realidad aumentada.....	10
Figura 7. Robot Pepper en búsqueda de marcadores	11
Figura 8. Robot Pepper dirigiéndose al marcador de la derecha	11
Figura 9. Tarjeta Raspberry Pi 4 Modelo B 4 GB.....	12
Figura 10. Disipador de calor.....	13
Figura 11. Micro SD Sandisk Extreme.....	13
Figura 12. Pi Camera HQ.....	14
Figura 13. Ejemplo de distintas distancias focales según el tamaño de la lente	14
Figura 14. Lente 6 mm usada para el prototipo	15
Figura 15. Fuente de alimentación externa.....	15
Figura 16. Diseño de la carcasa en AutoCAD	16
Figura 17. Carcasa de tela para el prototipo.....	17
Figura 18. Soporte cámara y lente	18
Figura 19. Prototipo con vistas al soporte.....	19
Figura 20. Prototipo con la carcasa de tela.....	19
Figura 21. Ejemplos de marcadores ArUco	22
Figura 22. Diagrama de bloques creación de marcador	24
Figura 23. Marcador de la librería DICT_5X5_50	25
Figura 24. Rumbo a seguir para detección de un marcador	26
Figura 25. Diagrama de bloques detección de marcador.....	27
Figura 26. Visualización de los marcadores con los ejes de coordenadas creados por la estimación de posición	28
Figura 27. Diagrama de bloques estimación de posición de marcadores	29
Figura 28. Banco de imágenes para la calibración de la cámara	31
Figura 29. Diagrama de bloques para calibración de una cámara.....	32
Figura 30. GPIO RPi 4B	32
Figura 31. Diagrama de bloques para los <i>callbacks</i>	34
Figura 32. Visualización del vector de traslación	35
Figura 33. Diagrama de bloques aplicación.....	38
Figura 34. Prueba a 60 cm	40
Figura 35. Prueba a 50 cm	40
Figura 36. Prueba a 40 cm	41
Figura 37. Ensayo a 2 m con marcador 1500x1500 Mp.....	42
Figura 38. Ensayo a 1.8 m con marcador 1500x1500 Mp.....	43
Figura 39. Ensayo a 1.5 m con marcador 1500x1500 Mp.....	44
Figura 40. Ensayo a 1.8 m con marcador 1500x1500 Mp.....	45
Figura 41. Comportamiento de la estimación de distancia.....	46
Figura 42. Ensayo a 2 m con marcador 900x900 Mp	47
Figura 43. Ensayo a 1.8 m con marcador 900x900 Mp	47
Figura 44. Ensayo a 1 m con marcador 900x900 Mp	48
Figura 45. Ensayo a 1.5 m con marcador 900x900 Mp	48
Figura 46. Identificación de un marcador a la izquierda del usuario a la luz del día usando el Modo Día.....	49

ÍNDICE DE FIGURAS

Figura 47. Identificación de un marcador a la derecha del usuario a la luz del día usando el Modo Día.....	50
Figura 48. Identificación de un marcador a la derecha del usuario a la luz del día usando el Modo Día.....	50
Figura 49. Identificación de un marcador a la izquierda del usuario usando el Modo Noche	51
Figura 50. Identificación de un marcador a la izquierda del usuario usando el Modo Día.....	52
Figura 51. Identificación de un marcador a la izquierda del usuario usando el Modo Noche	52
Figura 52. Intento de identificación de un marcador a la izquierda del usuario usando el Modo Día.....	53
Figura 53. Ensayo a 2 m test baja luminosidad Modo Día	53
Figura 54. Ensayo a 1.8 m test baja luminosidad Modo Día	54
Figura 55. Ensayo a 1.5 m test baja luminosidad Modo Día	56
Figura 56. Ensayo a 1 m test baja luminosidad Modo Día	57
Figura 57. Ensayo a 2 m test baja luminosidad Modo Noche	58
Figura 58. Ensayo a 1.8 m test baja luminosidad Modo Noche	60
Figura 59. Ensayo a 1.5 m test baja luminosidad Modo Noche	61
Figura 60. Ensayo a 1 m test baja luminosidad Modo Noche	62
Figura 61. Comportamiento de la estimación de la distancia del prototipo frente a la distancia referencia en Modo Día.....	63
Figura 62. Comportamiento de la estimación de la distancia del prototipo frente a la distancia referencia en Modo Noche	63

ÍNDICE DE TABLAS

Tabla 1. Recolecta de datos para 60 cm y cálculo de valor para 2 m	37
Tabla 2. Recolección de datos y cálculo del error cometido para 2 m y marcador 1500x1500 Mp.....	42
Tabla 3. Recolección de datos y cálculo del error cometido para 1.8 m y marcador 1500x1500 Mp	43
Tabla 4. Recolección de datos y cálculo del error cometido para 1.5 m y marcador 1500x1500 Mp	44
Tabla 5. Recolección de datos y cálculo del error cometido para 1 m y marcador 1500x1500 Mp.....	45
Tabla 6. Recolección de datos y cálculo del error cometido para 2 m test baja luminosidad Modo Día.....	54
Tabla 7. Recolección de datos y cálculo del error cometido para 1.8 m test baja luminosidad Modo Día.....	55
Tabla 8. Recolección de datos y cálculo del error cometido para 1.5 m test baja luminosidad Modo Día.....	56
Tabla 9. Recolección de datos y cálculo del error cometido para 1 m test baja luminosidad Modo Día.....	57
Tabla 10. Recolección de datos y cálculo del error cometido para 2 m test baja luminosidad Modo Noche.....	59
Tabla 11. Recolección de datos y cálculo del error cometido para 1.8 m test baja luminosidad Modo Noche	60
Tabla 12. Recolección de datos y cálculo del error cometido para 1.5 m test baja luminosidad Modo Noche	61
Tabla 13. Recolección de datos y cálculo del error cometido para 1 m test baja luminosidad Modo Noche.....	62
Tabla 14. Presupuestado del proyecto	65

1 - INTRODUCCIÓN

Es evidente que el avance de la electrónica y la comunicación está siendo muy rápido y a su vez intrusivo en nuestro día a día. Por ello, las investigaciones para realizar labores cotidianas multiplican sus esfuerzos para poder llegar cuanto antes. De ahí que, la vertiente de los entornos inteligentes, cada vez sea una realidad más certera.

Este proyecto busca adentrarse dentro de la vertiente de los entornos inteligentes para poder cubrir una necesidad, que, por desgracia, demandan las personas invidentes. Esta necesidad está, en poder reconocer puntos de interés a través de un robot programado, de tal manera, que este sea capaz de comunicarse con eficacia con su usuario.

Llevar a cabo esta tarea tendrá distintos puntos de discusión tales como: la localización del robot para que pueda tener una visión correcta del entorno, el hardware a utilizar, el desarrollo del código, el presupuesto, la recolecta de información de otros proyectos elaborados por otros investigadores, o la forma de realizar ensayos, entre otros.

1.2 - Motivación

La motivación de este proyecto reside en demostrar mi aprendizaje a lo largo de mi carrera académica, junto con la búsqueda de poder aportar a la sociedad que me rodea. Actualmente, no es un campo muy investigado, en el que existan múltiples patentes o proyectos que ya están en funcionamiento, de ahí que mi interés fuese aún mayor.

Para conseguir un funcionamiento correcto de estas tecnologías se necesitan una gran cantidad de experimentos y pruebas, algo que requiere tiempo y dinero en investigación, puesto que debe asegurarse en todo momento el buen funcionamiento, ya que la orientación de un invidente está en juego. De ahí que la mayoría de los proyectos que si se comercializan tengan un precio muy elevado, lejos de las capacidades económicas del ciudadano medio.

INTRODUCCIÓN

Por lo comentado, a mi parecer, resulta interesante el llevar a cabo un proyecto que pueda informar de la posición de ciertos puntos de interés para una persona invidente y que además pueda ser asequible para un presupuesto bajo

1.3 - Objetivos

Los principales objetivos a tener en cuenta para poder llevar a cabo con éxito el presente proyecto, siguiendo una actuación ordenada y minuciosa, serán:

- Conocer el entorno en el que se va a desarrollar el prototipo a través de la búsqueda de información que hagan conseguir al autor los conocimientos necesarios para realizar este proyecto.
- Adquisición del hardware necesario para poder programar la aplicación que se desarrolle.
- Testear los distintos códigos que se implementen para conseguir un funcionamiento óptimo.

Por orden, conocer el entorno en el que se va a desarrollar el prototipo, es el punto más importante. Aquí, nos podemos encontrar dos tipos de actuación. Una de ellas sería la creación de una *machine learning* que sea capaz de detectar elementos en el entorno. Y otro punto de actuación para esta labor, sería el que se va a implementar en este proyecto, trata sobre la detección de unos marcadores binarios que representarán un punto de interés. Este último es más fácil de implementar que el primero y a su vez, requiere menos información de objetos externos. Esto no quiere decir que sea un método de actuar ni peor ni mejor, cualquier método puede adaptarse perfectamente a la elaboración de este proyecto.

La adquisición del hardware necesario puede ser un verdadero quebradero de cabeza, y más si resulta ser un proyecto de bajo presupuesto. Por decirlo de alguna manera, es necesario un estudio de mercado para poder conocer los productos existentes que sean capaces de ofrecer una buena experiencia, a lo que calidad-precio se refiere.

El último objetivo será el de poner a prueba, tanto el desarrollo del entorno que se va a programar, como el hardware usado, a base de pruebas y experimentos que

INTRODUCCIÓN

verifiquen un correcto funcionamiento o la existencia de fallas que necesiten ser solucionadas.

1.4 - Estructura de la memoria

Lo que resta de memoria estará estructurada de la siguiente forma. Tendrá siete capítulos sin contar el contar el capítulo introductorio. En primer lugar, se posiciona *ESTADO DE LA TECNOLOGÍA Y DEL ARTE* en el que se hará un breve resumen sobre el estado actual de la tecnología y cómo está esta aplicada en las personas invidentes. Posee un subapartado, el cual ofrecerá diversos tipos de *gadgets* y las características principales de estos, y otro subapartado que dictará sobre el procesamiento de las marcas fiduciales en nuestro día a día.

Tras la descripción del capítulo dos, se expondrá la actuación seguida para poder lograr un hardware correcto y la construcción del prototipo final, en el capítulo llamado *DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO*.

En el capítulo cuatro, *SOFTWARE DEL PROTOTIPO*, se ubicará la explicación del procesamiento de las marcas fiduciales utilizadas y el desarrollo del software del prototipo.

Una vez explicado el algoritmo, se visualizarán los resultados obtenidos en los distintos ensayos en el capítulo llamado, *RESULTADOS*.

En el capítulo número seis, *PRESUPUESTO*, se pondrá a disposición del lector el presupuesto empleado para que se haga realidad este proyecto de bajo presupuesto.

Finalizando con la elaboración del documento, en el capítulo siete, *CONCLUSIONES Y TRABAJOS FUTUROS*, se verá una conclusión por parte del autor tras haber realizado el presente proyecto, además de unas posibles valoraciones para poder mejorar el proyecto con ideas más ambiciosas.

INTRODUCCIÓN

Por último, en el capítulo apodado, *BIBLIOGRAFÍA*, habrá unas líneas dedicadas para la bibliografía empleada, que ha servido al autor del proyecto como punto de apoyo.

2 - ESTADO DE LA TECNOLOGÍA Y DEL ARTE

Como se ha mencionado anteriormente, hay una vertiente que persigue ayudar de una manera correcta al ciudadano invidente o con discapacidad visual. Son bien conocidas las distintas ayudas que posee un invidente para intentar llevar una vida normal, tales como el bastón, el perro guía o señalizaciones en braille en el entorno. Pero vivimos en un mundo en continuo avance y la necesidad de avanzar y mejorar siempre está ahí, por lo que, ¿por qué no aplicar diversas tecnologías para mejorar estas ayudas que ya posee, e incluso desbancarlas?

Desde lo más “sencillo”, hay bastones que poseen su propio sistema GPS o pulseras o cinturones con sensores de proximidad que detectan, hasta lo más innovador, ojos artificiales, entrenados mediante un determinado lenguaje, que realizan la labor de un ojo humano, se intenta incorporar como ayuda a las personas invidentes.

También, existe una investigación en desarrollo, que trata de simular un perro guía a través de un giroscopio, cuanto menos curioso. Este último es un claro ejemplo del estado actual de la electrónica, la cual, tiene en su mano un inmenso abanico de posibilidades a la hora de simular ciertos tipos de actividades.

2.1 - Sistemas de guía para personas invidentes

En mi búsqueda de información para encaminar mi proyecto, he encontrado diversos *gadgets* interesantes, de los cuales, pocos de ellos se comercializan actualmente, puesto que siguen en desarrollo para poder ser un sistema de guía asistida eficiente y seguro, como se ha mencionado anteriormente en los principales objetivos para el desarrollo de este prototipo.

A continuación, hablo sobre los proyectos que he podido encontrar y que generen en mí curiosidad y admiración. Todos ellos son una gran herramienta para satisfacer algunas necesidades para el usuario invidente con un gran trabajo detrás.

2.1.1 OrCam MyEye

Se trata de un *gadget* basado en una cámara inteligente y una gafa. La empresa lo cataloga como un objeto cómodo, con un peso de unos 22 gramos, y lo suficientemente potente para poder actuar como un tercer ojo. En cuanto a su desempeño, es capaz de percibir señales, el código de barras de unos 70.000 productos, el rostro de 100 personas, colores, sexo y el valor del dinero.

A través de un altavoz, que solo es oído por el sujeto que realiza el uso del aparato, se transmite lo que la cámara detecta, eligiéndose el idioma que requiera el usuario, ya que posee información de hasta 15 idiomas distintos.



Figura 1. Gafa OrCam MyEye

ESTADO DE LA TECNOLOGÍA Y EL ARTE - SISTEMAS DE GUÍA PARA PERSONAS INVIDENTES

2.1.2 - weWalk

weWALK da nombre a un bastón inteligente, desarrollado por Kursat Ceylan, que consta de la tecnología típica de un smartphone. En el mango del bastón se ubica un altavoz, un micrófono y un sensor háptico. Este último, envía vibraciones al usuario como si fuese una notificación. También cuenta con conexión Bluetooth que le permite conectarse a los smartphones para poder hacer uso, por ejemplo, de Google Maps y poder encontrar tiendas próximas.

weWALK utiliza un sensor ultrasónico que detecta objetos a un nivel superior de la altura del pecho, avisando con una vibración en el mango cuando haya un objeto próximo que no haya sido detectado con el bastón.



Figura 2. Bastón weWALK

2.1.3 - Sunu Band

Vi Como se puede observar en el nombre del artilugio y en la imagen, esta vez, se trata de una pulsera que se utiliza como un extra a la asistencia de una persona vidente junto al típico bastón. Se vale de un sensor de proximidad que detecta objetos desde la altura de las rodillas hasta la altura de la cabeza en un rango de hasta 4.5 metros en adelante.

ESTADO DE LA TECNOLOGÍA Y EL ARTE - SISTEMAS DE GUÍA PARA PERSONAS INVIDENTES



Figura 3. Pulseras Sunu Band

2.1.4 - Perro guía robótico

Es un prototipo que ha desarrollado un estudiante para las personas invidentes que quieran las personas de un perro guía, pero no puedan lidiar con la mascota por cualquier tipo de problema. Esta mano robótica consta de un giroscopio que sirve para mover la mano del usuario y guiarlo. Otro dato destacable es que está conectado en línea para poder advertir de la situación climatológica o sobre el tráfico. Sin duda es el sistema de guía más interesante e innovador que existen en la actualidad.



Figura 4. Prototipo perro guía

ESTADO DE LA TECNOLOGÍA Y EL ARTE - SISTEMAS DE GUÍA PARA PERSONAS INVIDENTES

2.1.5 - Sistema de guiado en desarrollo por Massachusetts of Technology

Este sistema está definido por una cámara 3D y un pequeño PC colocados en el cuello del usuario, un cinturón en el cual se encuentran dispuestos 5 motores vibradores situados en la mitad frontal y una línea braille colocada en el costado del usuario.

El funcionamiento de este sistema se basa en la detección de objetos de la cámara y la señalización de estos mediante los motores de vibración con una determinada frecuencia según la distancia a estos objetos. Otra utilidad de este proyecto, es la detección, si se desea, de un asiento libre para el usuario.

La línea braille sirve como complemento de los motores de vibración, indicando el tipo de objeto que se va a encontrar y la orientación.

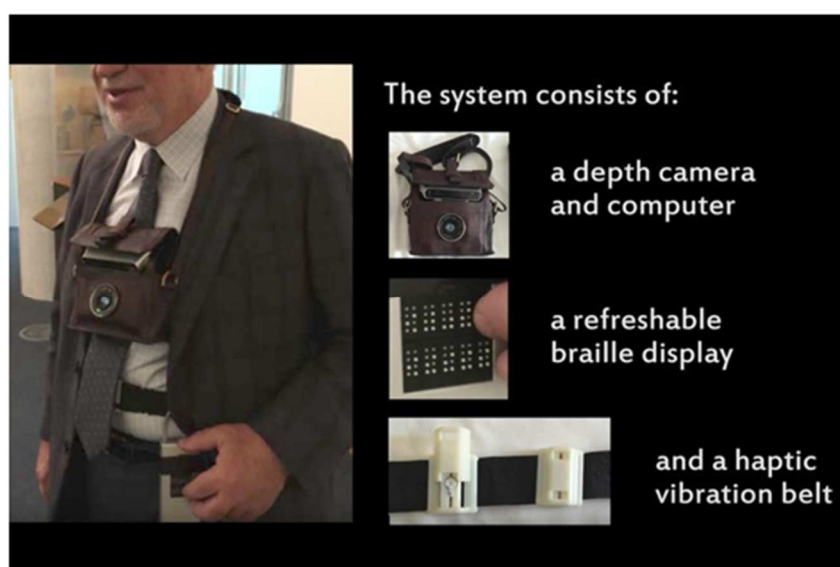


Figura 5. Sistema de guiado para invidentes

Este último, como se podrá ver más adelante, ha servido para encaminar el diseño hardware del prototipo

2.2 - Procesamiento de marcas fiduciales

La estimación de posicionamiento es de gran importancia en muchas aplicaciones de visión por computador. Este proceso está basado en buscar correspondencias entre puntos de la vida real y sus proyecciones en imágenes 2D. Este paso es normalmente difícil, por lo que es común usar marcas sintéticas o fiduciales y correcciones técnicas. Las marcas de referencia son usadas en diversos campos, como pueden ser, la física, el estudio geocientífico, la realidad virtual, imágenes médicas, biología celular o en las PCB (*Printed Circuit Board*).

El más popular, es el uso de marcas fiduciales con forma de cuadrado binario. El principal beneficio de estas marcas es que te otorgan suficientes correspondencias para obtener la pose de cámara. También la cosificación interna las hace especialmente robusta, permitiendo la posibilidad de detectar errores y corregirlos.

Al igual que se realizó una búsqueda orientativa sobre distintos tipos de tecnología orientada a guiar a personas invidentes, se realiza otra búsqueda para el procesamiento de marcas fiduciales en diversos tipos de aplicaciones.

2.2.1 - Marcadores fiduciales y realidad aumentada

Mediante el foro *Learn OpenCV*, en el cual, cientos de investigadores comparten sus investigaciones, la ingeniera Sunita Nayak, muestra su trabajo que se basa en cambiar la imagen existente en un marco utilizando marcadores fiduciales. Estos estarán dispuestos en las esquinas del marco y al correr un determinado algoritmo, se cargará una imagen escogida para superponerla al contenido del cuadro tal y como se puede observar en las siguientes figuras.



Figura 6. Marcadores fiduciales y realidad aumentada

ESTADO DE LA TECNOLOGÍA Y EL ARTE - PROCESAMIENTO DE MARCAS FIDUCIALES

2.2.1 - Sistema de navegación para Pepper Robot

A través de la plataforma de código abierto, GitHub, los contribuyentes Rémi Humbert y Emile Kroeger, publican el algoritmo que han utilizado para que el robot Pepper Robot se oriente en un entorno gracias a unos marcadores ArUco posicionados estratégicamente en una tienda.

El robot busca los marcadores tal y como se muestra en la figura siete y se dirige al marcador ordenado, como se muestra en la figura ocho.

