



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

**DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO PARA
PERSONAS NO VIDENTES COMO AYUDA PARA EL CRUCE DE LAS
CALLES URBANAS BASADO EN EL PROCESAMIENTO DE IMÁGENES.**

TRABAJO DE TITULACIÓN
TIPO: DISPOSITIVO TECNOLÓGICO
Presentado para optar al grado académico de:
INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES

AUTOR: OSCAR DANIEL SEGURA MEDRANDA

TUTOR: PABLO EDUARDO LOZADA YÁNEZ

Riobamba – Ecuador

2019

©2019, Segura Medranda Oscar Daniel

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERIA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal de Titulación certifica que: El Dispositivo Tecnológico: DISEÑO Y CONSTRUCCIÓN DE UN SISTEMA ELECTRÓNICO PARA PERSONAS NO VIDENTES COMO AYUDA PARA EL CRUCE DE LAS CALLES URBANAS BASADO EN EL PROCESAMIENTO DE IMÁGENES, de responsabilidad del señor Oscar Daniel Segura Medranda, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

FIRMA

FECHA

Ing. Washington Luna

**DECANO DE LA FACULTAD DE
INFORMÁTICA Y ELECTRÓNICA**

Ing. Freddy Chávez V.

**DIRECTOR DE ESCUELA DE INGENIERÍA
ELECTRÓNICA EN CONTROL Y
REDES INDUSTRIALES**

Ing. Pablo Lozada Y.

**DIRECTOR DEL TRABAJO DE
TITULACIÓN**

Dr. Geovanny Vallejo V.

**MIEMBRO DEL TRABAJO DE
TITULACIÓN**

Yo, Oscar Daniel Segura Medranda soy responsable de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación y el patrimonio intelectual del Trabajo de Titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

Oscar Daniel Segura Medranda

DEDICATORIA

El desarrollo de este trabajo de titulación está dedicado a Dios por darme la bendición de seguir en pie durante toda mi formación académica y culminar mi meta de ser un profesional.

A mis queridos padres Lizardo e Isabel quienes fueron mi pilar y ese gran apoyo incondicional y mi guía en el transcurso de mi formación profesional.

A mis hermanos, hermana, demás familiares y amigos que me ofrecieron ese apoyo con sus palabras, su apoyo moral creyendo en la capacidad de alcanzar esta meta.

Oscar Segura

AGRADECIMIENTO

En primer lugar, a Dios que por medio de su bendición me ha dado la perseverancia para seguir en este camino hasta poder culminar mi meta y ver alcanzado este gran éxito.

A mi ejemplo más grande mi vida que son mis padres, por el apoyo incondicional y su enorme confianza que desde el comienzo de este camino me han brindado para que por medio de este logro alcanzado se sientan orgullosos.

A mis hermanos y hermana Jomaira, Bladimir, Lizardo, Pamela, Mariana Y Hector Segura que siempre me han dado su apoyo, ofreciéndome su consejo alentando el deseo de culminar mi carrera.

A mis familiares, amigos y compañeros Ángela, Cristina, Edwin y Alexander, quienes me brindaron su apoyo incondicional para poder culminar este camino y alcanzando mí meta.

A mi tutor de tesis Ing. Pablo Lozada y al Dr. Geovanny Vallejo miembro del tribunal, docentes de la Escuela Superior Politécnica de Chimborazo en la Facultad de Informática y Electrónica, gracias a su guía, sus conocimientos y consejos en el transcurso del desarrollo de mi trabajo de titulación que fueron de gran ayuda para la culminación del mismo.

Oscar Segura

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	xi
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE ABREVIATURAS.....	xv
ÍNDICE DE ANEXOS.....	xvi
RESUMEN.....	xvii
SUMMARY	xviii
INTRODUCCIÓN	1

CAPITULO I

1. MARCO REFERENCIAL	7
1.1. Discapacidad visual.	7
<i>1.1.1. Clasificación de la Discapacidad Visual.....</i>	<i>7</i>
<i>1.1.1.1. Visión normal.</i>	<i>8</i>
<i>1.1.1.2. Discapacidad visual moderna.</i>	<i>8</i>
<i>1.1.1.3. Discapacidad visual grave.</i>	<i>8</i>
<i>1.1.1.4. Ceguera.</i>	<i>8</i>
1.2. Orientación y Movilidad.	8
<i>1.2.1. Orientación.</i>	<i>9</i>
<i>1.2.2. Movilidad.....</i>	<i>9</i>
1.3. Herramientas de navegación para invidentes.	9
<i>1.3.1. Bastón.....</i>	<i>9</i>
<i>1.3.2. Ayudas electrónicas.</i>	<i>9</i>
<i>1.3.2.1 Lazzus.</i>	<i>10</i>
<i>1.3.2.2. SeeLight.....</i>	<i>10</i>
<i>1.3.2.3 Dispositivo de ayuda para los ciegos UCSU.</i>	<i>11</i>
<i>1.3.2.4 HALO: “Sistema de retroalimentación para ciegos con discapacidad visual”.....</i>	<i>12</i>
<i>1.3.2.5 Dispositivo de navegación para invidentes basado en la tecnología time of flight.</i>	<i>13</i>
1.3. Señales de tránsito.	14
<i>1.3.2. La vía pública.....</i>	<i>14</i>
<i>1.3.3. Peatones.</i>	<i>14</i>
<i>1.3.4. Colores de las señales de tránsito.....</i>	<i>15</i>