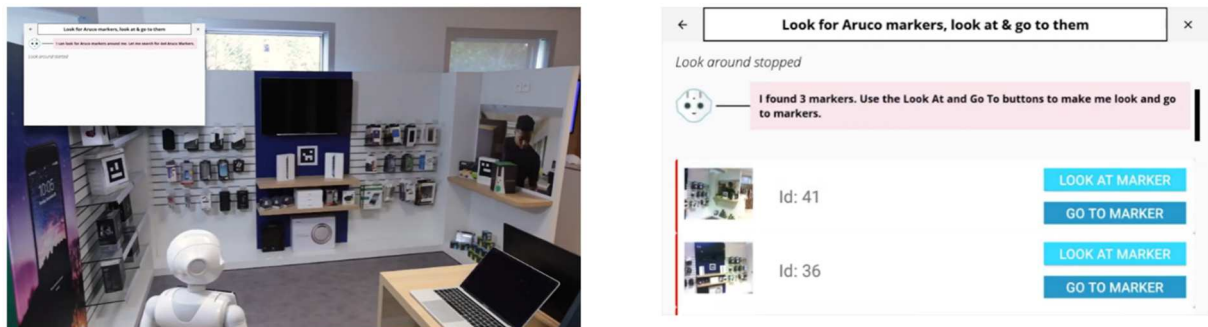


Figura 7. Robot Pepper en búsqueda de marcadores

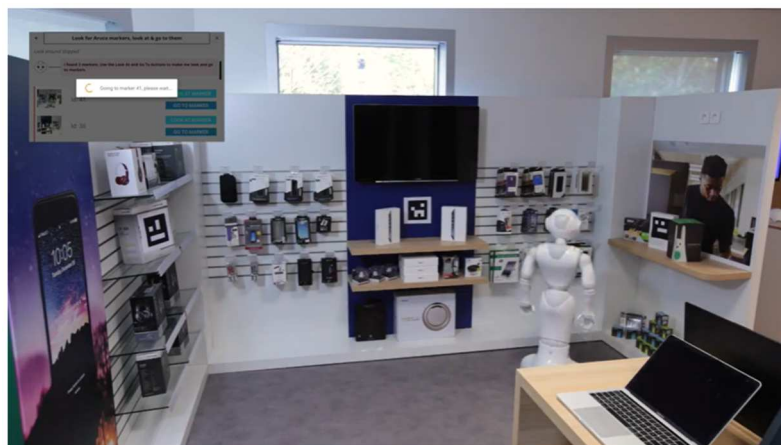


Figura 8. Robot Pepper dirigiéndose al marcador de la derecha

3 - DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO

Para llevar a cabo un prototipo electrónico, como en este caso, acto seguido de saber el entorno software que se va a utilizar, se debe de buscar el hardware que lo soporte, tal y como se comentó en los principales objetivos de este trabajo.

El hardware que se mostrará a continuación, actualmente, está constituido por elementos del mayor bajo coste que se ha podido utilizar, sin perder nunca el factor calidad y potencia de vista. Además, se debe de tener en cuenta que cada uno de ellos deben de encajar perfectamente, dentro de lo que es el prototipo.

3.1 - Diseño hardware

El prototipo se basará en la utilización de una placa con la suficiente potencia como para poder ser programada con la intención de detectar marcas fiduciales en el entorno, procesar la información que recibe de estas y poder orientar a un invidente con eficacia. Tras la búsqueda de posibles placas que puedan cumplir con estos objetivos, finalmente se escoge la Raspberry Pi 4 Modelo B 4GB.



Figura 9. Tarjeta Raspberry Pi 4 Modelo B 4 GB

Además de ser una buena opción en cuanto a la relación coste-potencia dentro del mercado de placas “mini ordenadores”, pose otro aliciente, que es la existencia de

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO - DISEÑO HARDWARE

documentación para adentrarse en el entorno de la visión por computador. Por ejemplo, a la hora del desarrollo de este prototipo en cuanto a software, he podido disponer de un libro, que reunía los conocimientos necesarios para poder iniciar este proyecto, llamado *Packt Raspberry Pi Computer Vision Programming 2nd Edition*. La elección de este componente lleva la recomendación de un disipador de calor, por lo que, otro integrante del hardware sería una carcasa de aluminio con doble ventilador.



Figura 10. Disipador de calor

Otro inconveniente de la RPi 4B (Raspberry Pi 4B) es que carece de memoria interna, obligando al autor a realizar una búsqueda en el mercado una tarjeta micro SD (*solid disk*) lo suficientemente rápida y con una capacidad generosa de memoria, teniendo en cuenta que el peso del sistema operativo puede llegar a unos 4 GB y que se trabajará con amplios bancos de fotografías. Finalmente, se escoge la micro SD que ofrece SanDisk *Extreme*, con una capacidad de 64 GB y una velocidad de 160 MB/s.



Figura 11. Micro SD Sandisk Extreme

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO - DISEÑO HARDWARE

En cuanto al “ojo” de este proyecto, se prosigue con los productos que ofrecen Raspberry Pi, y se escoge la cámara más actual que poseen, la Raspberry Pi High Quality Camera. Consta de un sensor Sony IMX477 de 12.3 megapíxeles, una diagonal de imagen de 7.9 mm y un sensor retroiluminado con enfoque posterior ajustable compatible con monturas C y CS.



Figura 12. Pi Camera HQ

Para que sea posible la visión se necesita del montaje de una lente que se adecue a la finalidad del prototipo. Para ello, se necesita saber cual será la distancia focal ideal.

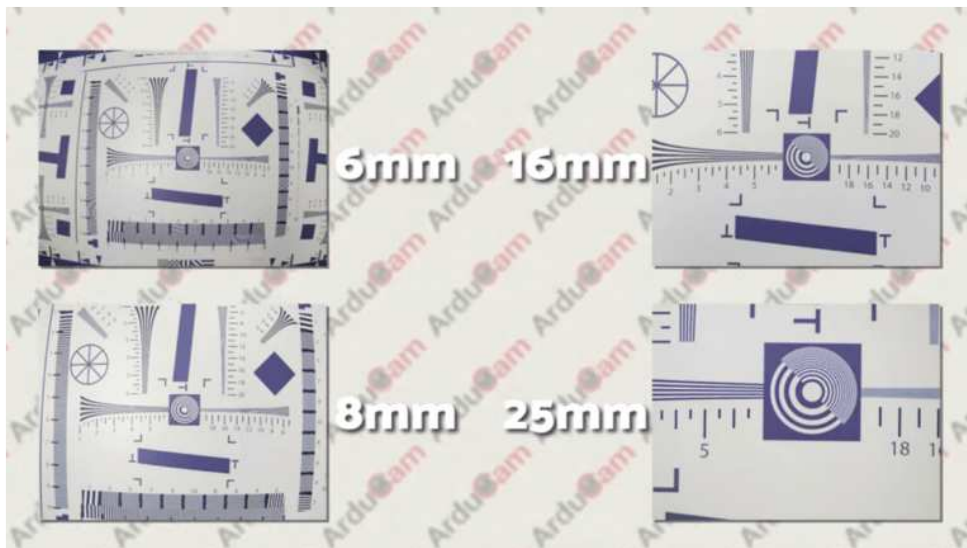


Figura 13. Ejemplo de distintas distancias focales según el tamaño de la lente

Vista la figura 8, inicialmente, se escoge una lente de 25 mm, pero al ponerla en práctica, se llega a la conclusión de que se pierden muchos datos eficaces para la detección de marcas fiduciales. Finalmente se escoge la lente de 6 mm, que será

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO - DISEÑO HARDWARE

eficaz para detectar objetos a distancias cortas (nuestro prototipo pretende detectar las marcas fiduciales a una distancia de unos 2 metros).



Figura 14. Lente 6 mm usada para el prototipo

Por otro lado, este prototipo necesita ser portado por el usuario, así que, necesitará de una alimentación portátil. La RPi 4B necesita una alimentación de 15 W, si se desea un uso óptimo, pero aquí llega la primera problemática. En el mercado actual no se comercializa ninguna batería con esa potencia de alimentación y que cumpla con las dimensiones del prototipo, por lo que se decide utilizar una placa de 10 W originaria para el modelo 3B.

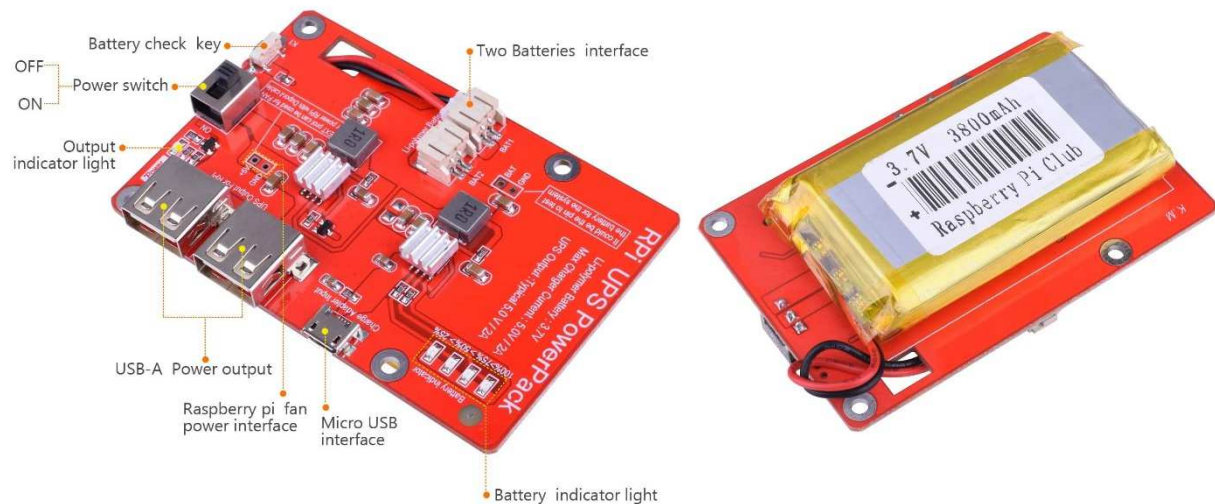


Figura 15. Fuente de alimentación externa

En primera instancia, se pretendía que el prototipo formara parte de un sombrero, pero debido a la robustez de este, la idea se transforma a un objeto colgante del cuello, inspirado en la ilustración cinco. Para ello, es necesario la realización de una

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO - DISEÑO HARDWARE

carcasa que pudiese ser colgada con ayuda del cuello, por lo que, también con ayuda de la figura cinco, se decide realizar una carcasa de tela, usando como guía AutoCAD y las medidas de prototipo.

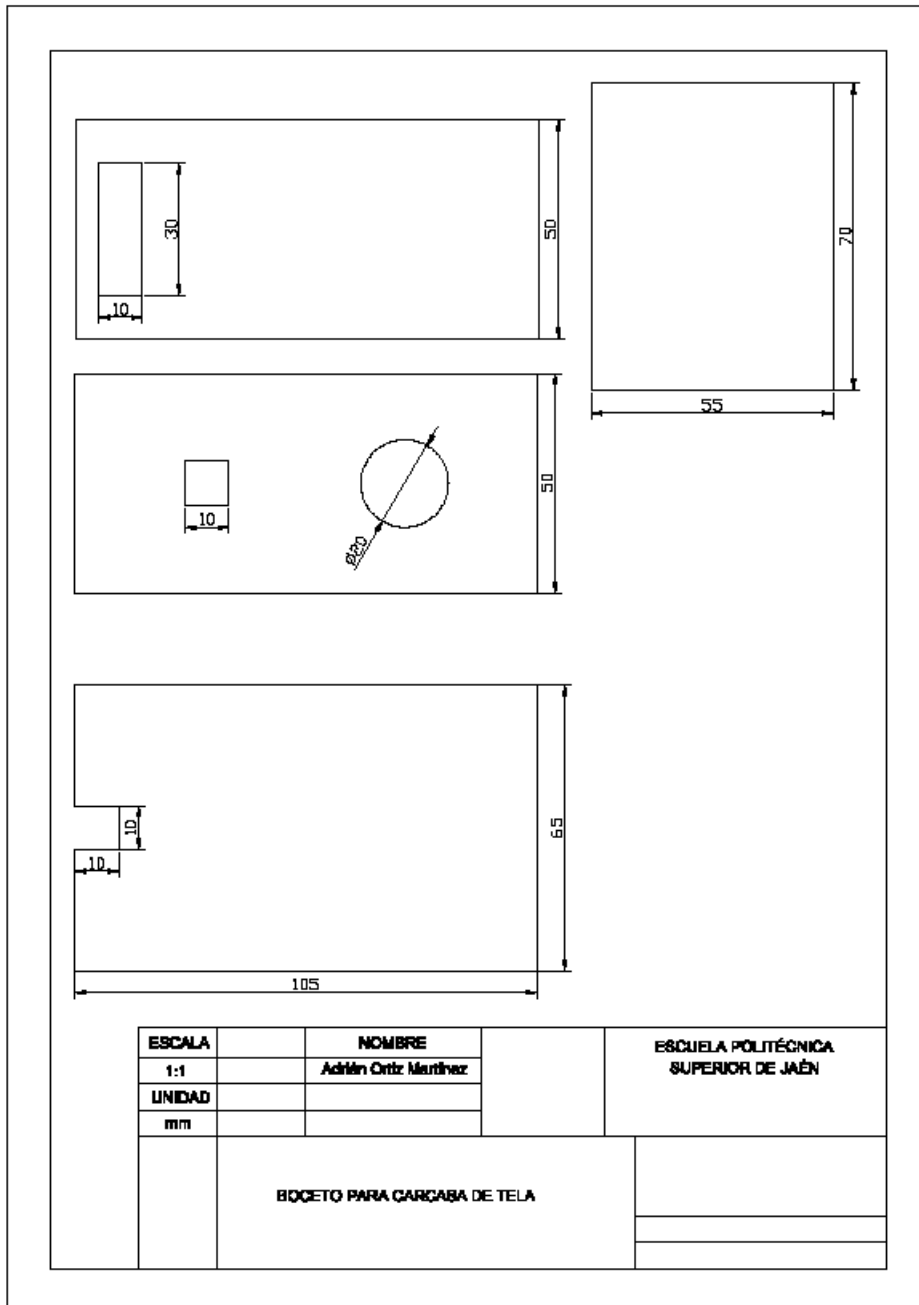


Figura 16. Diseño de la carcasa en AutoCAD

Dando como resultado final, la carcasa que se puede visualizar en la figura 14.



Figura 17. Carcasa de tela para el prototipo

Otro paso a realizar en el diseño es el de elegir la posición de la cámara y la lente. En esta ocasión, los taladros de la alimentación externa que va a tener la RPi, se decide crear dos placas que sirvan como soporte para la lente. Este soporte será fabricado con una chapa de 1.5 mm para que de la suficiente rigidez a la cámara. Como en el caso de la carcasa, se realiza un previo diseño en AutoCAD, tal y como se muestra en la siguiente figura.

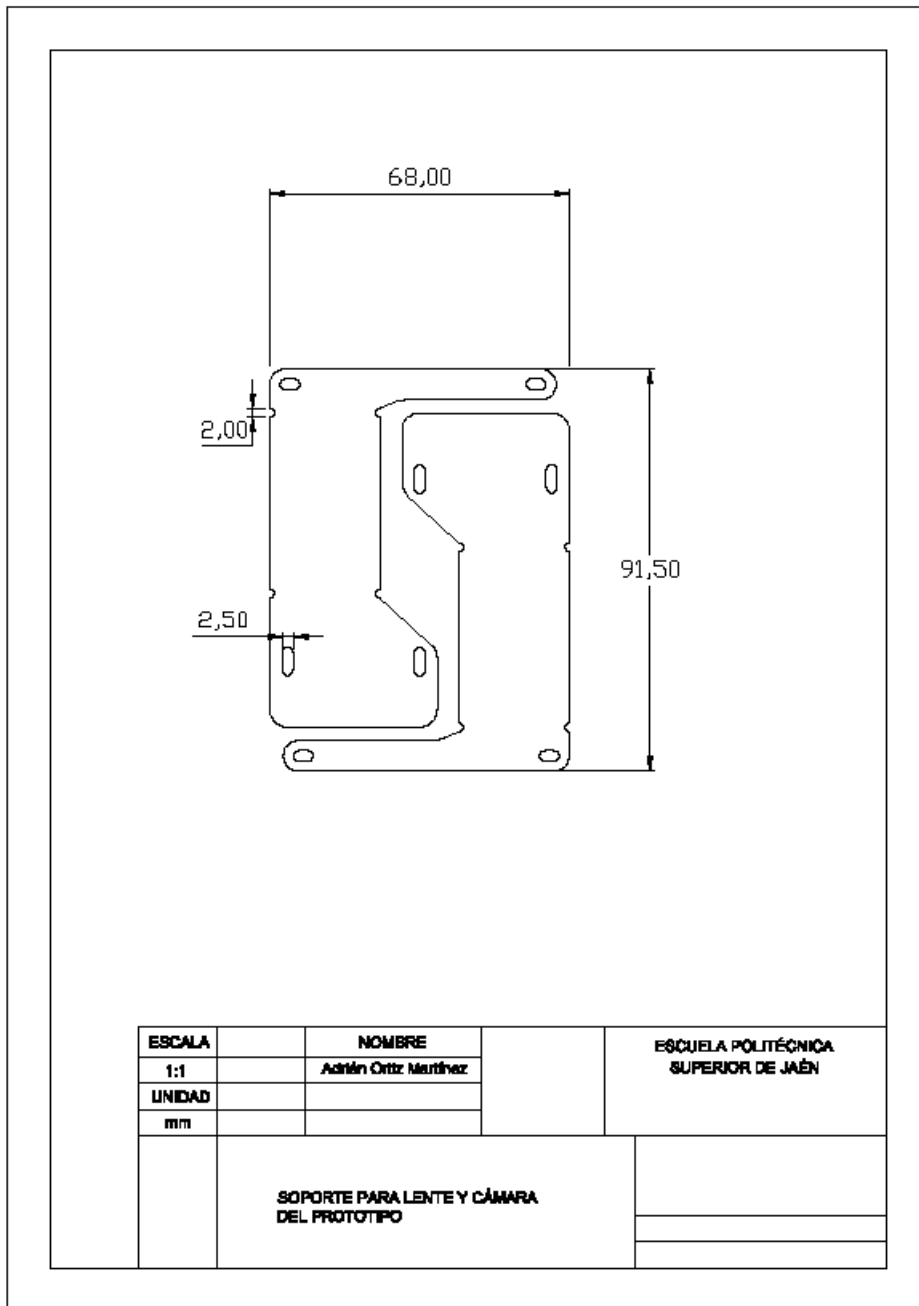


Figura 18. Soporte cámara y lente

Finalmente, el diseño hardware junto con la carcasa de tela se muestra a continuación.

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO - DISEÑO HARDWARE

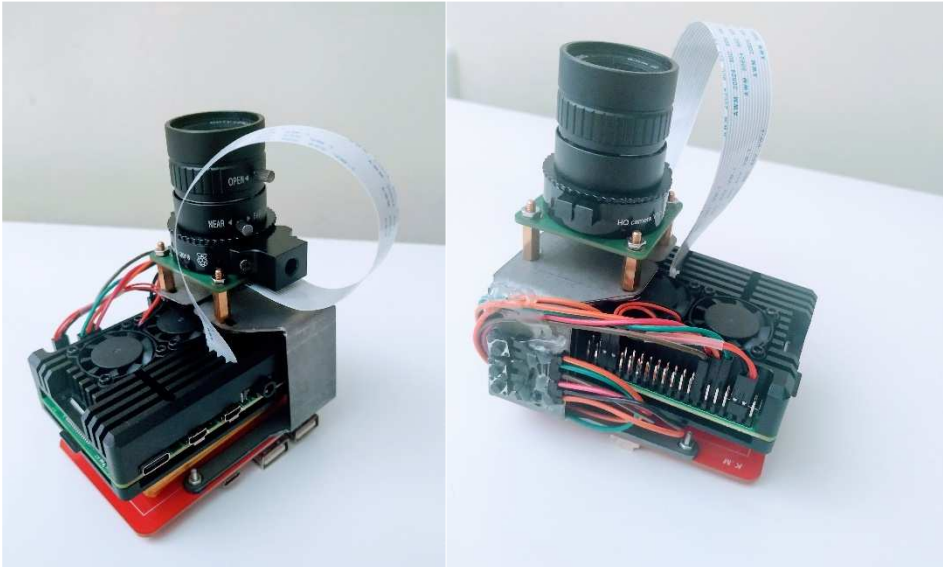


Figura 19. Prototipo con vistas al soporte



Figura 20. Prototipo con la carcasa de tela

El peso final del prototipo es de 432 g, que colgado en el cuello mediante de usuario no resulta molesto.

3.2 - Construcción del prototipo

La combinación entre el hardware y el software, harán que el prototipo sirva como un complemento a utilizar junto a un bastón o perro guía para poder detectar diversos puntos de interés para una persona invidente. Este, estará a la altura del pecho del invidente, gracias a la utilización de la carcasa de tela, y estará en todo momento buscando marcadores fiduciales. Como ya se ha explicado en el apartado 2, cada marcador tendrá un ID único, por lo que cada uno, representará un punto de interés en el camino del invidente.