1.3.5.	Señales luminosas.....	16
1.3.5.5.	Semáforos.....	16
1.3.5.6.	Semáforos peatonales.....	16
1.3.5.7.	Semáforo peatonal rojo fijo.....	17
1.3.5.8.	Semáforo peatonal verde.....	17
1.4.	Sistemas embebidos.....	19
1.4.2.	Componentes.....	19
1.4.2.5.	Hardware.....	20
1.4.2.6.	Software.....	20
1.4.2.7.	Sistemas operativos embebidos.....	20
1.5.	Visión artificial.....	21
1.5.2.	Arquitectura de un sistema de visión artificial.....	22
1.5.2.5.	Dispositivos de captura de imágenes.....	22
1.5.2.6.	Digitalización de una imagen.....	23
1.5.3.	Aplicaciones de la visión artificial.....	23
1.5.4.	Representación de una imagen.....	24
1.5.5.	Segmentación.....	25
1.5.5.5.	Binarización.....	25
1.5.5.6.	Detección de contornos.....	26
1.5.6.	Operaciones morfológicas.....	26
1.5.6.5.	Erosión.....	26
1.5.6.6.	Dilatación.....	27
1.6.	Software para el desarrollo del sistema.....	28
1.6.2.	Open CV.....	28
1.6.2.5.	OpenCV – Python.....	28
1.7.	Hardware para el desarrollo del sistema embebido.....	29
1.7.2.	Raspberry Pi.....	29
1.7.2.1.	Características principales de las tarjetas de desarrollo.....	29
1.7.3.	Módulo de cámara.....	30
1.7.4.	Batería Portátil.....	31

CAPÍTULO II

2	METODOLÓGICO.....	32
2.1	Arquitectura general del sistema.....	32

2.2	Diagramas de bloques del procesamiento de las imágenes.	33
2.3	Requerimiento funcional del prototipo.	34
2.4	Requerimientos del sistema:	34
2.5	Selección de los elementos electrónicos para la construcción del prototipo.	34
2.5.1	<i>Elementos hardware del prototipo.</i>	34
2.5.1.1	<i>Selección de dispositivo de captura.</i>	34
2.5.1.2	<i>Tarjeta de desarrollo Raspberry pi 3.</i>	36
2.5.1.3	<i>Alimentador portátil.</i>	37
2.5.1.4	<i>Memoria micro SD.</i>	39
2.5.1.5	<i>Auriculares.</i>	40
2.5.1.6	<i>Esquema de conexión electrónico.</i>	40
2.6	Requerimiento para el diseño del algoritmo.	41
2.6.1	<i>Instalación de Programas en la Raspberry Pi 3.</i>	41
2.6.1.1	<i>Sistema operativo Raspbian Stretch.</i>	41
2.6.1.2	<i>Características del sistema operativo Raspbian Stretch.</i>	41
2.6.1.3	<i>Instalación del sistema operativo en la Raspberry PI 3.</i>	42
2.6.1.4	<i>Descarga e instalación de Python 3.0 y librerías para uso de imágenes y video.</i>	43
2.6.2	<i>Programación del prototipo.</i>	43
2.6.2.1	<i>Captura de imagen.</i>	44
2.6.3	<i>Selección de librerías para el desarrollo del programa.</i>	44
2.6.4	<i>Detección de líneas paralelas, semáforo y paso cebra.</i>	46
2.6.5	<i>Filtrado para colores primarios.</i>	46
1.3.7.	<i>Segmentación.</i>	47
2.6.5.1	<i>Filtro 1.</i>	48
2.6.5.2	<i>Filtro 2.</i>	48
2.6.5.3	<i>Filtro 3.</i>	49
2.6.6	<i>Detección del pare.</i>	49
2.6.6.1	<i>Filtrado de color rojo.</i>	50
2.6.7	<i>Comparación de base de datos.</i>	51
2.6.8	<i>Detección de semáforo rojo.</i>	51
2.6.8.1	<i>Creación de mascara para filtración de color.</i>	52
2.6.8.2	<i>Umbralización para color rojo.</i>	53
2.6.9	<i>Detección de semáforo en verde.</i>	54
2.6.9.1	<i>Análisis morfológico.</i>	55
2.6.10	<i>Detección de líneas laterales y paso cebra.</i>	55
2.6.10.1	<i>Creación de matrices interlineadas.</i>	56

2.6.10.2	<i>Relleno de espacio para reconstrucción de imagen</i>	57
2.6.10.3	<i>Filtrado por morfología</i>	57
2.6.11	<i>Diagramas de flujos para el reconocimiento de las señales peatonales</i>	59
2.6.11.1	<i>Detección de próximo a paso lateral, semáforo y paso cebra</i>	59
2.6.11.2	<i>Detección de pare y semáforos</i>	60
2.6.11.3	<i>Detección de líneas laterales y paso cebra</i>	60

CAPÍTULO III

3.	<i>MARCO DE RESULTADOS</i>	61
3.1.	<i>Tiempo de respuesta del algoritmo</i>	61
3.2.	<i>Desarrollo de pruebas del prototipo</i>	61
3.2.1.	<i>Experimento de validación del prototipo implementado</i>	62
3.2.1.1.	<i>Prueba de identificación de aproximación a paso cebra</i>	62
3.2.1.2.	<i>Prueba de identificación del pare</i>	63
3.2.1.3.	<i>Prueba de identificación de aproximación a semáforo</i>	64
3.2.1.4.	<i>Prueba de identificación de aproximación a esquina</i>	65
3.2.1.5.	<i>Prueba de identificación de líneas paralelas</i>	66
3.2.1.6.	<i>Prueba de identificación del paso cebra</i>	67
3.2.1.7.	<i>Prueba de identificación del semáforo peatonal en rojo</i>	68
3.2.1.8.	<i>Prueba de identificación del semáforo peatonal en verde</i>	69
3.3.	<i>Análisis de confiabilidad del sistema</i>	70
3.4.	<i>Costo del prototipo</i>	75
	<i>CONCLUSIONES</i>	76
	<i>RECOMENDACIONES</i>	77