Para la identificación del marcador, la idea principal es que este actue como una señal junto al punto de interés, es decir, como si se tratase de una señal de tráfico para los conductores. La RPi estará programada para que, cuando se detecte un punto de interés a menos de dos metros, esta avise de al usuario sobre el punto de interés y su localización. En concreto, se avisará si el punto de interés se encuentra a la derecha o a la izquierda del usuario.

Además, el prototipo tendrá una pequeña interfaz de usuario formada por una botonera (colocada con silicona en el soporte que está próximo al GPIO de la RPi) que está programada como *callbacks*. Esto quiere decir que cada vez que se pulse un determinado botón, se podrá modificar ciertos valores dentro del código para poder pausar/activar la detección de marcadores, pasar al modo noche o pasar al modo día.

El primer paso, en cuanto a software se refiere, es decidir el sistema operativo que se va a instalar en la tarjeta. Este proyecto, siguiendo las instrucciones que dicta el libro *Packt Raspberry Pi Computer Vision Programming 2nd Edition*, se instalará *Raspberry Pi OS*. Este sistema operativo se trata de un sistema operativo Linux basado en Debian, de libre acceso, que principalmente está orientado para la enseñanza informática.

Una vez decidido el sistema operativo que tendrá la tarjeta, el siguiente paso es el lenguaje con el que se va a realizar el código del proyecto, que será Python. En este caso se trabajará con la versión Python 3. Este lenguaje, junto a a las librerías de OpenCV, entre ellas, la librería ArUco, permitirá que se lleve a cabo la implementación del prototipo.

4 - SOFTWARE DEL PROTOTIPO

Para poder iniciar este capítulo, primero hay que entender la visión por computador. Esta engloba campos tales, la ciencia del computador, las matemáticas y la ingeniería electrónica. Su misión es la de extraer suficiente información para tomar decisiones, realizando labores como el reconocimiento, clasificación y análisis de objetos a través de un previo procesado de la imagen que permita obtener una información más clara de esta.

Como ya se ha comentado en el apartado, *Construcción del prototipo*, se usará OpenCV (*Open Source Computer Vision*), que es una fuente de documentos para visión por computador y *machine learning*. Posee una larga colección de funciones para *machine learning* y una gran cantidad de GUI (*Graphical User Interface*). OpenCV es gratuito y está escrito en C++. Se puede usar en diversos sistemas operativos y programar con distintos lenguajes, como, por ejemplo, en este caso, que se va a programar con Python 3. La librería posee más de dos mil quinientos algoritmos optimizados para la visión por computador, teniendo una comunidad de más de cuarenta y siete mil profesionales de la visión por computador, con unos dieciocho millones de descargas.

Python empezó como un hobby para Guido van Rossum, un informático holandés, que, sin embargo, con el paso del tiempo, se ha convertido en un lenguaje de alto nivel. Actualmente la programación con Python tiene dos grandes vertientes: Python 2 y Python 3. En su mayoría son incompatibles entre sí, siendo la última versión la que actualmente posee soporte técnico. Este lenguaje requiere la aportación del contenido que poseen diversas librerías, como, por ejemplo, la librería *NumPy*, que, sin ella, no se podría realizar labores relacionados con la computación numérica.

Este proyecto se va a enfocar en los marcadores fiduciales que otorga OpenCV dentro de la librería ArUco. Los marcadores pertenecientes a esta librería son unos marcadores binarios que presentan la forma que se muestra en la figura 18.

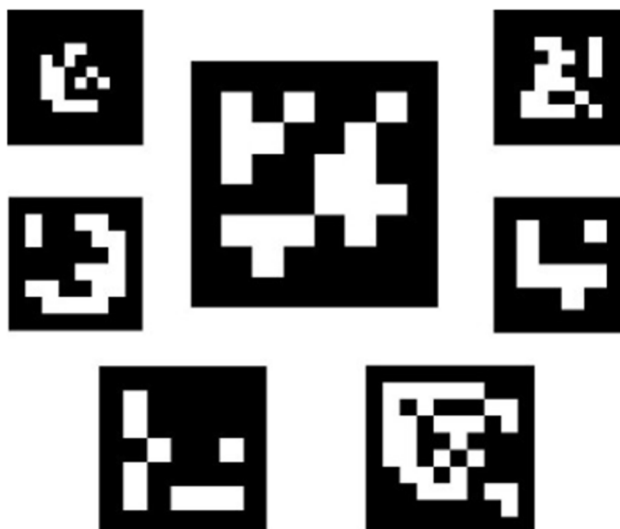


Figura 21. Ejemplos de marcadores ArUco

4.1 - Módulo ArUco

El módulo aruco está basado en la librería ArUco, una librería popular en la detección de marcas fiduciales desarrollada por Rafael Muñoz y Sergio Garrido en el entorno de OpenCV.

Están compuestos por un borde negro y una matriz binaria en su interior, que los define. El borde facilita una rápida detección en la imagen y la codificación binaria permite su detección. El tamaño del marcador lo determina la matriz binaria. Es necesario saber que el marcador lo podemos encontrar rotado en el entorno, pero eso no supone un problema porque cada esquina está identificada inequívocamente. Esto está basado, también, en una codificación binaria.

4.1.1 - Creación de un marcador ArUco

Los marcadores pertenecen a un determinado diccionario, que estará incorporado dentro de la librería ArUco, donde estarán presentes otros marcadores que son usados para un determinado trabajo. Las principales características de estos diccionarios son el tamaño del diccionario y el tamaño del marcador. Estas se pueden explicar brevemente en los siguientes puntos:

- El tamaño del diccionario determinará el número de marcadores que lo componen.

SOFTWARE DEL PROTOTIPO - MÓDULO ARUCO

- El tamaño de los marcadores lo determina el número de bits que posee.

Cada marcador, tiene un ID (Identificación) único, por lo que un posible pensamiento, es que el ID de un marcador puede ser su código binario, pero este es un pensamiento erróneo, puesto que el código binario puede ser un número demasiado grande para tratarse de una ID. El ID de un marcador es simplemente el índice con el que se identifica ese marcador en el diccionario.

Para crear un marcador, podemos escoger dos caminos: manual o automático. Si se quiere realizar un marcador de manera manual, se necesita saber cuál diccionario ArUco se quiere utilizar y extraerlo mediante la función `cv2.aruco.Dictionary_get()`, especificar el ID que se va a asignar al marcador, asignar una memoria de salida para la imagen ArUco (en píxeles) y usar la función `cv2.aruco.drawMarker()` para proceder a dibujar el marcador.

Adentrándonos a los diccionarios ArUco, sirven para especificar el tipo de marcador que se va a generar y detectar, por lo que, sin el diccionario correspondiente, sería imposible el procesamiento de marcador fiducial. Un ejemplo cotidiano podría ser con los idiomas y los diccionarios de lenguajes, en un diccionario inglés podrás encontrar la mayoría de las palabras que necesitas para entender una frase o conversación, pues con los diccionarios ArUco igual, cada uno contiene sus marcadores propios.

En concreto, existen 21 diferentes diccionarios construidos dentro de la librería de OpenCV que, en su mayoría, vienen dado por el formato `cv2.aruco.DICT_NXN_M`. NXN determina el tamaño 2D del bit del marcador ArUco. Por ejemplo, el marcador 5x5, tendrá un total de 25 bits. El valor M determinará el número de IDs ArUco que se pueden generar con el diccionario elegido.

Un ejemplo final podría ser `cv2.aruco.DICT_7X7_250`, que indica que se pueden generar cuadros binarios 7x7 y hasta 250 IDs ArUco distintas.

Una vez sabido cual es el modo de actuación de los diccionarios, llega la decisión, ¿cuál diccionario necesito para mi proyecto? Bien, esto vendrá dado por el valor de IDs que se necesitarán para el proyecto, el tamaño de resolución de la

SOFTWARE DEL PROTOTIPO - MÓDULO ARUCO

información de entrada y la distancia entre marcadores. Posiblemente se pueda resumir en 3 pasos, que se muestran a continuación:

- Número de IDs que necesito. Básicamente, si necesito un número alto de IDs distintas, procederé a la elección de un diccionario que me otorgue dicha premisa. Por ejemplo, cv2.aruco.DICT_7X7_250, que me ofrecerá 250 valores únicos. Será conveniente el diccionario necesario requerido para tus necesidades.
- Tamaño de resolución de información de entrada. Si la cuadrícula de la imagen va a ser grande, más grande tendrá que ser el tamaño del marcador.
- Distancia entre marcadores. La detección de OpenCV puede generar problemas si la distancia entre los marcadores no es la correcta, por lo que se recomienda utilizar diccionarios pequeños con largos tamaños de bit.

El flujo de datos que describe el método manual para crear un marcador, usando las librerías de OpenCV, sería el siguiente:

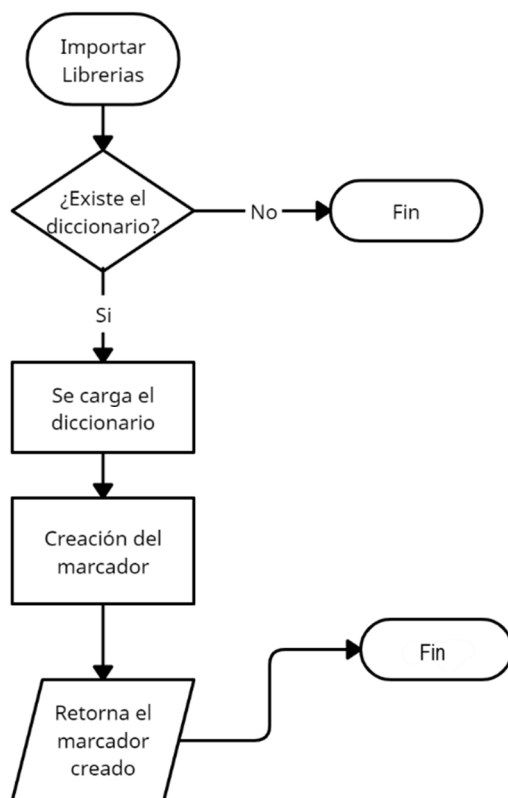


Figura 22. Diagrama de bloques creación de marcador

SOFTWARE DEL PROTOTIPO - MÓDULO ARUCO

En este caso, se ha creado un marcador de tamaño 5x5 con la posibilidad de crear 50 marcadores distintos, por lo que se escoge la librería `DICT_5X5_50`. El .png final vendrá dado por un array de ceros inicial, el cual se pasa como argumento a la función `cv2.aruco.drawMarker()`, función que creará el marcador.

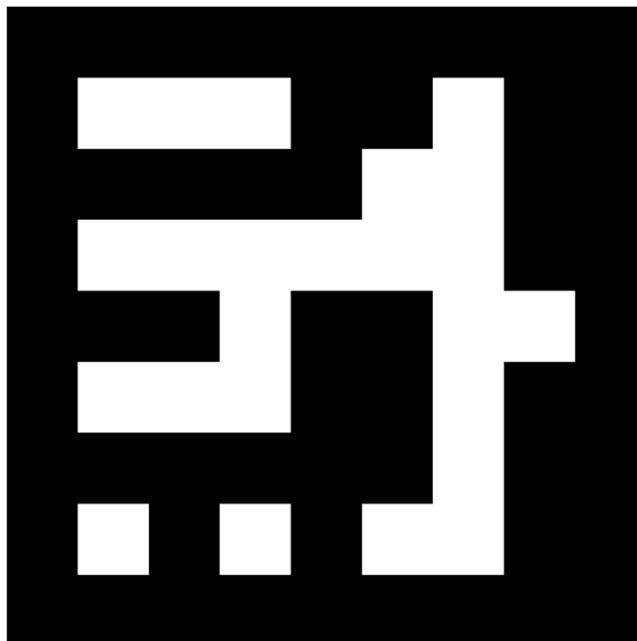


Figura 23. Marcador de la librería `DICT_5X5_50`

Además, a la función `cv2.aruco.drawMarker()`, se le pasa el tamaño (en píxeles) deseado. En el caso de este proyecto, he seleccionado un tamaño de 1500x1500 píxeles, que equivale a unos 0.396875 metros. Este tamaño tras algunas pruebas con el prototipo será el idóneo para poder detectarlo a una determinada función.

Si se decide escoger el camino automático, existen generadores de marcadores ArUco online que pueden facilitar el proceso de dibujar el marcador.

4.1.2 - Detectar un marcador ArUco

A la hora de detectar una marca fiducial, se debe de tener en cuenta los siguientes aspectos:

- Especificar el diccionario ArUco al que pertenecen los marcadores que se quiere detectar.
- Definir los parámetros del marcador.

El orden de los pasos podría visualizarse en la siguiente figura.

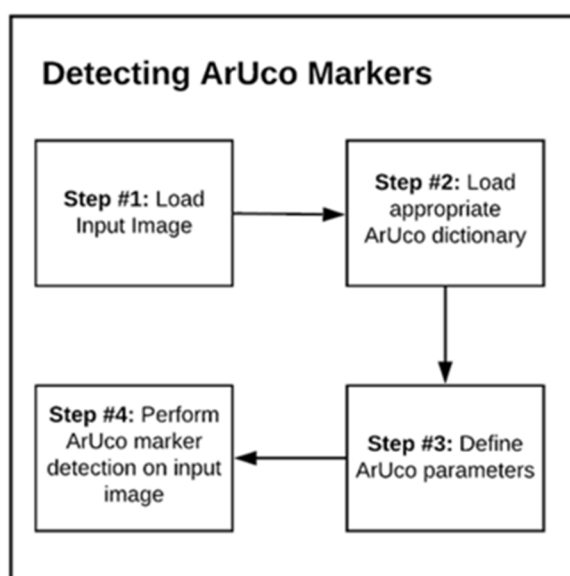


Figura 24. Rumbo a seguir para detección de un marcador

Tal y como se puede observar en la figura 19, el código de detección de marcadores se basa en encontrar en la imagen de entrada formas cuadradas que puedan ser candidatas a ser un marcador. Posteriormente, se realiza una segmentación mediante umbralización y se extraen los contornos de la imagen. Una vez encontrados los posibles candidatos, y si cumple los requisitos para ser un marcador, se analiza su codificación interna para comprobar si pertenece a la librería que se ha especificado.

Para la realización de lo que se ha comentado anteriormente, existe un módulo llamado `cv2.aruco.detectMarkers()`.

SOFTWARE DEL PROTOTIPO - MÓDULO ARUCO

Este módulo tendrá como parámetros de salida la información sobre los contornos (los puntos que pertenecen a cada una de las esquinas del marcador) y el id que posee el marcador en cuestión. En el siguiente diagrama de bloques se podrá observar el funcionamiento del código correspondiente para poder detectar un marcador.

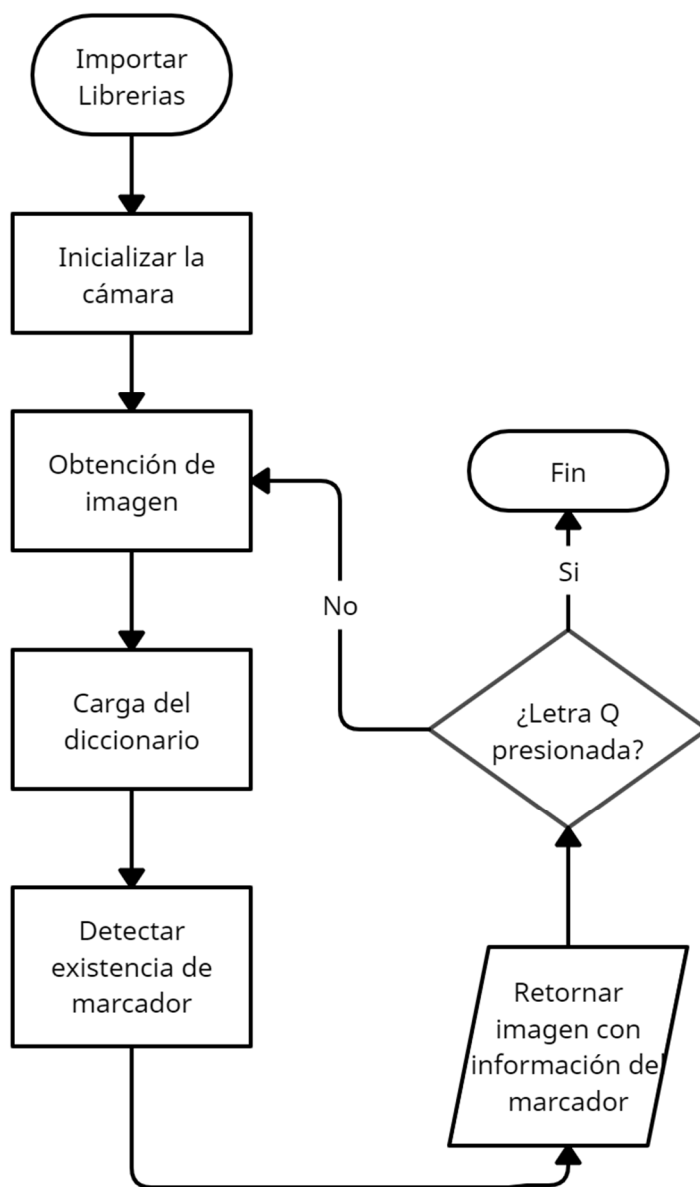


Figura 25. Diagrama de bloques detección de marcador

Para poder parar el código cuando se vea oportuno, existe un condicionante que, si se pulsa la tecla Q, se accede a un *break*, que sale del *loop for*.

4.1.3 - Estimar posición de un marcador ArUco

Una vez detectado el marcador correspondiente, lo siguiente que se querrá conseguir será obtener la posición que tiene el marcador respecto a la cámara. Para ello, se hará una estimación de la posición, que necesitará una calibración (se hablará más profundamente en el apartado *Calibración de la cámara*) correcta de la cámara. Lo que se quiere conseguir con la calibración de la cámara es una matriz 3x3, que albergará las distancias focales y coordenadas del centro de la cámara (conocidos como parámetros intrínsecos) y los coeficientes de distorsión. Este último es un vector que modela la distorsión producida por la cámara.

Estos parámetros se pasarán como entrada a la función `cv2.aruco.estimatePoseSingleMarkers()` y retornará el vector de rotación y traslación. Este último será el que ayude al prototipo a conocer la posición del marcador respecto al prototipo. Visualmente, se verá en el centro del marcador los ejes que muestran la orientación del marcador tal y como se muestra en la siguiente figura.

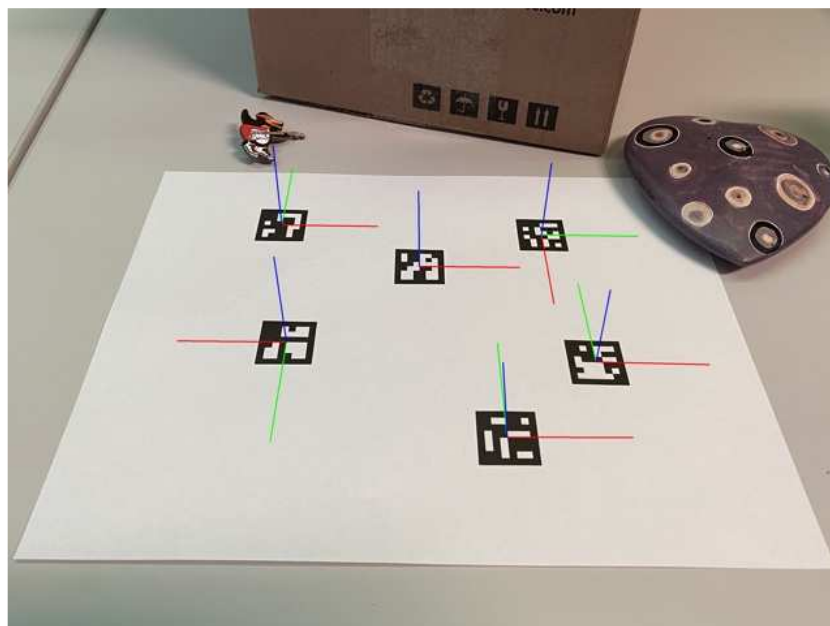


Figura 26. Visualización de los marcadores con los ejes de coordenadas creados por la estimación de posición

En el siguiente flujo de datos se observa la composición del código que además de realizar una detección del marcador, estima la posición donde este está, con respecto a la cámara.

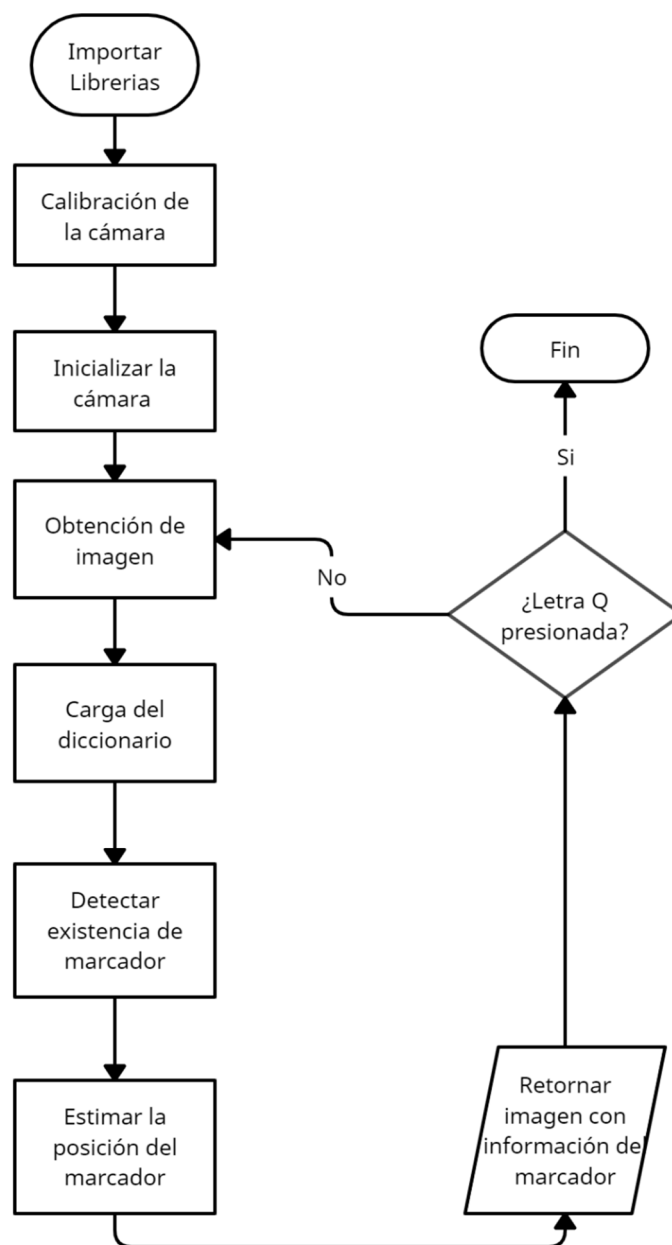


Figura 27. Diagrama de bloques estimación de posición de marcadores

Para poder parar el código cuando se vea oportuno, existe un condicionante que, si se pulsa la tecla Q, se accede a un *break*, que sale del *loop for*.

4.2 - Diseño y programación de una aplicación

La aplicación final para llevar a cabo la funcionalidad de este proyecto se podrá explicar en varios bloques. En primer lugar, habrá una calibración de la cámara utilizando la función llamada, *cv2.findChessboardCorners()* que aporta la librería de OpenCV. Seguidamente, se procederá a la detección y clasificación de los marcadores, en el caso de que se haya activado dicha función mediante la interfaz de usuario. Y por último se estimará la distancia a la que está el invidente del marcador, para poder darle una indicación correcta a través de unos auriculares de conector tipo jack.

4.2.1 - Calibración de la cámara

Si se quiere conocer la orientación del marcador fiducial en el entorno, es imprescindible conocer la posición de la cámara respecto a este. Para ello es necesario saber la matriz intrínseca de la cámara, que está compuesta por las longitudes focales f_x y f_y expresadas en píxeles, y los puntos c_x y c_y que representa los puntos del centro de la imagen.

$$C = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Con dicha matriz conocida, ya se podría saber los valores 3D del punto de interés a través de los vectores de rotación y traslación que proporciona la función que se utilizará para estimar la posición del punto, pero de ello se hablará más adelante, mientras tanto, sigamos con la calibración de la cámara. Como se mencionó anteriormente, se utilizará la función *cv2.findChessboardCorners()*, que su funcionalidad es la de encontrar tableros de ajedrez en una serie de fotografías tomadas con la cámara. Se almacenarán los datos de las posiciones donde se encontraba el tablero en las fotografías y se introducirá como entrada a la función *cv2.calibrateCamera()* para obtener la matriz de la cámara y los vectores mencionados anteriormente.

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

En este caso, se han utilizado dieciséis imágenes, en las cuales, el tablero de ajedrez se encuentra en distintas posiciones respecto a la cámara y con distintos brillos que se muestran en la siguiente ilustración.

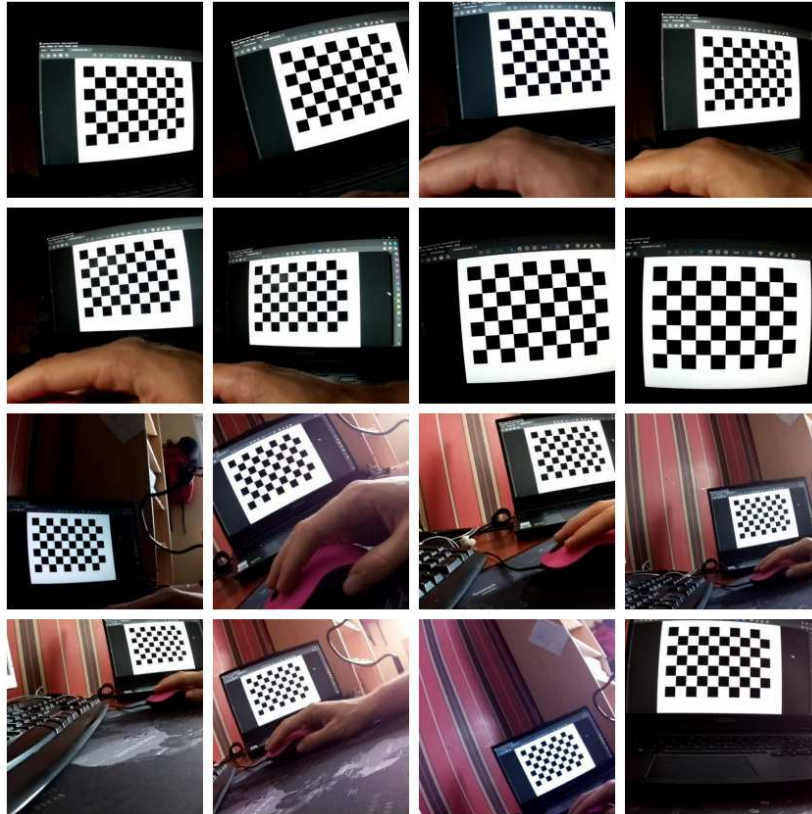


Figura 28. Banco de imágenes para la calibración de la cámara

Un dato a tener en cuenta para la calibración de la cámara, es que la distancia focal de esta debe de ser siempre la misma, cualquier cambio en la cámara afectaría a la calibración de esta y se tendría que utilizar otro banco de imágenes con las características actuales.

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

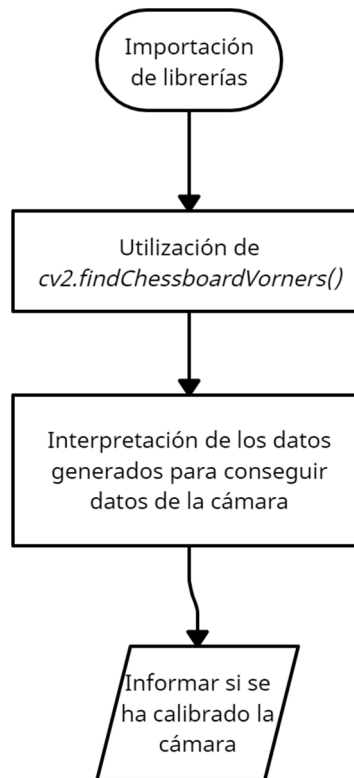


Figura 29. Diagrama de bloques para calibración de una cámara

4.2.2 - Interfaz de usuario

La interfaz de usuario estará codificada mediante *callbacks* usando pulsadores con algunos pines del GPIO de la RPi.

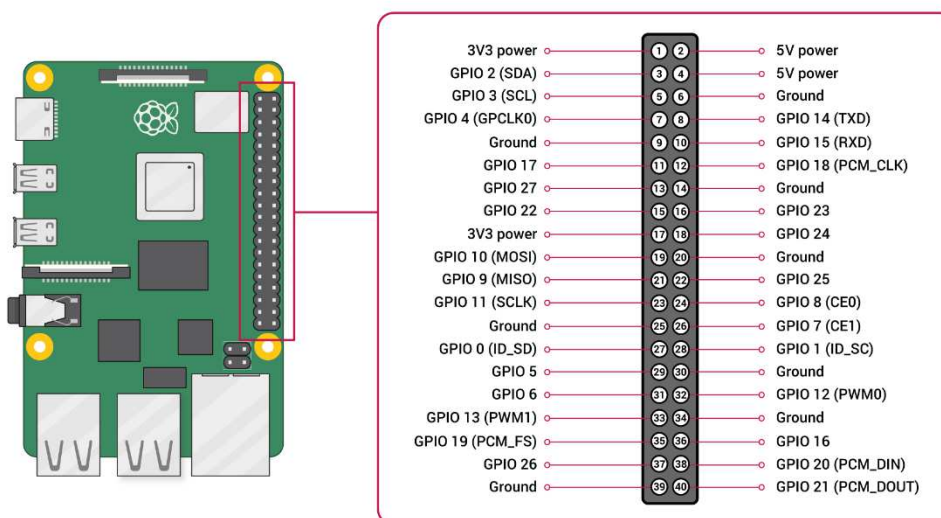


Figura 30. GPIO RPi 4B

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

Como se ha mostrado en la configuración hardware del prototipo, los pulsadores estarán situados, sujetos con silicona, en el soporte situado más próximo al GPIO. Estos pulsadores mandarían un pulso que daría un valor *TRUE* o *FALSE* a unas variables lógicas que se encargarán de realizar la función que se les dé.

Un pulsador está destinado para activar y pausar la detección de marcadores para satisfacer la intención del usuario si en algún momento decide que quiere dejar de recibir información.

Otro pulsador dará un valor positivo o negativo a una determinada variable para indicar que se va a procesar la imagen en modo día. Por defecto, el código está programado para que en un principio se realice la adquisición de la imagen en modo día. Este modo adquiere la imagen y no le hace ningún tipo de procesado.

Por último, existe otro botón que activará o desactivará la variable correspondiente al modo noche que hará que una imagen con poca luz, en la que es difícil distinguir un marcador fiducial, mejore el brillo y la saturación de la imagen.

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

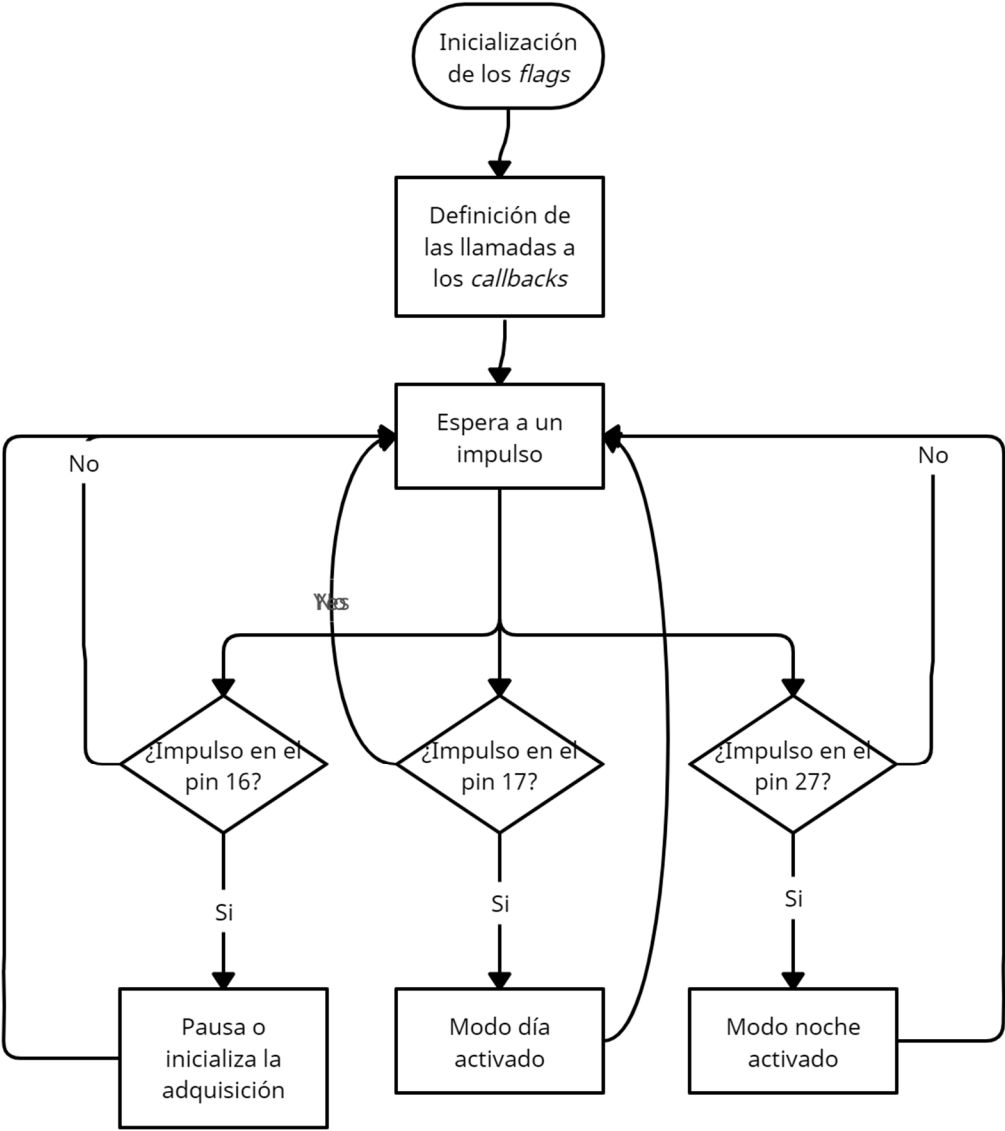


Figura 31. Diagrama de bloques para los *callbacks*

4.2.3 - Detección, clasificación y estimación de posición cámara-marcador

La detección de los marcadores de una forma próxima a la figura veinticinco. Pero en esta ocasión existirá dos caminos distintos según la luminosidad de la que se disponga. Para esa función estará la botonera comentada anteriormente.

La clasificación se hará mediante condicionales, que según el valor del ID, informará al usuario del punto de interés que está a menos de dos metros y su orientación con respecto a la cámara.

Con los datos adquiridos mediante la calibración de la cámara, utilizando la función `cv2.estimatePose()` se obtiene el vector de traslación que tendrá la información sobre donde se encuentra en el plano 2D de la cámara y la profundidad 3D a la que se encuentra el marcador fiducial.

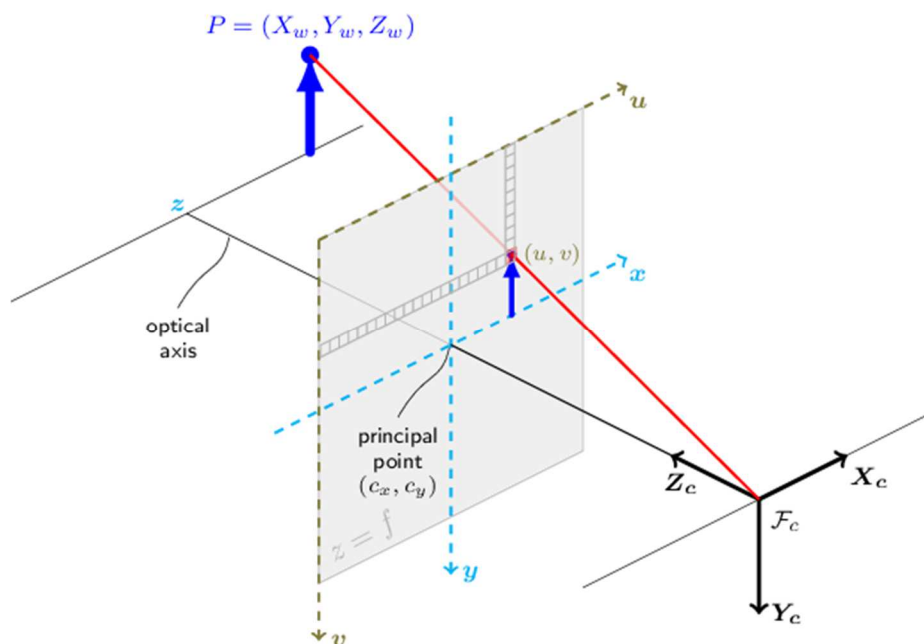


Figura 32. Visualización del vector de traslación

Como se puede observar en la figura 24, la cámara sería el punto F_c y el punto de interés, el punto P . Lo que ofrece el vector de traslación, que se llamará $Tvec$, será la posición x e y con respecto al punto principal (c_x, c_y) , además de la profundidad z a la que este se encuentra.

La distancia se consigue tras una conversión. Dicha conversión se realiza tras recoger los valores que otorgaba el vector de traslación para una distancia de 0.6

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

metros, teniendo el prototipo colocado a la altura del pecho. En total se recogen un total de sesenta y ocho muestras, a las que se les hallará el valor medio, para tener un valor lo más aproximado posible para la distancia de 0.6 metros (ilustración 27). A partir de ahí, las demás distancias serán calculadas mediante una sencilla regla de tres:

$$d = \frac{tvec[2] * 0.6}{0.2469455}$$

Donde d es la distancia a calcular y $tvec[2]$ es la profundidad otorgada por el vector de traslación. El valor 0.2469455 corresponde al valor medio de las muestras recogidas. La fórmula anterior se utilizará para poder calcular la distancia de 2 metros, puesto que esta será la distancia, a partir la cual, el prototipo detecte los distintos marcadores.

$$2 = \frac{tvec[2] * 0.6}{0.2469455}$$

$$tvec[2] = \frac{0.2469455 * 2}{0.6}$$

$$tvec[2] = 0.8231518$$

El resultado de arriba, será una condición más que estará en la detección del marcador. En la figura siguiente se podrán observar las muestras recogidas para llevar a cabo este cálculo.

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

Profundidad a 0,6 m	Media	2 m
0,24566689	0,2469455	0,8231518
0,24581103		
0,24587197		
0,24736428		
0,24725066		
0,24725066		
0,24628656		
0,24700145		
0,24706678		
0,24598963		
0,24598963		
0,24592510		
0,24691781		
0,24710950		
0,24763772		
0,24603400		
0,24729162		
0,24770363		
0,24683006		
0,24627471		
0,24696097		
0,24566912		
0,24711672		
0,24758883		
0,24691689		
0,24720558		
0,24582447		
0,24582447		
0,24685821		
0,24726129		
0,24599104		
0,24576665		
0,24762136		
0,24774361		
0,24715048		
0,24697073		
0,24599748		
0,24599748		
0,24717384		
0,24600956		
0,24600956		
0,24765713		
0,24711052		
0,24765713		
0,24730245		
0,24705957		
0,24598484		
0,24709335		
0,24598816		
0,24603135		
0,24708226		
0,24761644		
0,24761644		
0,24780702		
0,24766816		
0,24754802		
0,24733534		
0,24776052		
0,24759656		
0,24731254		
0,24630427		
0,24741307		
0,24663169		
0,24766829		
0,24652609		
0,24770933		
0,24829827		
0,24939118		
0,24914031		

Tabla 1. Recolecta de datos para 60 cm y cálculo de valor para 2 m

SOFTWARE DEL PROTOTIPO - DISEÑO Y PROGRAMACIÓN DE UNA APLICACIÓN

Finalmente, el flujo de datos del prototipo vendría dado por el siguiente diagrama. La figura 30 también formaría parte de este diagrama, puesto que una espera a un *callback*

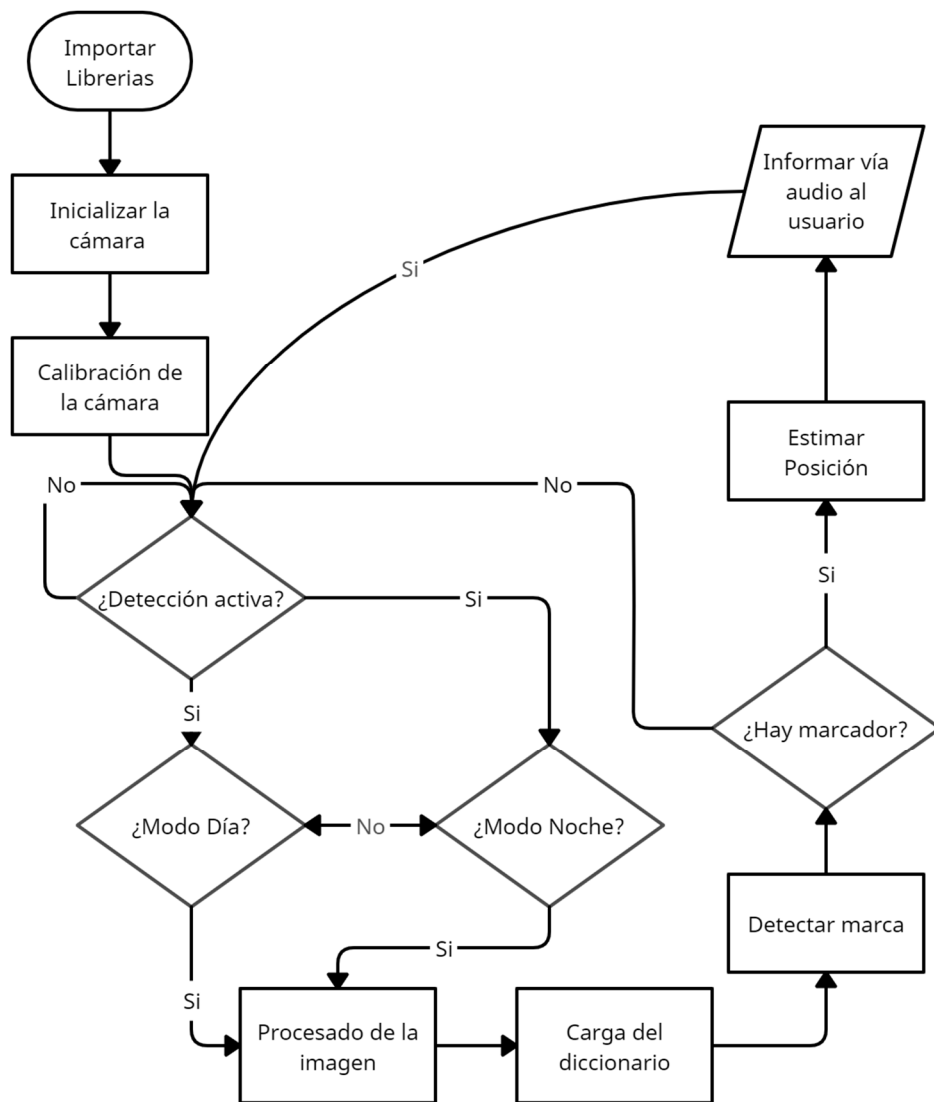


Figura 33. Diagrama de bloques aplicación

5 - RESULTADOS

Los ensayos del prototipo se han realizado mediante grabaciones de pantalla de la ventana que el propio código muestra mientras el programa está en ejecución. Una vez terminado el ensayo, se revisa el vídeo para comprobar cómo ha ido. Además, como el prototipo comunica, vía audio, la posición de los marcadores, también sirve para comprobar que todo está yendo bien.