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1:	Tipos de discapacidades en Ecuador.....	3
Tabla 2:	Personas con discapacidades en Riobamba, Chimborazo.....	3
Tabla 3-1:	Normalización de colores del ANT.....	15
Tabla 4-1:	Señales usadas para la programación	17
Tabla 5-1:	Aplicaciones de la visión artificial.....	23
Tabla 2-2:	Comparación entre tarjetas de desarrollo.	30
Tabla 1-2:	Características de cámara WideCam F100.....	35
Tabla 3-2:	Características Tarjeta de desarrollo Raspberry pi 3.....	36
Tabla 4-2:	Características de la batería portátil.....	38
Tabla 5-2:	Características de la batería portátil.....	39
Tabla 1-3:	Identificación de aproximación a paso cebra.	62
Tabla 2-3:	Identificación del pare.	63
Tabla 3-3:	Identificación de aproximación a semáforo.....	64
Tabla 4-3:	Identificación de peatones en la vía.....	65
Tabla 5-3:	Identificación de líneas paralelas.....	66
Tabla 6-3:	Identificación del paso cebra.	67
Tabla 7-3:	Identificación del semáforo peatonal en rojo.	68
Tabla 8-3:	Identificación del pare.	69
Tabla 9-3:	Nivel de confianza.....	70
Tabla 10-3:	Datos obtenidos del funcionamiento del sistema.....	71
Tabla 11-3:	Costos directos de fabricación del prototipo electrónico.....	75

ÍNDICE DE FIGURAS

Figura 1: Prototipo a implementar	5
Figura 1-1: Aplicación móvil del SeeLight.....	11
Figura 2-1: Aplicación móvil.....	12
Figura 3-1: Usuario utilizando el dispositivo HALO.....	13
Figura 4-1: Gráfica del dispositivo	14
Figura 5-1: Acera y Calzada.	15
Figura 6-1: Semáforo peatonal.	16
Figura 7-1: Semáforo peatonal en rojo	17
Figura 8-1: Semáforo peatonal en verde	17
Figura 9-1: Proceso de un sistema de visión artificial	21
Figura 10-1: Espectro electromagnético	22
Figura 11-1: Sistema coordinado de una imagen	24
Figura 12-1: Vecindad a cuatro de un píxel.....	24
Figura 13-1: Binarización	26
Figura 14-1: Erosión de una imagen binaria.....	27
Figura 15-1: Dilatación de una imagen binaria.....	27
Figura 16-1: Raspberry Pi.....	29
Figura 17-1: Características Raspberry Pi	29
Figura 18-1: Cámara Raspicam	31
Figura 19-1: Batería Portátil	31
Figura 1-2: Arquitectura general del sistema.....	32
Figura 2-2: Bloques del procesamiento de las imágenes	33
Figura 3-2: Cámara WideCam F100.....	35
Figura 4-2: Comparación de imágenes izquierda f 100 derecha cámara Raspberry	36
Figura 5-2: Tarjeta de desarrollo Raspberry pi 3	36
Figura 6-2: Alimentador portátil.....	39
Figura 7-2: Alimentador portátil.....	40
Figura 8-2: Auriculares.....	40
Figura 9-2: Imagen a descargar del SO Raspbian.....	42
Figura 10-2: Programa Win32 Disk Imager	42
Figura 11-2: Escritorio de la Raspberry Pi 3	43
Figura 12-2: Comando para captura	44
Figura 13-2: OpenCV y Python.	45
Figura 14-2: NumPy, scipy scikit image	45
Figura 15-2: Filtro de color HSV	46

Figura 16-2: Filtro de RGB invertido	47
Figura 17-2: Resta de colores invertidos	47
Figura 18-2: Segmentación	48
Figura 19-2: Segmentación total	48
Figura 20-2: Base de datos.....	49
Figura 21-2: Filtros de color	50
Figura 22-2: Símbolo del pare	50
Figura 23-2: Creación de base del pare	51
Figura 24-2: Resultado obtenido.....	51
Figura 25-2: Resultado obtenido.....	52
Figura 26-2: Creación de mascar	52
Figura 27-2: Umbralización color rojo	53
Figura 28-2: Filtración por morfología	53
Figura 29-2: Resultado obtenido.....	54
Figura 30-2: Color verde en formato HSV	54
Figura 31-2: Análisis por morfología	55
Figura 32-2: Interlineado horizontal	56
Figura 33-2: Interlineado vertical	57
Figura 34-2: Reconstrucción de espacios	57
Figura 35-2: Filtrado pro morfología	58
Figura 36-2: Reconstrucción total.....	58
Figura 37-2: Resultado obtenido.....	59
Figura 1-A: Descargar NOOBS en el PC	
Figura 2-A: Pasos a seguir para instalar NOOBOS en la tarjeta SD	
Figura 3-A: Pasos a seguir para instalar NOOBOS en la tarjeta SD	
Figura 4-A: Menú con los sistemas operativos disponibles para instalar	
Figura 5-A: Proceso de instalación del sistema operativo	
Figura 6-A: Culminación de instalación del sistema operativo	
Figura 7-A: Menú de arranque del sistema operativo	
Figura 1-B: Prototipo	
Figura 2-B: Pruebas de funcionamiento	
Figura 3-B: Pruebas de funcionamiento y audio	
Figura 4-B: Pruebas de distancia	

ÍNDICE DE GRÁFICOS

Gráfico 1: Personas con discapacidades por provincia	2
Gráfico 1-2: Detección de próximo a paso lateral, semáforo y paso cebra.....	59
Gráfico 2-2: Detección de pare y semáforos.....	60
Gráfico 3-2: Detección de líneas laterales y paso cebra	60
Gráfico 1-3: Porcentaje de funcionamiento del sistema.	74

ÍNDICE DE ABREVIATURAS

A	Amperio
BGR	Azul Verde Rojo
CA	Corriente alterna
CD	Corriente directa
CONADIS	Consejo Nacional para la Igualdad de Discapacidades
ETHERNET	Estándar de redes de área local
HARDWARE (HW)	Parte física de un sistema embebido
Mm	Milímetros
OMS	Organización Mundial de la Salud
OPENCV	Open Source Computer Vision
RAM	Memoria de acceso aleatorio
RSPICAM	Cámara de tarjeta raspberry Pi
SOFTWARE (SF)	Parte intangible de un sistema embebido
USB	Universal Serial Bus
V	Voltios

ÍNDICE DE ANEXOS

- Anexo A:** Dimensiones de la Raspberry Pi 3
- Anexo B:** Paso a paso la instalar del sistema operativo en la Raspberry Pi
- Anexo C:** Fotografías del prototipo y de las pruebas realizadas

RESUMEN

En el presente trabajo de titulación se propuso, el diseño y construcción de un sistema electrónico para personas invidentes como una ayuda al momento de desplazarse por las calles urbanas de la ciudad. Dentro del trabajo planteado se espera que sea de ayuda en el ámbito social e inclusivo para personas invidentes, en el desenvolvimiento de su vida cotidiana, por medio de la tecnología y la visión artificial. Por medio del procesamiento de imágenes se realizó el sistema inmerso en un prototipo. En el primero modulo se realizó la captura de imágenes las cuales pasará a ser procesadas. En la etapa dos se hará el procesamiento de las imágenes, es decir mediante un algoritmo y determinar el tipo de señales que se tienen y por último para finalizar el proceso se enviar un mensaje al usuario indicando la señal de transito que se tiene en la imagen. Los módulos se comunican de manera alámbrica, este proceso se realiza en una tarjeta Raspberry Pi 3. El prototipo implementado, permite capturar imágenes en las vías urbanas para determinar qué tipo de señal peatonal existe dentro de la imagen capturada por la cámara, si existe alguna señal peatonal en la imagen se emitirá un mensaje de voz hacia el usuario, en base a esa información, el usuario podrá tomar decisiones al momento de cruzar la calle y pueda hacerlo de manera más segura. El prototipo consta de tres etapas o módulos que se describen a continuación. Se evaluó tomando 20 pruebas, de las diferentes señales peatonales en el prototipo, de lo cual se obtuvo una media de 77% de efectividad. Se concluye, que es un prototipo capaz de cumplir con las expectativas establecidas para mejorar el estilo de vida de personas con la discapacidad.

PALABRAS CLAVES: <VISIÓN ARTIFICIAL>, <DISCAPACIDAD VISUAL>, <SEÑALES PEATONALES>, <RECONOCIMIENTO DE IMÁGENES>, <PROCESAMIENTO DE SEÑALES>, <RASPERRY PI 3 (SOFTWARE - HARDWARE)>, <SISTEMAS EMBEBIDOS>.

SUMMARY

The following graduation projects propose the design and construction of an electrical system for blind people like help at the moment of moving by the urban streets of the city. Within the planned work is expected to be helpful in the social field and even for blind people, in the development of their daily lives, through technology and artificial vision. Through the processing of images, the system was immersed in a prototype In the first module the capture of images of which will be processed was carried out. In stage two the processing of the images will be done, In order word, by means of an algorithm and determine the type of signals that have and finally to finish the process a message will be sent to the user indicating the traffic signal that is in the image. The modules communicate in a wired way; this process is carried out on a Raspberry Pi 3 card. The prototype implementing, allows to capture images on urban roads to determine what type of pedestrian signal exists within the image captured by the camera, if there is any pedestrian signal in the image a voice message will be issued to the user, based on that information, the user can make decisions when crossing the street and can do so more safely. The prototype consists of three stages or modules that are described below. It was evaluated taking 20 tests, of the different pedestrian signals in the prototype, of which an average of 77% of effectiveness was obtained. It is concluded that it is a prototype able to meet the established expectations to improve the lifestyle of people with disabilities.

KEY WORDS: <ARTIFICIAL VISION>, <VISUAL DISABILITY>, <PEDESTRIAN SIGNALS>, <IMAGE RECOGNITION>, <SIGNAL PROCESSING>, <RASPBERRY PI 3 (SOFTWARE- HARDWARE)>, <EMBEDDED SYSTEMS>.

INTRODUCCIÓN

ANTECEDENTES

De acuerdo con la Organización Mundial de la Salud (OMS), se estima que a nivel mundial 285 millones de personas tienen deficiencias visuales, de las cuales 39 millones tienen ceguera y 246 millones son débiles visuales. De estas, el 90 % viven en países en desarrollo Según la OMS (SALUD, 2015)

La atención a las personas con discapacidad en el Ecuador se ha caracterizado como en los problemas sociales, por ser de baja cobertura y deficiente Ecuador tiene 14,8% de discapacitados, según OEA en el Ecuador existen más de 200.000 ecuatorianos no videntes y deficientes visuales según el conadis, En el cantón de Riobamba de acuerdo con el conadis existen 769 personas con deficiencia visual. (ORGANIZACIÓN, 2017)

Las personas con ceguera total o con poca visión usualmente tienen problemas para manejarse fuera de entornos conocidos, por esta razón, muchas personas con poca visión caminan junto a un amigo o familiar que los ayude a conducirse en entornos desconocidos. Esto provoca un problema de inseguridad para estas personas, pueden originar accidentes de tránsito (OEA 2009).

La tecnología remota de Aíra utiliza lentes inteligentes para conectar a aquellas personas con visión reducida a una red de agentes certificados. Los agentes pueden “ver” desde la perspectiva del usuario utilizando video en casi tiempo real y luego comunicarle información e instrucciones al usuario. Esto ayuda a que el usuario logre hacer tareas cotidianas y retos nuevos. Según Aíra, los lentes inteligentes pueden ayudar a los usuarios a transitar por calles congestionadas, utilizar el transporte público, comprar en tiendas, o incluso escalar una montaña. (INFOTECNOVISION, 2017)

De acuerdo a la información dada por el CONADIS y la ASOCIACION DE PERSONAS CON DISCAPACIDAD FISICA DE CHIMBORAZO precedida por el Sr. Luis Palacios las personas con discapacidad visual tienen problemas al cruzar las calles por falta de instalación de semáforos de tres tiempos, impiden a las personas ciegas saber cuándo deben cruzar una calle, la falta de señalización en el ciclo vías, veredas y calzadas en mal estado y entre otras necesidades.

Dado a nivel del país no se han desarrollado este tipo de prototipos se desea realizar el tema propuesto que es el diseño y construcción de un sistema electrónico para personas no videntes como ayuda para el cruce de las calles urbanas basado en el procesamiento de imágenes.