Por lo tanto, lo que se podrá mostrar en esta memoria, serán capturas tomadas de los distintos vídeos que se han realizado.

5.1 - Distancia

Los test para comprobar que la estimación de la distancia es la correcta, se realizan con un marcador ArUco adherido a la pared y el prototipo situado a una determinada distancia controlada por un metro.

Lo que se podrá observar será el marcador detectado y, abajo a la izquierda, la posición número dos del vector de traslación, que muestra la profundidad del punto de interés. Las figuras a continuación son una captura de la grabación que se realiza para poder testear el funcionamiento del algoritmo de estimación de posición y distancia.

Es importante resaltar que, en este tipo de prueba, resulta imposible recoger una tabla comparativa comparando la distancia medida por el prototipo y la distancia real, puesto que el metro, es un metro convencional. Quizás si hubiera sido posible con un metro digital, pero no se dispone de ese medio. No obstante, se obtienen unos valores muy cercanos a la realidad, y el procedimiento a realizar las medidas será el de situarnos con nuestro prototipo colgado en el pecho junto a la distancia con la que se va a comparar la estimación.

RESULTADOS

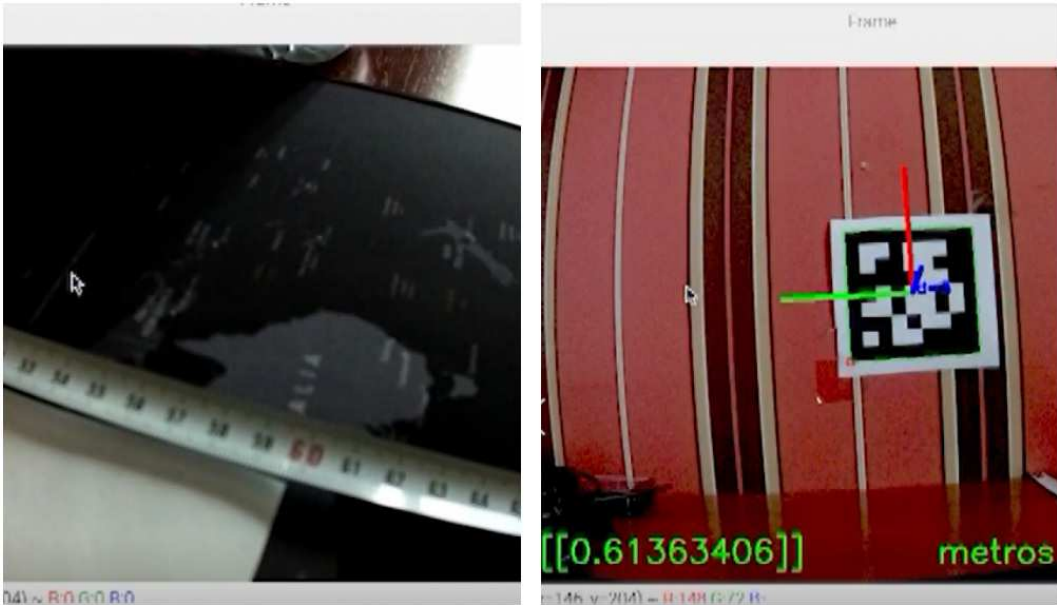


Figura 34. Prueba a 60 cm



Figura 35. Prueba a 50 cm

RESULTADOS

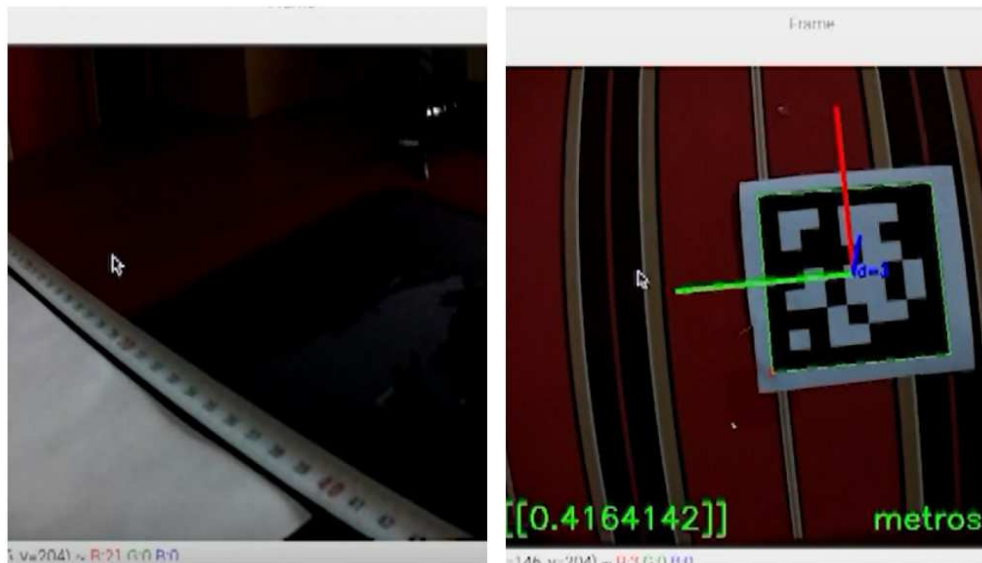


Figura 36. Prueba a 40 cm

Tal y como se puede observar, el resultado del algoritmo de estimación de la posición y distancia funciona de una forma satisfactoria, al menos a la luz de última hora de la tarde del mes de mayo. Además, el testeo se ha realizado con los dos modos de trabajo que posee el prototipo, correspondiéndose la figura treintaitrés con el modo noche, y las figuras treintaicuatro y treintaicinco con el modo día. No se aprecia diferencia de resultado con esta cantidad de luz.

5.1.2 - Comparación de medida de distancia entre marcadores de distinto tamaño

Los ensayos anteriores están realizados con marcadores que tienen un tamaño de 1500x1500 Mp, pero previamente, tenían un tamaño de 900x900 Mp. La decisión de aumentar los Mp de los marcadores fue tomada por mal funcionamiento del prototipo en su primera puesta en marcha. Para corroborar esta decisión, se realiza un ensayo con ambos marcadores para mostrar la diferencia en el funcionamiento del prototipo.

RESULTADOS



Figura 37. Ensayo a 2 m con marcador 1500x1500 Mp

N	Muestreo para 2 m			
	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,8446	2,052116868	2	2,605843406
2	0,8317	2,020773857	2	1,038692826
3	0,835	2,028791836	2	1,439591812
4	0,8321	2,021745733	2	1,087286643
5	0,8369	2,033408249	2	1,67041244
6	0,8351	2,029034805	2	1,451740266
7	0,8448	2,052602806	2	2,630140315
8	0,8357	2,03049262	2	1,524630991
9	0,8353	2,029520743	2	1,476037174
10	0,8459	2,055275466	2	2,76377331
11	0,835	2,028791836	2	1,439591812
12	0,8405	2,042155136	2	2,107756788
13	0,8295	2,015428537	2	0,771426836
14	0,8462	2,056004373	2	2,800218672
15	0,8367	2,032922311	2	1,646115532

Tabla 2. Recolección de datos y cálculo del error cometido para 2 m y marcador 1500x1500 Mp

RESULTADOS



Figura 38. Ensayo a 1.8 m con marcador 1500x1500 Mp

Muestreo para 1.8 m				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,7465	1,813764199	1,8	0,764677695
2	0,7466	1,814007168	1,8	0,778175977
3	0,7468	1,814493106	1,8	0,805172542
4	0,7463	1,81327826	1,8	0,73768113
5	0,7465	1,813764199	1,8	0,764677695
6	0,7515	1,825912653	1,8	1,439591812
7	0,7519	1,826884529	1,8	1,493584941
8	0,7498	1,821782178	1,8	1,210121012
9	0,7514	1,825669684	1,8	1,42609353
10	0,7602	1,847050963	1,8	2,613942376
11	0,7466	1,814007168	1,8	0,778175977
12	0,7472	1,815464982	1,8	0,859165671
13	0,7472	1,815464982	1,8	0,859165671
14	0,7513	1,825426714	1,8	1,412595247
15	0,7462	1,813035291	1,8	0,724182848

Tabla 3. Recolección de datos y cálculo del error cometido para 1.8 m y marcador 1500x1500 Mp

RESULTADOS

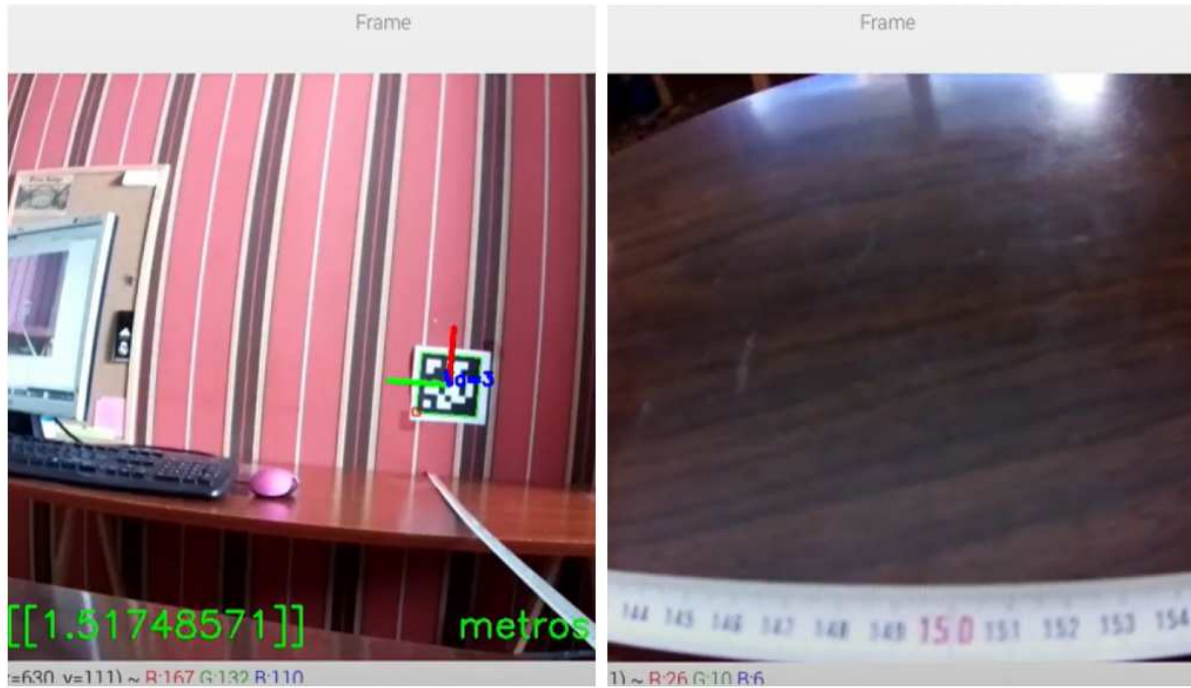


Figura 39. Ensayo a 1.5 m con marcador 1500x1500 Mp

Muestreo para 1.5 m				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,6199	1,50616534	1,5	0,411022697
2	0,6248	1,518070825	1,5	1,204721699
3	0,6196	1,505436433	1,5	0,362428881
4	0,6241	1,516370042	1,5	1,091336127
5	0,6243	1,51685598	1,5	1,123732005
6	0,621	1,508838	1,5	0,589200024
7	0,62449	1,517317621	1,5	1,154508089
8	0,6211	1,509080969	1,5	0,605397963
9	0,6198	1,505922371	1,5	0,394824759
10	0,6209	1,508595031	1,5	0,573002085
11	0,6219	1,511024722	1,5	0,734981474
12	0,6214	1,509809877	1,5	0,65399178
13	0,6218	1,510781753	1,5	0,718783535
14	0,6214	1,509809877	1,5	0,65399178
15	0,6199	1,50616534	1,5	0,411022697

Tabla 4. Recolección de datos y cálculo del error cometido para 1.5 m y marcador 1500x1500 Mp

RESULTADOS

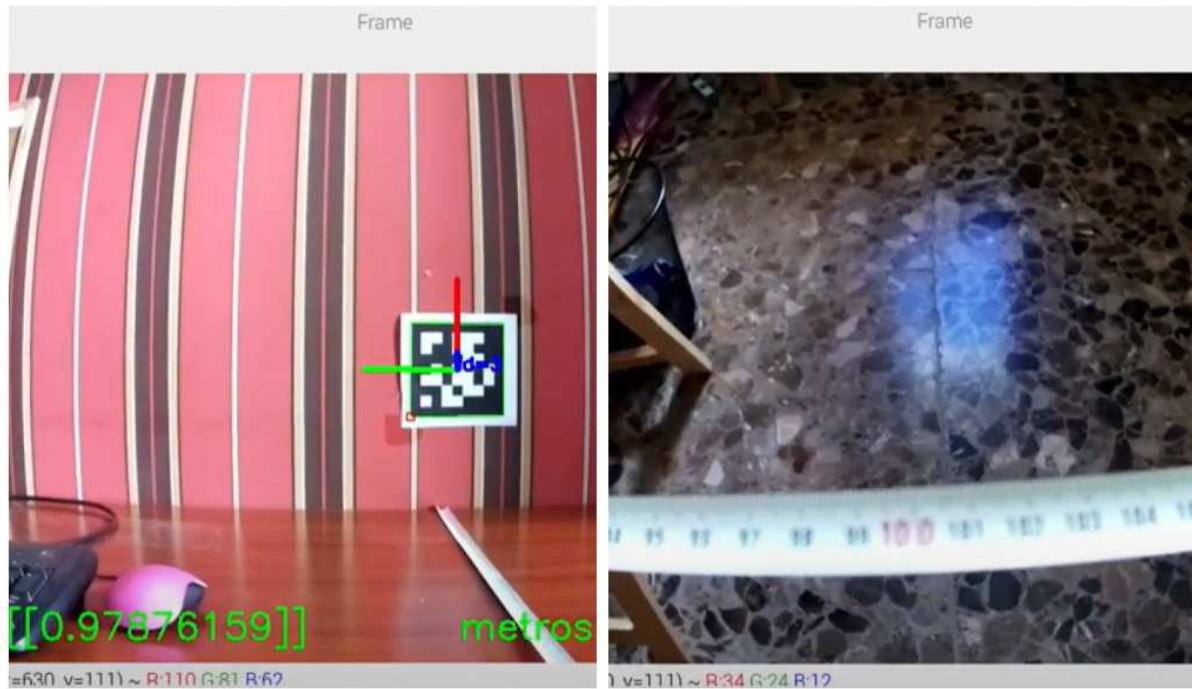


Figura 40. Ensayo a 1.8 m con marcador 1500x1500 Mp

N	Muestreo para 1 m			
	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,4086	0,99277167	1	0,722833019
2	0,4086	0,99277167	1	0,722833019
3	0,4069	0,988641195	1	1,135880459
4	0,4076	0,990341979	1	0,965802102
5	0,4063	0,987183381	1	1,281661909
6	0,4066	0,987912288	1	1,208771184
7	0,4066	0,987912288	1	1,208771184
8	0,4067	0,988155257	1	1,184474276
9	0,4066	0,987912288	1	1,208771184
10	0,4066	0,987912288	1	1,208771184
11	0,4067	0,988155257	1	1,184474276
12	0,4069	0,988641195	1	1,135880459
13	0,4065	0,987669319	1	1,233068092
14	0,4065	0,987669319	1	1,233068092
15	0,4064	0,98742635	1	1,257365

Tabla 5. Recolección de datos y cálculo del error cometido para 1 m y marcador 1500x1500 Mp

Como se ha podido observar en las figuras y las tablas anteriores, la extracción de la información del marcador ha sido muy positivo, con un error sobre el valor esperado muy bajo. Finalmente, el error medio cometido en el ensayo ha sido de un 1,178260566 %.

RESULTADOS

A través de la figura 41, que muestra el comportamiento de la estimación de la distancia para cada punto estudiado, se puede llegar a la conclusión de que cuanto más cerca del marcador fiducial, existe menos distorsión entre la estimación y la referencia.

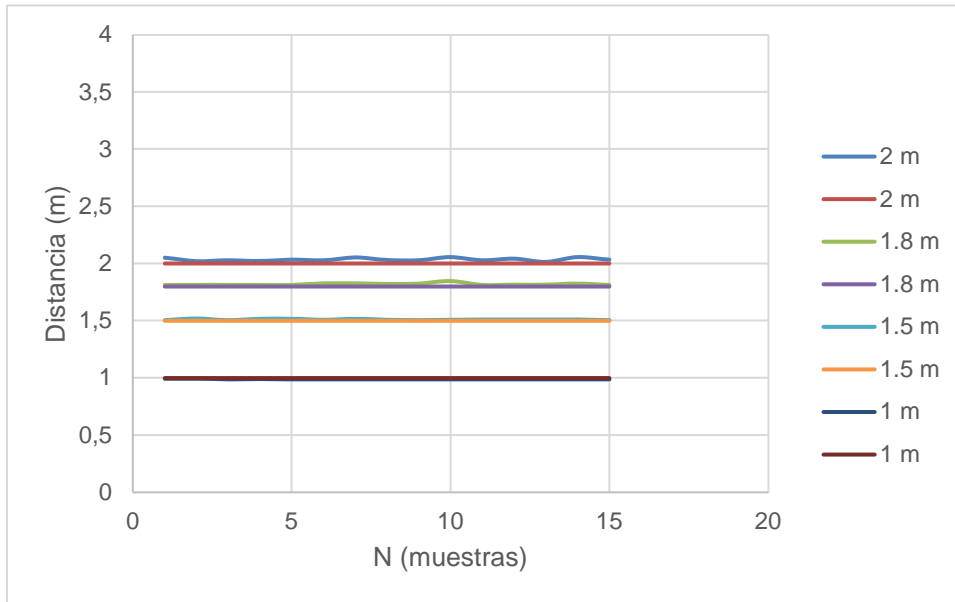


Figura 41. Comportamiento de la estimación de distancia

No se puede decir lo mismo sobre los ensayos realizados con el marcador más pequeño, que, en un principio, fue la primera elección, pero que, con las características de la cámara del prototipo, los valores observados no corresponden con la realidad.