FORMULACIÓN DEL PROBLEMA

¿Cómo diseñar y construir un sistema electrónico para personas no videntes como ayuda para el cruce de las calles urbanas basado en el procesamiento de imágenes?

SISTEMATIZACIÓN DEL PROBLEMA

¿Cómo será el funcionamiento del prototipo para personas invidentes?

¿Cuáles son los requerimientos de diseño que debe cumplir el prototipo a implementar?

¿Cuáles herramientas hardware y el software que cumple con el diseño propuesto?

¿El prototipo cumplirá con los requerimientos de este trabajo de investigación?

JUSTIFICACIÓN TEÓRICA

Las estadísticas de discapacidades en Ecuador como se muestra en la Gráfica 1, se revelan por provincias, donde la mayoría de personas con discapacidades se encuentran en las provincias del Guayas con un aproximado de 97703 individuos, después seguido de la provincia de Pichincha con un aproximado de 62494 individuos. Dado que el trabajo de titulación está enfocado a la provincia de Chimborazo, donde detalla 13911 individuos con algún tipo de discapacidad es viable realizar este tipo de trabajos de titulación.

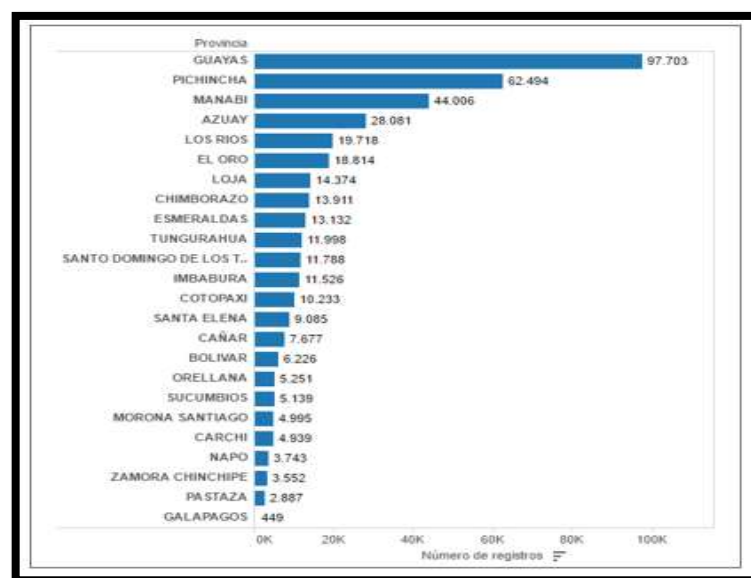


Gráfico 1: Personas con discapacidades por provincia

Fuente: CONADIS, 2008.

Los tipos de discapacidades en el Ecuador de mayor grado como se muestra en la Tabla 1, evidencia la discapacidad física como la de mayor grado con un aproximado de 195.046 personas, dentro de la Tabla 1 se evidencia estos valores.

Tabla 1: Tipos de discapacidades en Ecuador.

TIPO DISCAPACIDAD ACTUAL EN ECUADOR	
TIPO	NUMERO DE INDIVIDUOS
AUDITIVA	52.508
FISICA	195.046
INTELECTUAL	92.121
LENGUAJE	5.615
PSICOLOGICO	7.560
PSICOSOCIAL	10.355
VISUAL	48.516
TOTAL, GENERAL	411.721

Fuente: (CONADIS). 2008.

En la siguiente Tabla 2, se evidencia el número de individuos con discapacidades en Riobamba perteneciente a la provincia de Chimborazo; se toma como principal campo las discapacidades visuales ya que en ellas se encuentra las discapacidades motrices superiores, que son a las que va dirigido el trabajo de titulación.

Tabla 2: Personas con discapacidades en Riobamba, Chimborazo.

Provincia	Cantón	Tipo Discapacidad	FEMENI NO	MASCULINO	Total, general
CHIMBORAZO	RIOBAMBA	AUDITIVA	626	653	1.279
		FISICA	1449	1.460	2.909
		INTELECTUAL	570	675	1.245
		LENGUAJE	29	34	63
		PSICOLOGICO	13	26	39
		PSICOSOCIAL	65	70	135
		VISUAL	335	382	717
Total, general			3.087	3.300	6.387

Fuente: (CONADIS, 2008)

Estas personas encuentran una serie de dificultades debido a que la mayoría de ambientes no son aptos para ellas. Entonces, surge la necesidad de dar solución a esta problemática mediante el desarrollo de aplicaciones basadas en metodologías y técnicas como la visión artificial que en los últimos años ha servido para la creación de sistemas de apoyo y soporte a personas con deficiencias visuales.

La visión artificial como sustituto de la visión humana es una herramienta importante en el desarrollo de dispositivos de apoyo a personas ciegas y débiles visuales. Entre las tareas para las cuales se ha usado la visión artificial para apoyo a personas invidentes con resultados prometedores en los que se incluye: movilidad, orientación, reconocimiento de objetos, acceso a acceso a información impresa e interacción social. (CENATAV 2017).

En este proyecto de investigación, se propone el diseñar y construir un sistema electrónico para personas no videntes como ayuda para el cruce de las calles urbanas basado en el procesamiento de imágenes. que sea de bajo costo en comparación a otros prototipos como por ejemplo una tesis uso métodos de visión artificial para PC como apoyo en la automoción que costó 5500 dólares que se podría realizar otras metodologías y otros recursos con los cuales se pueden abaratar costos.

JUSTIFICACIÓN APLICATIVA

La implementación de este prototipo aportará de manera especial a las personas invidentes, en el cruce de las calles urbanas dado que es un problema muy preocupante en estos días por el incremento del parque automotor en las ciudades, cada vez es más dificultoso el caminar por la acera y cruzar por la calzada y más aún para este tipo de personas con discapacidad visual. El prototipo ayudara a estas personas mediante mensajes auditivos es decir este prototipo, tendrá el siguiente proceso: mediante una cámara que ejecutara capturas de imágenes, las cuales serán analizadas con un algoritmo el cual reconoce señales peatonales como son: el pare, semáforo peatonal, intercepción de esquinas, aproximación a semáforo, peatones en la vía entre otras señales peatonales, si existe alguna señale peatonal dentro de la imagen capturada por la cámara que sea de interés y en el caso que se detecte una señal peatonal se emitirá un mensaje de voz mediante un auricular hacia el usuario diciéndole que señal fue detectada al instante. En la Figura 1 se muestra un esquema funcional del prototipo.