RESULTADOS

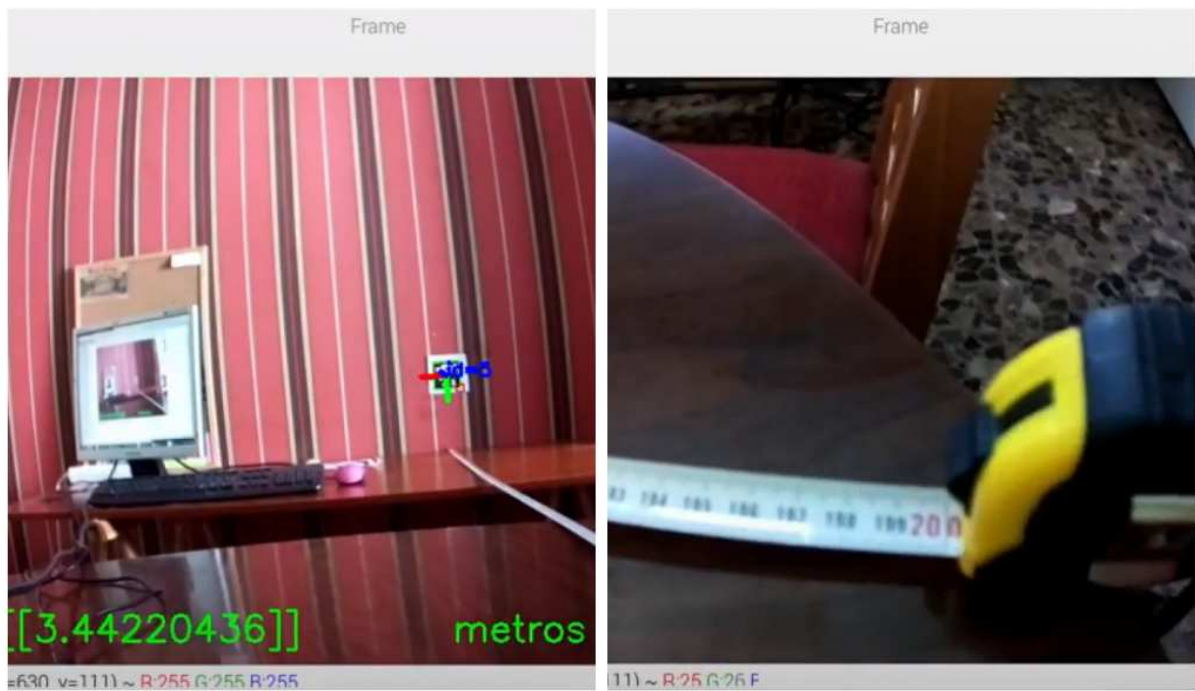


Figura 42. Ensayo a 2 m con marcador 900x900 Mp

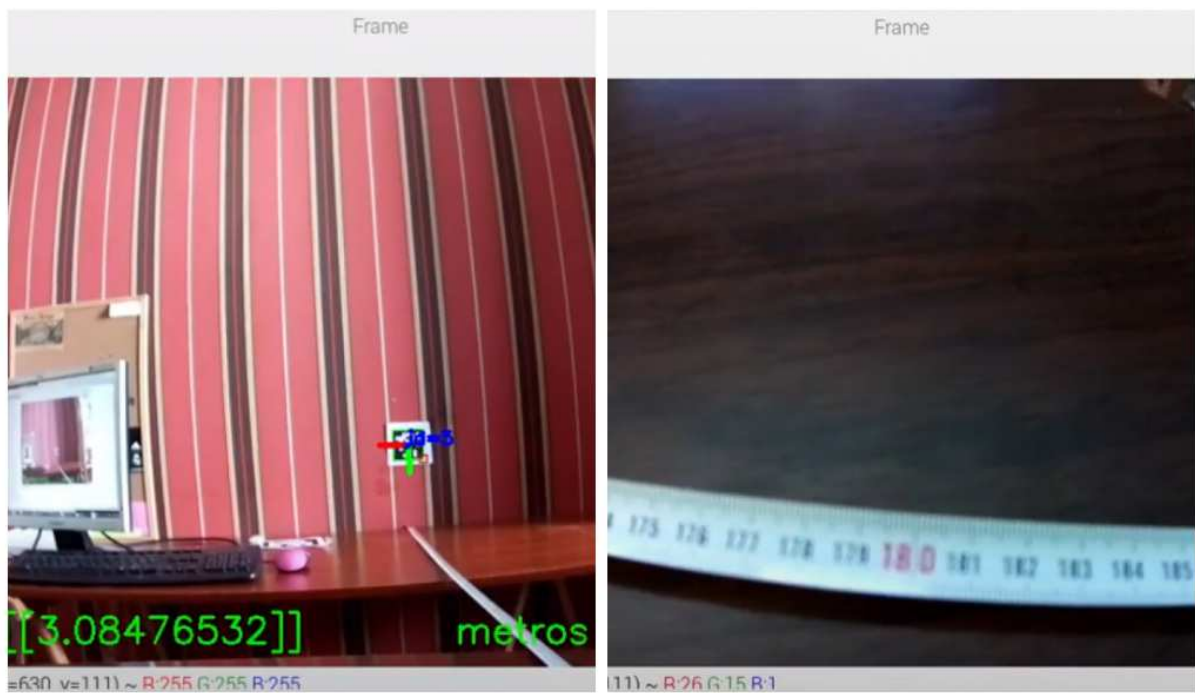


Figura 43. Ensayo a 1.8 m con marcador 900x900 Mp

RESULTADOS

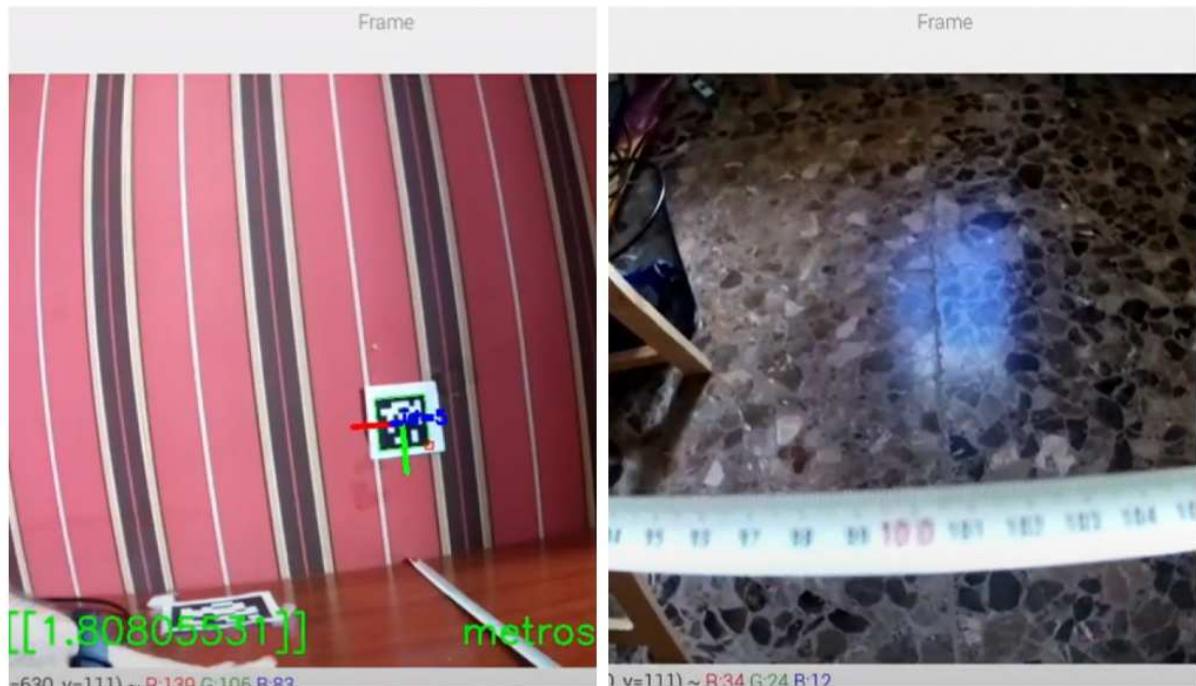


Figura 44. Ensayo a 1 m con marcador 900x900 Mp

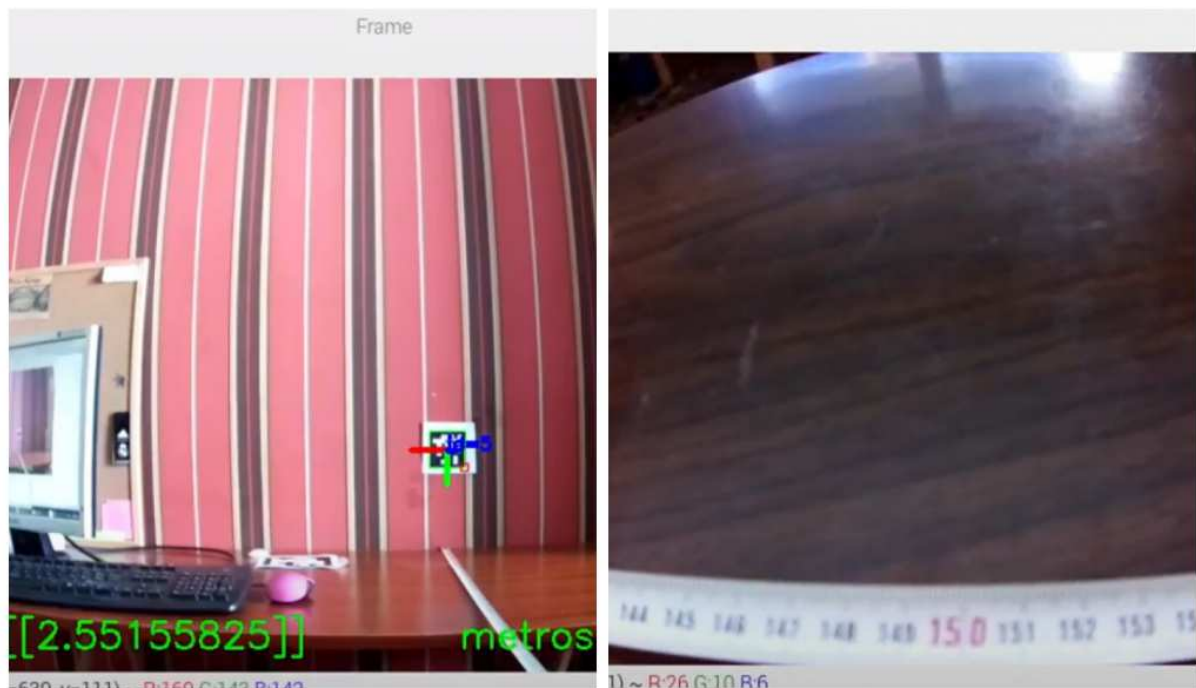


Figura 45. Ensayo a 1.5 m con marcador 900x900 Mp

En este caso no es relevante la realización de una tabla para conseguir el error cometido en la estimación de la distancia, puesto que a simple vista se puede ver que

RESULTADOS

es bastante. De hecho, al prototipo le cuesta reconocer en ciertas ocasiones el marcador, por lo que, incluso salen valores dispares para una misma medida,

5.2 - Modo Día y Modo Noche

Este subapartado está destinado para poder visualizar pruebas durante el día y durante la noche, para poder comparar cómo se comporta el prototipo en casos de una suficiente luminosidad, y en casos totalmente opuestos.

En primer lugar, se puede observar la figura cuarentaitrés, la cual contiene información de un marcador situado a la izquierda del usuario y como se puede ver en la consola del programa que corre el algoritmo, lo detecta satisfactoriamente. Esta información, como ya se ha comentado anteriormente, llega al usuario a través de unos auriculares con conexión tipo *jack*. También aparece unas unidades de metro, pero en este ensayo es irrelevante esa información.

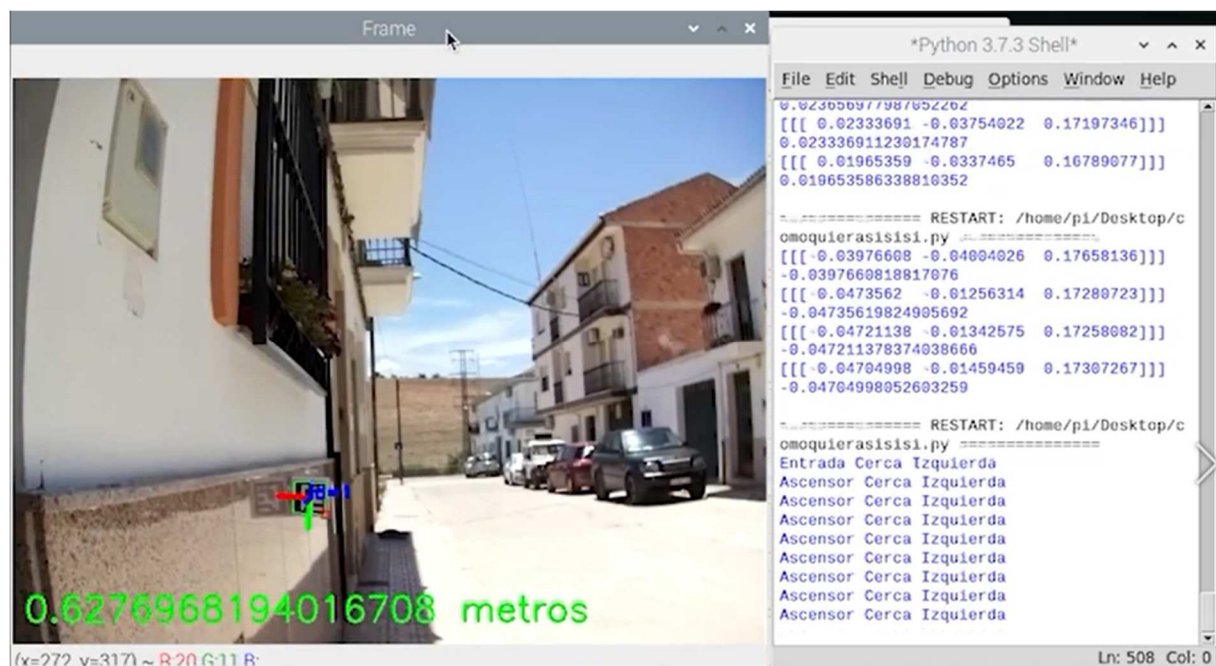


Figura 46. Identificación de un marcador a la izquierda del usuario a la luz del día usando el Modo Día

RESULTADOS

Para comprobar el correcto funcionamiento, en los dos sentidos posibles de la orientación, izquierda o derecha, también se deposita un par de marcadores para que el usuario lo pueda detectar a mano derecha.



Figura 47. Identificación de un marcador a la derecha del usuario a la luz del día usando el Modo Día

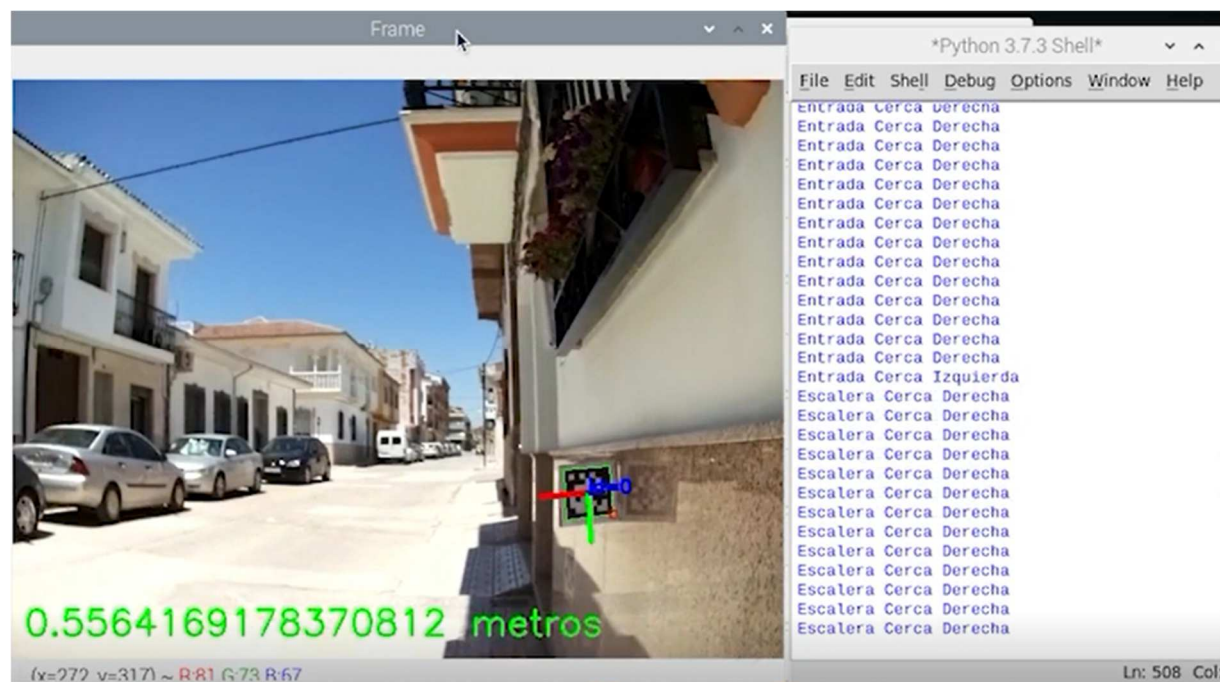


Figura 48. Identificación de un marcador a la derecha del usuario a la luz del día usando el Modo Día

RESULTADOS

Habiendo visto que el prototipo actúa correctamente durante la luz del día, es hora de comprobar el comportamiento en horario nocturno. Como se puede observar en las figuras cuarenta y seis y cuarenta y siete, cuando el marcador está iluminado por una lámpara, con una buena intensidad lumínica, el prototipo no suele tener problemas y cumple con su función, ya sea usando el Modo Noche o el Modo Día.

Aunque cumpla sin problemas, el trabajo lo realiza más óptimo con el Modo Noche activado una vez que ha empezado a ocultarse el Sol.

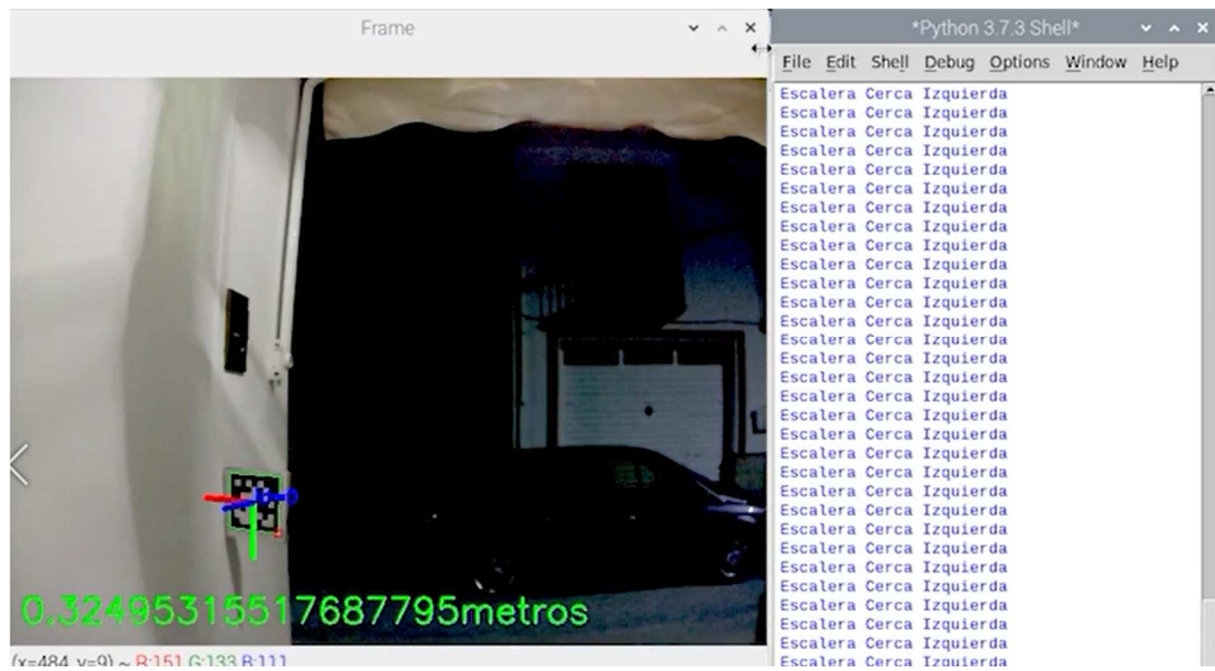


Figura 49. Identificación de un marcador a la izquierda del usuario usando el Modo Noche

RESULTADOS

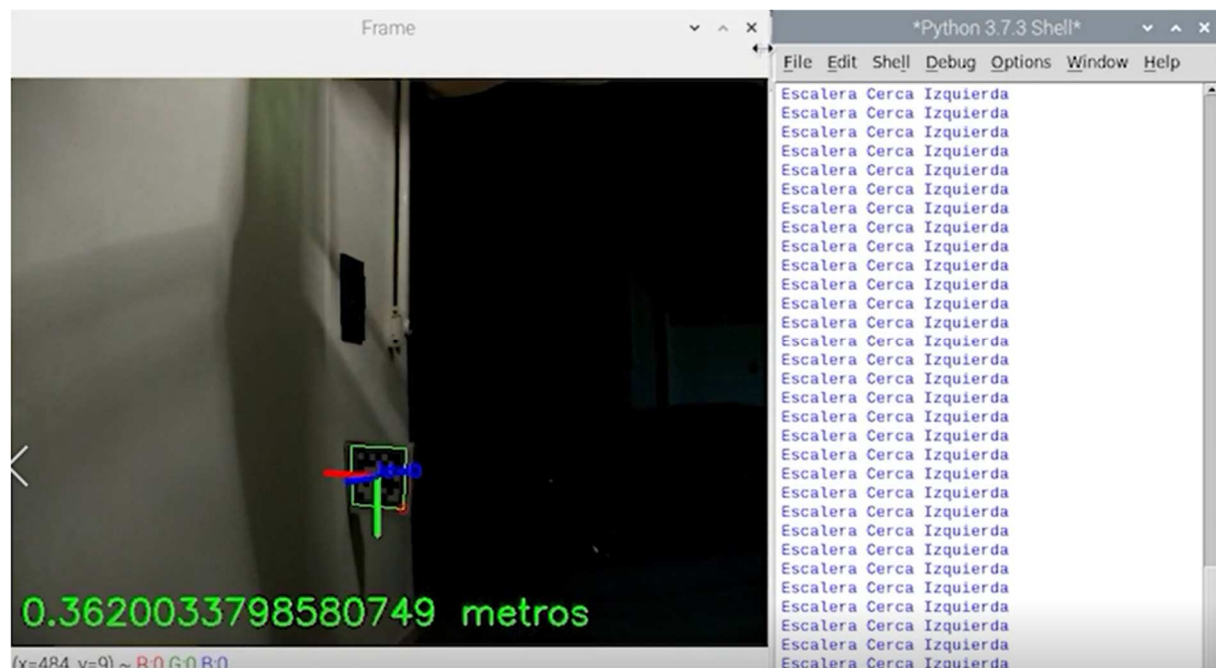


Figura 50. Identificación de un marcador a la izquierda del usuario usando el Modo Día

Y, por último, un ejemplo más visual de la diferencia entre el Modo Día y Modo Noche. El contexto es que el marcador se encuentra justo debajo de una farola a la altura del pecho del usuario, por lo que se tiene una menor luminosidad con respecto al caso anterior. Bien, el Modo Noche actúa sin problemas y puede localizar el marcador de una manera exacta, mientras que en el Modo Día a penas si se puede sacar algún tipo de información.

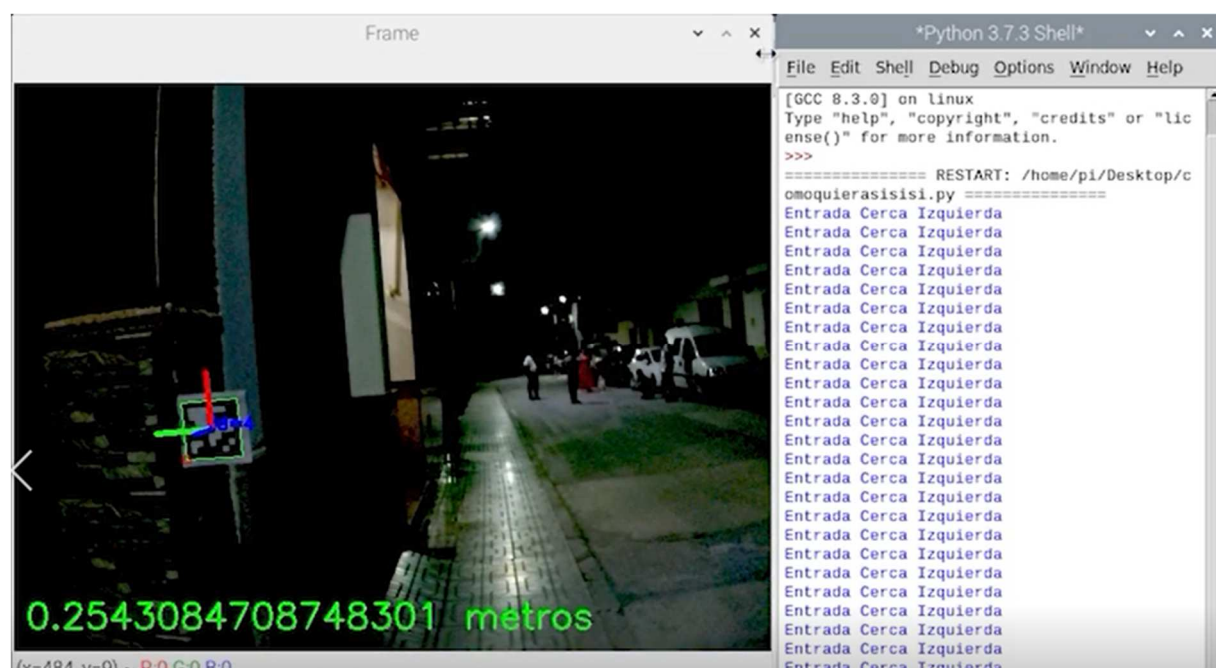


Figura 51. Identificación de un marcador a la izquierda del usuario usando el Modo Noche

RESULTADOS

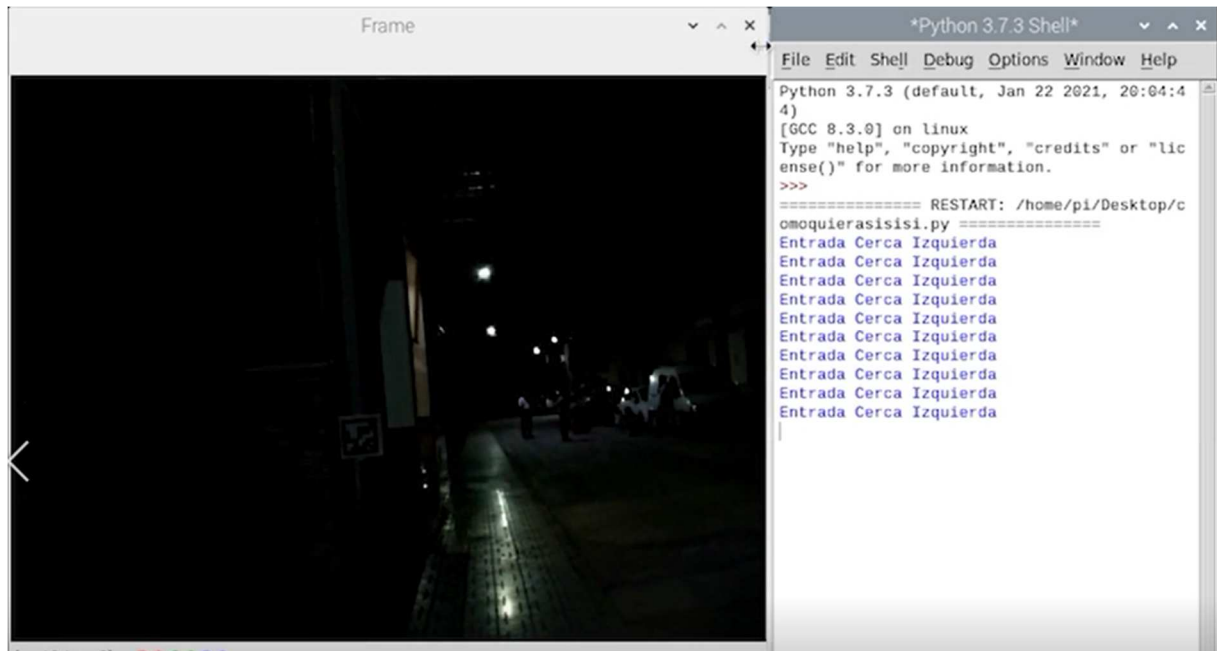


Figura 52. Intento de identificación de un marcador a la izquierda del usuario usando el Modo Día

Para finalizar el presente capítulo, se realiza el mismo ensayo que se ha llevado a cabo en el punto 5.1.2. Alternando entre el Modo Noche y el Modo Día, con la luz de interior correspondiente a última hora de la tarde. Se empezará exponiendo los resultados obtenidos con el prototipo trabajando en Modo Día.

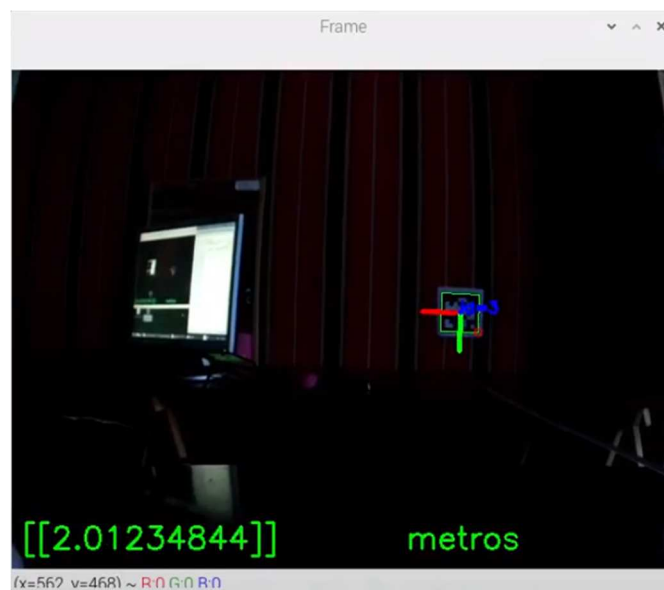


Figura 53. Ensayo a 2 m test baja luminosidad Modo Día

RESULTADOS

Muestra para 2 metros test baja luminosidad Modo Día				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,8349	2,028548867	2	1,427443358
2	0,8305	2,017858228	2	0,892911377
3	0,8307	2,018344166	2	0,917208285
4	0,8219	1,996962886	2	0,151855676
5	0,8214	1,995748041	2	0,212597947
6	0,8216	1,996233979	2	0,188301039
7	0,8199	1,992103505	2	0,394824759
8	0,8295	2,015428537	2	0,771426836
9	0,8226	1,99866367	2	0,066816498
10	0,8233	2,000364454	2	0,018222681
11	0,823	1,999635546	2	0,018222681
12	0,8255	2,005709773	2	0,285488672
13	0,8236	2,001093361	2	0,054668043
14	0,8375	2,034866063	2	1,743303165
15	0,8323	2,022231671	2	1,111583551

Tabla 6. Recolección de datos y cálculo del error cometido para 2 m test baja luminosidad Modo Día

La detección del marcador se realiza, pero no de una manera poco deseada, puesto que el prototipo advierte que el marcador está a la izquierda. Todo indica que la profundidad no sufre, a priori, pero si la orientación.

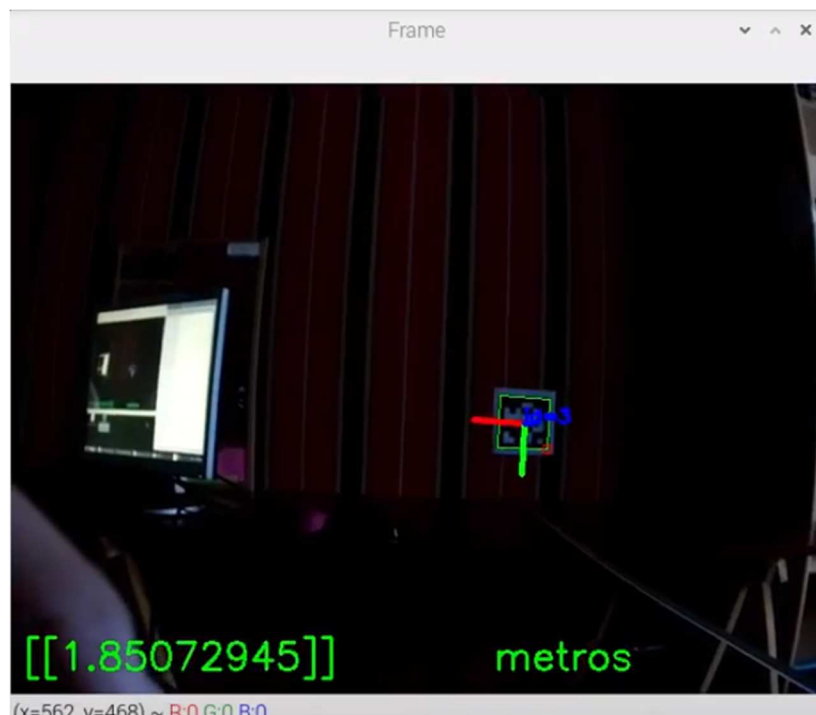


Figura 54. Ensayo a 1.8 m test baja luminosidad Modo Día

RESULTADOS

Muestro para 1,8 metros test baja luminosidad Modo Día				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,7664	1,862115046	1,8	3,450835881
2	0,7598	1,846079086	1,8	2,559949246
3	0,7592	1,844621272	1,8	2,478959552
4	0,7545	1,833201725	1,8	1,844540282
5	0,7636	1,855311912	1,8	3,072883976
6	0,7656	1,860171293	1,8	3,342849622
7	0,7593	1,844864241	1,8	2,492457835
8	0,7656	1,860171293	1,8	3,342849622
9	0,7593	1,844864241	1,8	2,492457835
10	0,7656	1,860171293	1,8	3,342849622
11	0,7695	1,869647087	1,8	3,869282634
12	0,7626	1,852882221	1,8	2,937901152
13	0,7589	1,843892365	1,8	2,438464705
14	0,7588	1,843649396	1,8	2,424966423
15	0,7654	1,859685355	1,8	3,315853058

Tabla 7. Recolección de datos y cálculo del error cometido para 1.8 m test baja luminosidad Modo Día

Al igual que en el caso anterior, el prototipo llega a detectar el marcador, pero avisa de una manera errónea sobre la posición en la que se encuentra el marcador. También se puede observar que se ha incrementado el error cometido con respecto la distancia teórica, pero esto puede ser debido a la colocación del prototipo.

RESULTADOS

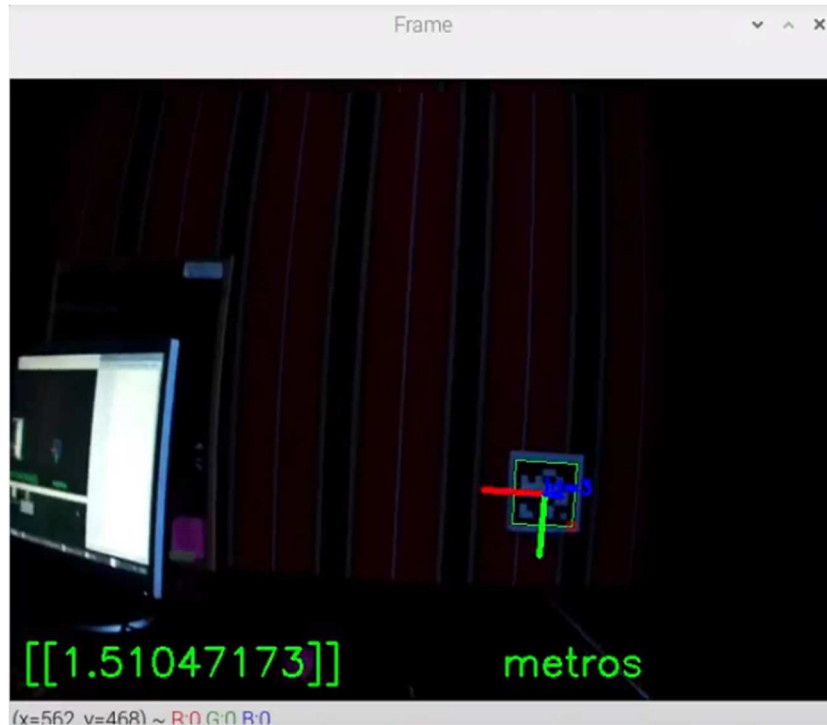


Figura 55. Ensayo a 1.5 m test baja luminosidad Modo Día

Muestra para 1,5 metros test baja luminosidad Modo Día				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,6328	1,537508352	1,5	2,500556804
2	0,6344	1,541395857	1,5	2,759723825
3	0,6374	1,54868493	1,5	3,24566199
4	0,6329	1,537751321	1,5	2,516754743
5	0,6334	1,538966167	1,5	2,597744437
6	0,6368	1,547227115	1,5	3,148474357
7	0,6321	1,535807568	1,5	2,387171232
8	0,6343	1,541152888	1,5	2,743525886
9	0,6343	1,541152888	1,5	2,743525886
10	0,6334	1,538966167	1,5	2,597744437
11	0,6343	1,541152888	1,5	2,743525886
12	0,6344	1,541395857	1,5	2,759723825
13	0,6355	1,544068517	1,5	2,937901152
14	0,6342	1,540909919	1,5	2,727327948
15	0,6335	1,539209136	1,5	2,613942376

Tabla 8. Recolección de datos y cálculo del error cometido para 1.5 m test baja luminosidad Modo Día

RESULTADOS

En esta ocasión el prototipo es capaz de informar correctamente de la orientación del marcador. Se mantiene el porcentaje de error que en la medida anterior.

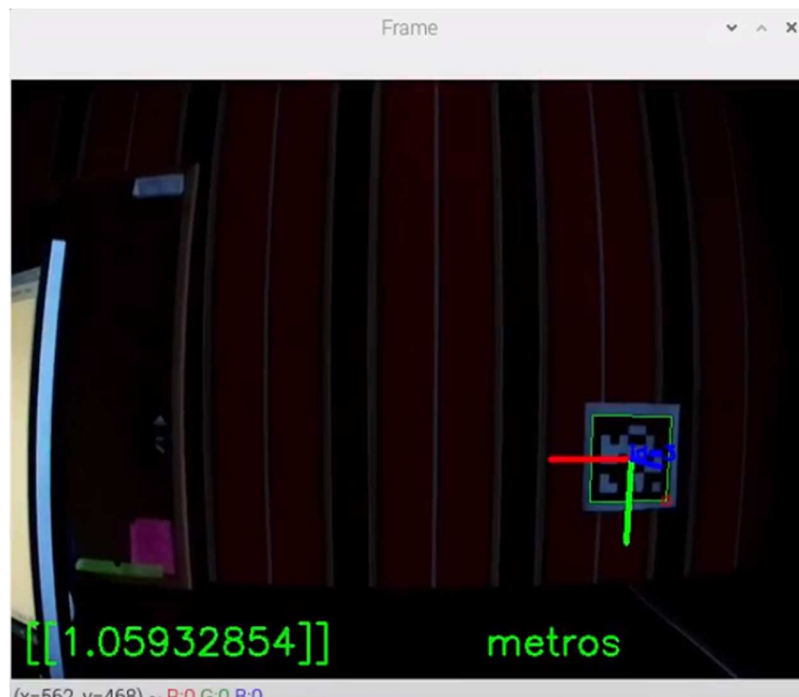


Figura 56. Ensayo a 1 m test baja luminosidad Modo Día

Muestra para 1 metro test baja luminosidad Modo Día				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,4447	1,080483508	1	8,048350847
2	0,4446	1,080240539	1	8,024053939
3	0,4483	1,089230395	1	8,923039543
4	0,4464	1,084613983	1	8,461398287
5	0,4478	1,08801555	1	8,801555002
6	0,4413	1,07222256	1	7,222255968
7	0,4438	1,078296787	1	7,829678673
8	0,4422	1,074409281	1	7,440928142
9	0,4385	1,065419425	1	6,541942538
10	0,4385	1,065419425	1	6,541942538
11	0,4423	1,074652251	1	7,46522505
12	0,4436	1,077810849	1	7,781084857
13	0,438	1,06420458	1	6,420457997
14	0,4316	1,048654559	1	4,865455871
15	0,4359	1,059102229	1	5,910222924

Tabla 9. Recolección de datos y cálculo del error cometido para 1 m test baja luminosidad Modo Día

A esta distancia, el prototipo conserva la buena detección y orientación conseguida en el punto anterior, pero, sin embargo, crece el error cometido.

RESULTADOS

Ahora se expondrán los resultados que se obtienen al cambiar al Modo Noche para poder llegar a una conclusión en este ensayo.