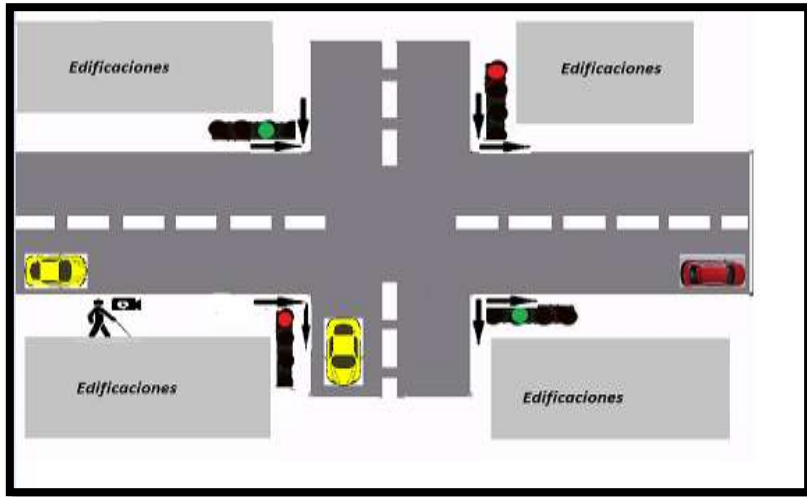


Figura 1: Prototipo a implementar
Realizado por: Oscar Segura, 2018.

A nivel del país se han desarrollado pocos de estos prototipos por ello es viable realizar este tipo de proyectos.

Objetivos.

Objetivos Generales.

- Diseño y construcción de un sistema electrónico para personas no videntes como ayuda para el cruce de las calles urbanas basado en el procesamiento de imágenes.

Objetivos Especificos.

- Definir como debe funcionar un dispositivo para invidentes que le ayude a identificar cuando puede cruzar las calles urbanas.
- Determinar cuáles son los requerimientos que debe cumplir el prototipo a implementar.
- Diseñar el sistema que cumple los requerimientos del prototipo.
- Establecer cuales son el hardware y el software que cumplan con el diseño.
- Implementar el prototipo propuesto para el trabajo de titulación
- Determinar si el prototipo cumplió los requerimientos especificados.

CAPITULO I

1. MARCO REFERENCIAL

En el presente capítulo se planea abarcar temáticas acorde al proyecto para entender las ideas básicas y generalidades de los elementos a tener en cuenta en el desarrollo del mismo, en el cual se definirán conceptos de importancia y definiciones, así como la discapacidad visual, orientación y movilidad para personas invidentes, las técnicas que estas personas usan para desplazarse por zonas urbanas, además los elementos que se usaran para el desarrollo de este prototipo como una tarjeta de desarrollo, las señales peatonales a identificar mediante un algoritmo, la identificación de estas señales de tránsito, entre otros conceptos que son necesarios para una mejor comprensión.

1.1. Discapacidad visual.

Es la condición o falta de capacidad de ejecutar una actividad o una acción en la misma circunstancia, se considera normal para una persona. Con arreglo de la Clasificación Internacional de Enfermedades, la función visual se subdivide en cuatro niveles:(CENATAV 2017)

La discapacidad visual moderada y la discapacidad visual grave se reagrupan comúnmente bajo el término “baja visión”; la baja visión y la ceguera representan al mismo tiempo el total de casos de discapacidad visual. (Universia, 2014 pág. 11)

1.1.1. Clasificación de la Discapacidad Visual.

Existen varias formas de clasificar a la discapacidad visual, sin embargo con arreglo a la Clasificación Internacional de Enfermedades CIE-10 la función visual se subdivide en:

- Visión normal.
- Discapacidad visual moderada.
- Discapacidad visual grave.
- Ceguera.

1.1.1.1. Visión normal.

Puede realizar tareas sin ayudas especiales, es decir que puede ver los objetos claramente sin necesidad de algún tipo de ayuda.

1.1.1.2. Discapacidad visual moderna.

Puede realizar tareas casi normales con ayudas especiales, no puede visualizar los objetos de manera normal por eso necesita algún tipo de ayuda como lentes.

1.1.1.3. Discapacidad visual grave.

Tiene dificultad para tareas visuales gruesas, no puede realizar tareas que exijan control de detalles.

1.1.1.4. Ceguera.

Es la ausencia de luz que no se puede corregir con gafas o lentes de contacto. Las principales causas de ceguera crónica son las cataratas, el glaucoma, la degeneración macular relacionada con la edad, las opacidades corneales, la retinopatía diabética, el tracoma y las afecciones oculares infantiles, como las causadas por la carencia de vitamina A. (Organización Mundial de la Salud, 2011)

1.2. Orientación y Movilidad.

De acuerdo con el Conadis la “Orientación y Movilidad es el área donde la persona con discapacidad visual adquiere las herramientas para su desplazamiento independiente. Es impartida por docentes especializados (en niños y niñas) o por instructores (en jóvenes y adultos). Es mucho más que el uso de un bastón y permite que la persona se desplace en forma autónoma, segura y elegante”. (PRONADIS, 2017)

1.2.1. Orientación.

La orientación implica la comprensión del ambiente, es decir, la toma de conciencia de la situación de la persona en el espacio y la capacidad para relacionarse espacialmente entre los elementos del ambiente. (VISUAL, 2016)

1.2.2. Movilidad.

La movilidad, o capacidad para desplazarse con independencia, seguridad y eficacia. Implica el aprendizaje de técnicas de protección (para interiores y exteriores) y de otras técnicas que permiten a la persona con discapacidad visual caminar en línea recta, seguir referencias, cruzar calles, y utilizar el transporte público. (VISUAL, 2016)