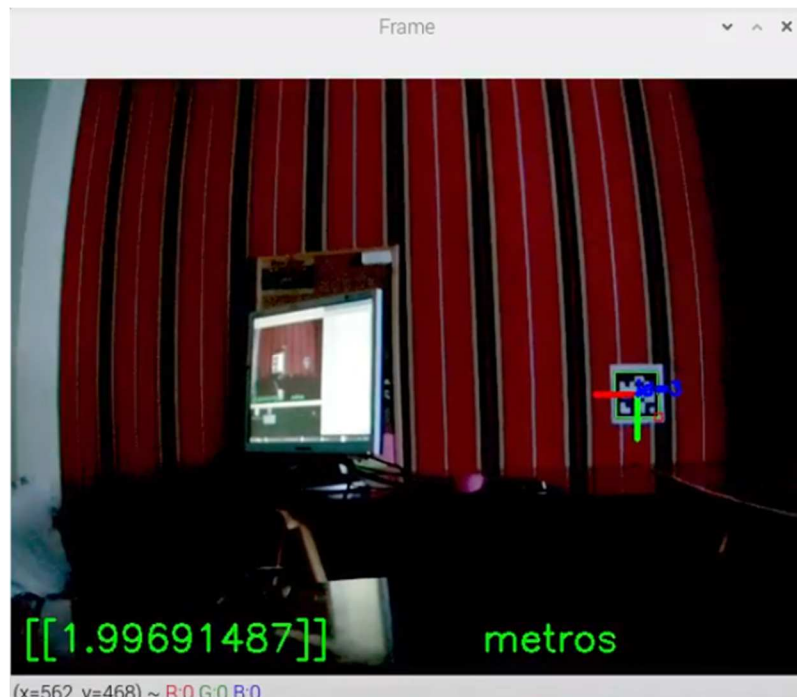


Figura 57. Ensayo a 2 m test baja luminosidad Modo Noche

En este modo, el prototipo no tiene problemas para ofrecer la orientación correcta del marcador, además de una buena detección y seguimiento. Habrá poco que comentar en los nuestros siguientes, por lo que no se ha notado ningún fallo en la ejecución de la funcionalidad del prototipo. Una vez mostradas todas las figuras y tablas, se realizará una valoración final con respecto el Modo Noche y el Modo Día en la estimación de la distancia con baja luminosidad.

RESULTADOS

Muestro para 2 metro test baja luminosidad Modo Noche				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,8045	1,954686266	2	2,265686691
2	0,8042	1,953957359	2	2,302132054
3	0,8183	1,988216	2	0,589200024
4	0,8155	1,981412865	2	0,929356739
5	0,8117	1,97218004	2	1,390997996
6	0,8128	1,9748527	2	1,257365
7	0,8134	1,976310514	2	1,184474276
8	0,8238	2,001579299	2	0,078964952
9	0,8042	1,953957359	2	2,302132054
10	0,8123	1,973637855	2	1,318107271
11	0,8237	2,00133633	2	0,066816498
12	0,8019	1,94836907	2	2,581546498
13	0,8002	1,944238596	2	2,788070218
14	0,81	1,968049566	2	1,597521715
15	0,8208	1,994290227	2	0,285488672

Tabla 10. Recolección de datos y cálculo del error cometido para 2 m test baja luminosidad Modo Noche

RESULTADOS

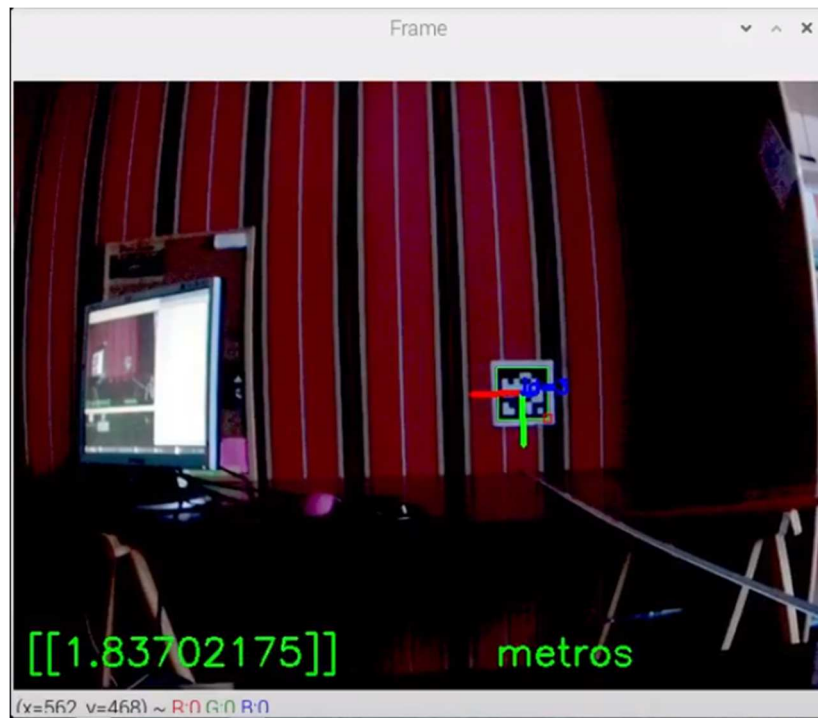


Figura 58. Ensayo a 1.8 m test baja luminosidad Modo Noche

Muestro para 1,8 metro test baja luminosidad Modo Noche				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,7666	1,862600984	1,8	3,477832446
2	0,7555	1,835631416	1,8	1,979523106
3	0,7616	1,85045253	1,8	2,802918329
4	0,7618	1,850938468	1,8	2,829914893
5	0,7657	1,860414262	1,8	3,356347905
6	0,7674	1,864544737	1,8	3,585818705
7	0,7558	1,836360323	1,8	2,020017953
8	0,7616	1,85045253	1,8	2,802918329
9	0,7551	1,83465954	1,8	1,925529976
10	0,7612	1,849480654	1,8	2,748925199
11	0,7657	1,860414262	1,8	3,356347905
12	0,7565	1,838061107	1,8	2,114505929
13	0,7613	1,849723623	1,8	2,762423482
14	0,7569	1,839032983	1,8	2,168499058
15	0,7643	1,857012695	1,8	3,167371952

Tabla 11. Recolección de datos y cálculo del error cometido para 1.8 m test baja luminosidad Modo Noche

RESULTADOS

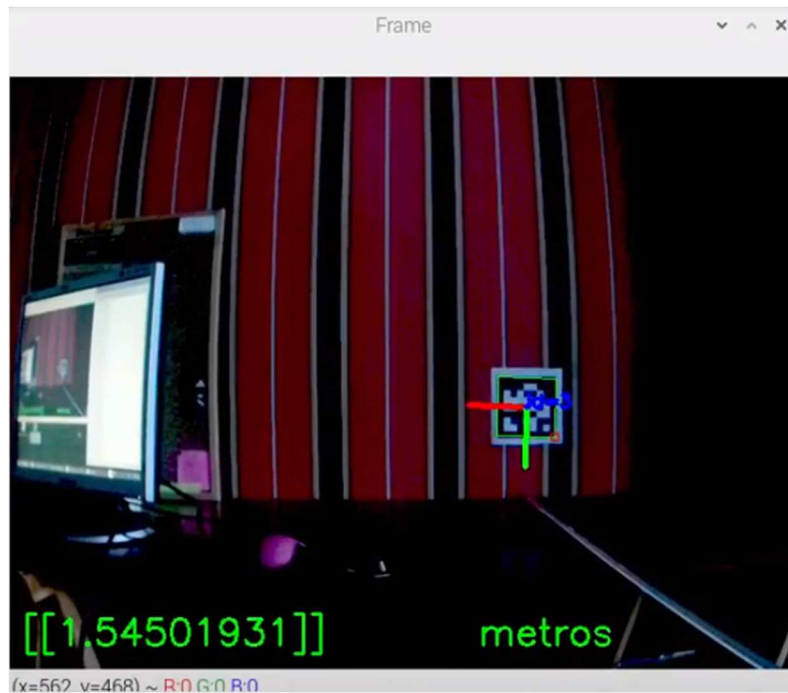


Figura 59. Ensayo a 1.5 m test baja luminosidad Modo Noche

Muestra para 1,5 metro test baja luminosidad Modo Noche				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,6393	1,553301342	1,5	3,553422827
2	0,6331	1,538237259	1,5	2,549150621
3	0,6378	1,549656806	1,5	3,310453745
4	0,6385	1,55135759	1,5	3,423839316
5	0,6339	1,540181012	1,5	2,678734131
6	0,6344	1,541395857	1,5	2,759723825
7	0,6349	1,542610703	1,5	2,840713519
8	0,6283	1,526574743	1,5	1,771649558
9	0,6281	1,526088805	1,5	1,73925368
10	0,6401	1,555245095	1,5	3,683006337
11	0,6279	1,525602867	1,5	1,706857802
12	0,6356	1,544311486	1,5	2,954099091
13	0,6399	1,554759157	1,5	3,65061046
14	0,6299	1,530462249	1,5	2,030816579
15	0,6241	1,516370042	1,5	1,091336127

Tabla 12. Recolección de datos y cálculo del error cometido para 1.5 m test baja luminosidad Modo Noche

RESULTADOS

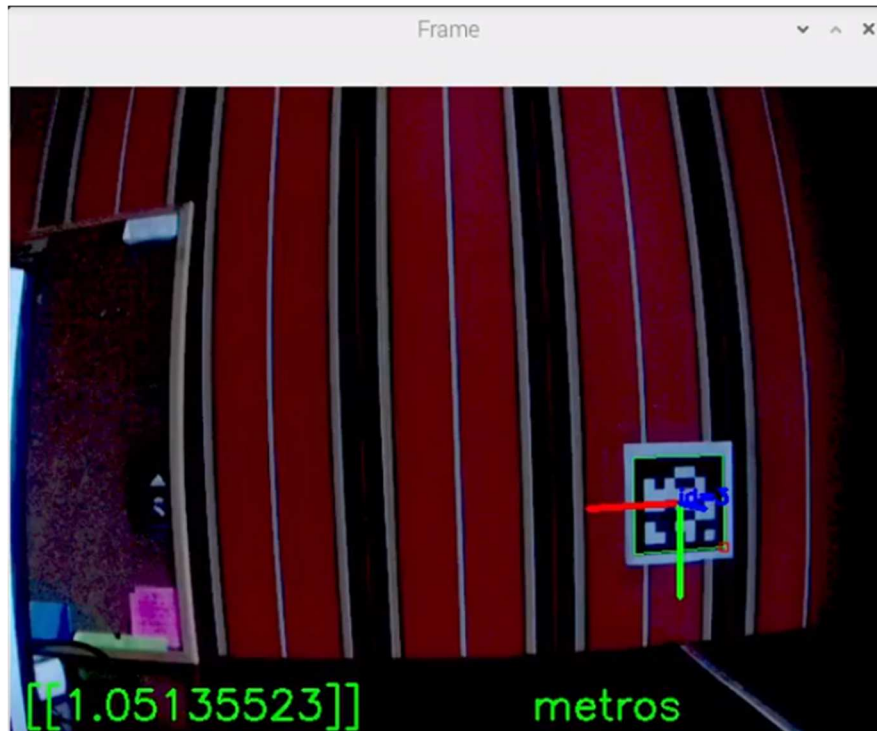


Figura 60. Ensayo a 1 m test baja luminosidad Modo Noche

Muestra para 1 metro test baja luminosidad Modo Noche				
N	Valor Profundidad	Valor Profundidad (m)	Medida Teórica (m)	Error Cometido (%)
1	0,4392	1,067120209	1	6,712020895
2	0,4375	1,062989735	1	6,298973456
3	0,4332	1,052542064	1	5,254206402
4	0,4334	1,053028002	1	5,302800219
5	0,432	1,049626435	1	4,962643504
6	0,4315	1,04841159	1	4,841158963
7	0,4328	1,051570188	1	5,157018769
8	0,4315	1,04841159	1	4,841158963
9	0,4309	1,046953775	1	4,695377513
10	0,4316	1,048654559	1	4,865455871
11	0,4308	1,046710806	1	4,671080605
12	0,4363	1,060074106	1	6,007410557
13	0,4345	1,055700662	1	5,570066209
14	0,4433	1,077081941	1	7,708194132
15	0,4341	1,054728786	1	5,472878576

Tabla 13. Recolección de datos y cálculo del error cometido para 1 m test baja luminosidad Modo Noche

RESULTADOS

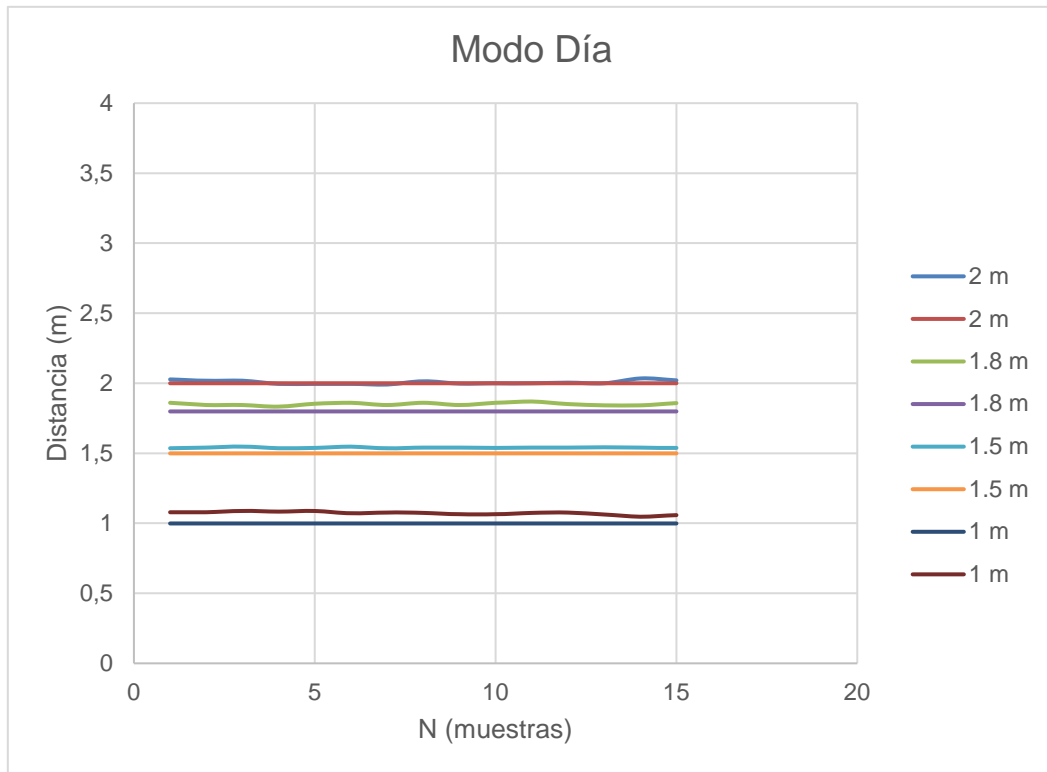


Figura 61. Comportamiento de la estimación de la distancia del prototipo frente a la distancia referencia en Modo Día

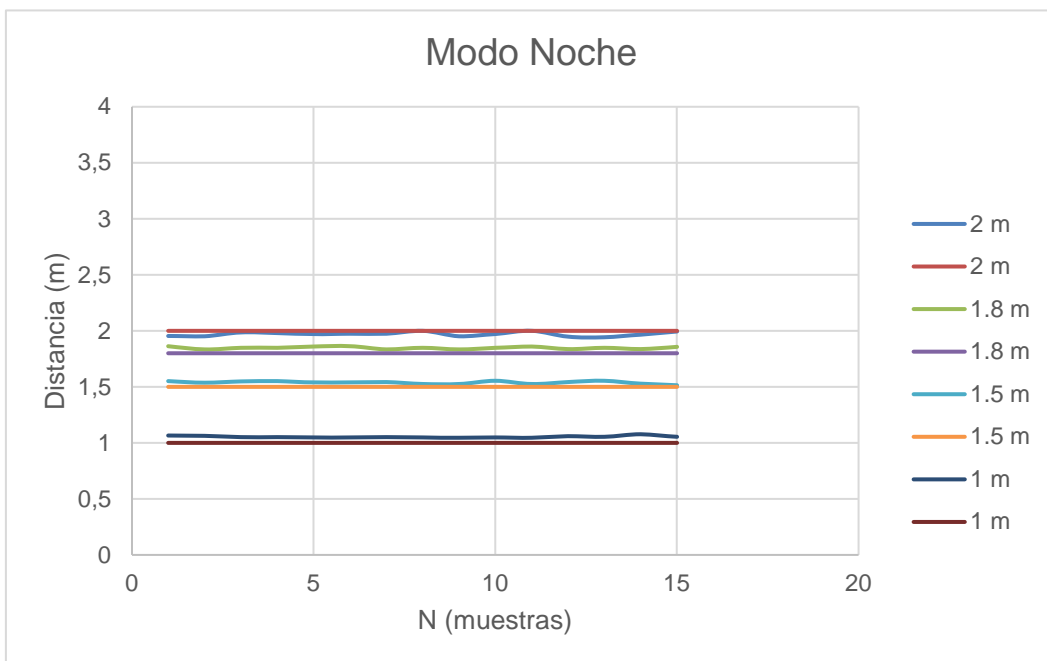


Figura 62. Comportamiento de la estimación de la distancia del prototipo frente a la distancia referencia en Modo Noche

RESULTADOS

Tal y como se puede observar en las figuras 61 y 60, se puede llegar a la conclusión de que, en situaciones de baja luminosidad, como puede ser el término de la tarde en zonas de sombra, el funcionamiento del Modo Noche cumple su objetivo y promete unos resultados mejores a la hora conseguir que el prototipo logre su función.

6 - PRESUPUESTO

Este capítulo está destinado para agrupar los precios de cada uno de los elementos que ha supuesto la creación de este trabajo. En su mayoría, los elementos han sido financiados por el alumno, salvo el soporte metálico, que ha sido cortesía de la empresa TColodro, en la que el alumno ha participado en varios proyectos. Otro elemento que ha tenido un coste nulo, se trata de la carcasa de tela, puesto que ha sido realizada con material reciclado.

A continuación, el desglose de precios y el precio total, visibles en la figura siguiente.

PRODUCTO	RESUMEM	CANTIDAD (u)	PRECIO (€)
Raspberry Pi 4B	Placa mini ordenador programable	1	55,9
Raspberry Pi HQ Camera	Cámara de alta calidad para RPi 4B	1	55,95
Lente Fija 6 mm	Lente acoplable a la RPi HQ Camera	1	28,95
Batería UNIROI	Batería de 3800 mAh y 10 W	1	19,99
SanDisk Extreme	Micro SD 64 GB	1	15,39
Disipador de calor	Carcasa disipadora de calor doble ventilador	1	17,99
Soporte Metálico	Placa 1,5 mm soporte lente	2	0
Carcasa de tela	Carcasa de tela que carga con el prototipo	1	0
		TOTAL	194,17

Tabla 14. Presupuestado del proyecto

El precio final para el proyecto DESARROLLO DE UNA APLICACIÓN BASADA EN VISIÓN POR COMPUTADOR PARA ASISTENCIA A INIDENTES tiene un valor de CIENTO NOVENTA Y CUATRO CON DIECISIETE EUROS.

7 - CONCLUSIONES Y TRABAJOS FUTUROS

Es notorio que el número de aplicaciones relacionadas con la visión por computador va en aumento. Por esta razón veo conveniente tener un conocimiento importante en esta área, que, tras la realización de este TFG, no es una tarea que sea sumamente tediosa si se poseen ciertos conocimientos previos en programación y electrónica.

Tras el resultado de los experimentos realizados, tanto en interior y en exterior, otorgan el siguiente veredicto. La aplicación posee una gran calidad para poder estimar la posición del marcador en ámbitos de una luminosidad adecuada. Esto quiere decir, que puede presentar problemas de detección en horarios nocturnos, si el marcador no está acompañado de una suficiente luz artificial. Además, no es tan preciso para detectar los distintos marcadores a la distancia de dos metros, sino que se necesitaría una distancia menor para poder detectar con calidad el marcador.

A pesar de estos inconvenientes, el prototipo tiene una amplia variedad de mejora, como la incorporación de una cámara nocturna u otro dispositivo que permita trabajar mejor en la oscuridad.

Otra limitación del prototipo es que solo se puede detectar un punto de interés a la vez, pero se podría solventar utilizando un algoritmo que incluyese un entorno de simultaneidad a la hora de realizar las detecciones.

Otra mejora, también se podría dar con el uso del módulo Haar Cascade, que ofrece la librería de OpenCV, y que resultaría bastante interesante de implementar para poder detectar objetos de interés para las personas invidentes, como lo pueden ser las monedas y los billetes.

8 - BIBLIOGRAFÍA

- *S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez. 2014. "Automatic generation and detection of highly reliable fiducial markers under occlusion"*
- *Packt Raspberry Pi Computer Vision Programming 2nd Edition*