1.3. Herramientas de navegación para invidentes.

Hay una variedad de instrumentos para personas invidentes que les ayuda a desplazarse por el medio o entorno que los rodea, de una manera segura y práctica entre las cuales se tienen las siguientes:

1.3.1. Bastón.

Este mecanismo usualmente se usa para ubicar obstáculos y suministrar información a la persona, sobre el entorno que lo rodea al desplazarse en lugares cerrados o abiertos. Además, se identifica a la persona con una discapacidad visual. (Partners, 2015 pág. 5)

1.3.2. Ayudas electrónicas.

Son dispositivos electrónicos que ayudan u orientan a las personas con discapacidad visual a localizar e identificar las diferentes referencias y obstáculos a una persona ciega. Estos dispositivos se los conocen como ETAs, por sus siglas en inglés: *Electronic Travel Aids*. Estas tecnologías de apoyo tienen la finalidad de mejorar la orientación y movilidad de las personas ciegas durante su desplazamiento en los diferentes entornos que pueden ser cerrados o abiertos, además suelen ser conocidos o desconocidos. Se pueden encontrar diferentes dispositivos electrónicos usualmente están enfocados a una tarea específica (orientación, esquivar obstáculos,

acceder a información del medio que los rodeo u otro, etc.) Estos sistemas de navegación generalmente están provistos de: (HANDBOOK, 2018)

- Detección de obstáculos y peligros.
- Información de ubicación y orientación durante la navegación.
- Rutas óptimas hacia un destino deseado.

1.3.2.1 Lazzus.

Se trata de una herramienta con la que invidentes y personas con una alta discapacidad visual podrán conocer, a través de su dispositivo móvil, información sobre lo que tienen a su alrededor en cada momento.

Desarrollada para dispositivos Android, así como también para los móviles de Apple e incluso las Google Glass, Lazzus basa su tecnología en el GPS que se lleva metido en el bolsillo. El mismo que se convertirá en los ojos de los usuarios. El GPS del dispositivo móvil informa a Lazzus sobre la ubicación del usuario, mientras que la brújula del smartphone indica a la aplicación cuál es su orientación. A partir de ahí, la herramienta informa por medio de audios al invidente sobre los puntos de interés que hay en ese preciso instante en su campo visual. Todo ello en tiempo real.

El usuario puede recibir información de su localización exacta en ese momento, así como de los puntos de interés que están a su alrededor (por ejemplo, el nombre de los establecimientos) gracias al campo de visión artificial que crea la 'App'. Si desea dirigirse a uno de ellos, la aplicación le orientará. No existe descripción de los componentes tanto del hardware como del software usado. (Partners, 2015)

1.3.2.2. SeeLight.

La aplicación móvil funciona de tal manera que después que el usuario sea registrado en el servicio, puede completar un formulario que incluirá los datos relacionados con los pasos de peatones (dónde se colocan, si hay un semáforo, si hay sonidos, con qué frecuencia funcionan, entre otros parámetros. Cualquiera puede agregar semáforos al sistema con la API abierta.

Todos pueden usarlo: agregando los semáforos en el mapa, una vez que todas las entradas, la información será geolocalizada y cargada en un mapa interactivo que servirá como una guía para los ciegos para garantizar una mayor seguridad al caminar por la ciudad, y luego, hacer que la persona sea más independiente.

La aplicación anuncia en voz alta la dirección y la distancia al semáforo más cercano y notifica al usuario sobre el mismo en el modo de fondo. SeeLight se puede considerar como una aplicación social, ya que funciona con la ayuda de personas que realmente pueden ver e informar la presencia

de un cruce, proporcionando esta información a las personas con discapacidad visual. No existe descripción de los componentes del hardware como del software utilizado, en la Figura 1-2 se puede apreciar la imagen de la aplicación móvil. (CAVALIERI, 2015)



Figura 1-1: Aplicación móvil del SeeLight
Fuente: <http://seelight.hungryboys.ru/en/>

1.3.2.3 Dispositivo de ayuda para los ciegos UCSU.

El grupo de investigación de la universidad de Carlos III de Madrid (UC3M) ha diseñado un sistema de ayuda para los ciegos, la propuesta consiste en un procesador de estereovisión que, midiendo la diferencia de imágenes captadas por dos cámaras ligeramente desplazadas, calcula la distancia a cada punto de la escena.

Después, para transmitir esa información al usuario se emplea un código de sonidos que informa de la posición y distancia de los distintos obstáculos. La cantidad de información recibida de la escena depende del perfil cognitivo seleccionado. Se puede elegir entre "seis perfiles, desde uno muy sencillo, con alarma sonora sólo cuando te vas a chocar, a otros que describen la escena con sesenta y cuatro sonidos simultáneos", indica Revuelta. El sistema aún no está en una fase comercialización, porque de momento tan solo se ha construido un prototipo en la Figura 2-1 se muestra el prototipo. (Revuelta, y otros, 2014)



Figura 2-1: Aplicación móvil

Fuente: <https://www.xatakandroid.com>

1.3.2.4 HALO: “Sistema de retroalimentación para ciegos con discapacidad visual”

La empresa Polymythic, se dedica a la tecnología accesible que tuvo la idea de desarrollar un dispositivo de retroalimentación háptica para las personas con discapacidad visual.

Utiliza una serie de telémetros que toman datos de sensores y de la retroalimentación de salida provenientes de impulsos motores de vibración (sensores) que se encuentran colocados en una superficie de la cabeza de una persona. Cuando la persona se acerca a un objeto, la intensidad y la frecuencia de la vibración aumenta, siendo directamente proporcional a la distancia de un objeto. Si los sensores carecen de retroalimentación, entonces es seguro continuar en esa dirección.

El HALO está compuesto por un controlador Arduino Mega 2560 que funciona con una sola batería de 9V. Al ser un dispositivo muy útil, también tiene la ventaja de ser relativamente económico. Dado que se encuentra dentro del rango asequible, no solo permanecerá como un concepto innovador, sino que seguramente se configurará como un producto maravilloso y una bendición para todos los necesitados, en la Figura 3-1 se puede visualizar el prototipo HALO. (WALYOU, 2010)

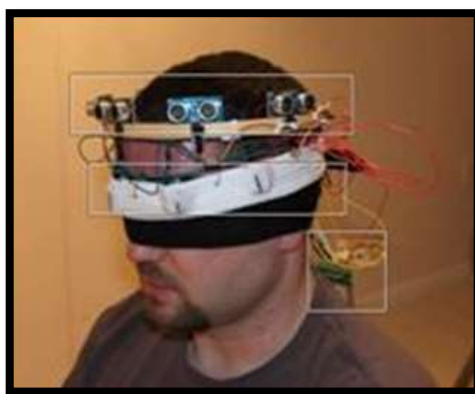


Figura 3-1: Usuario utilizando el dispositivo HALO

Fuente: <https://www.google.com.ec/search?q=halo+disponibles+para+ciegos&source:>

1.3.2.5 Dispositivo de navegación para invidentes basado en la tecnología time of flight.

En el siguiente resumen se hablará de un nuevo dispositivo de navegación y detección de obstáculos para las personas ciegas, basado en la tecnología Time-of-Flight y en sonidos acústicos. El dispositivo se ha desarrollado como un dispositivo de ayuda, complementario al bastón, para las personas invidentes.

Su objetivo primordial es detectar los obstáculos e informar al usuario mediante sonidos acústicos de la locación de los mismos, tanto en la distancia como en dirección es decir la distancia que existe entre el usuario y el objeto además en qué dirección está el objeto. El dispositivo tiene un rango de trabajo entre los 0.5 m y 5 m en distancia y entre 30° izquierda y 30° derecha en azimut, con una precisión de 0, 9°.

El dispositivo informa a los usuarios mediante auriculares estéreo, de la presencia de los obstáculos situados en su camino. Está compuesto por un sistema de sensores 3D-CMOS montado en un par de gafas de sol, un FPGA que se encarga de procesar la información recibida por los sensores y transformarla en sonidos acústicos, y un par de auriculares estéreo.

Las pruebas experimentales llevadas a cabo demuestran el potencial que puede tener para las personas invidentes. Se prueba que con la ayuda del dispositivo acústico los usuarios se sienten más seguros a la hora de transitar por las calles, debido a que tienen una amplia información sobre el entorno que los rodea, más allá de la obtenida con el bastón. En la Figura 4-1 se puede ver como se está usando el prototipo (Lengua, y otros, 2015)



Figura 4-1: Gráfica del dispositivo

Fuente: <https://revistas.unal.edu.co/index.php/dyna/article/view/35788/45295>

1.3. Señales de tránsito.

De acuerdo con la Agencia Nacional de Tránsito las señales “son dispositivos, signos y demarcaciones de tipo oficial colocados por la autoridad con el objeto de regular, advertir o encausar el tráfico, se utilizan para ayudar al movimiento seguro y ordenado del tránsito de peatones y vehículos”. (TRÁNSITO, 2017 pág. 34)

1.3.2. La vía pública.

La vía pública es todo espacio de dominio común por donde pueden transitar peatones y vehículos, estas vías se rigen por las normativas internacionales que las compartimos todos con los mismos derechos y obligaciones. (TRÁNSITO, 2017 pág. 8).

1.3.3. Peatones.

Los peatones son individuos que sin ser conductores pueden transitar a pie por los espacios públicos además “tendrán preferencia de cruce en zonas cebras, cruces peatonales y en toda intersección, aunque no tenga señalización, de hecho, siempre el peatón tendrá el derecho de cruce por ser el más vulnerable dentro de este panorama” como se aprecia en la Figura 5-1 la diferencia entre calzada y acera. (TRÁNSITO, 2017 pág. 9)

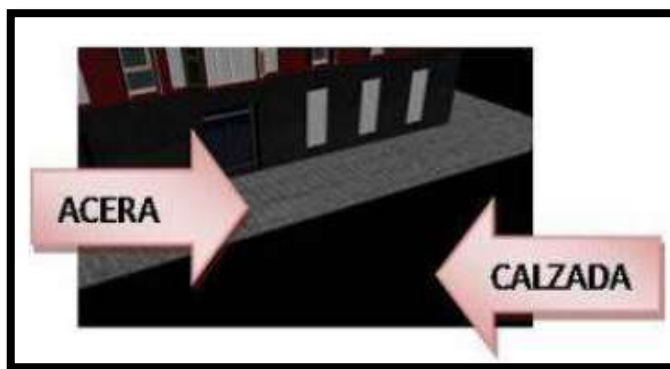


Figura 5-1: Acera y Calzada.
Fuente: AGENCIA NACIONAL DE TRANSITO



Los peatones deberán circular siguiendo los siguientes consejos de seguridad: Circulación por la vereda o acera:


- Al caminar por la acera, el niño menor debe desplazarse por la derecha, por el lado de las viviendas y no cerca de la calzada
- No transitar por el borde de la acera
- Estar atento en las salidas de garajes, entrada a playas de estacionamiento y estaciones de servicio

1.3.4. Colores de las señales de tránsito.

Normalizados los colores para las señales son y deben cumplir con las especificaciones de las normas INEN correspondientes o, en su defecto con las de la norma ASTM D 4956:

Tabla 3-1: Normalización de colores del ANT.

 ROJO	<p>Se usa como color de fondo en las señales de PARE, en señales relacionadas con movimientos de flujos prohibidos; en paletas y banderas de PARE; en señales especiales de peligro y señales de entrada a un cruce de ferrocarril entre otras.</p>
 BLANCO	<p>Se usa como color de fondo para la mayoría de señales regulatorias, delineadores de rutas, nomenclatura de calles y señales informativas; y, en las señales que tienen fondo verde, azul,</p>

	negro, rojo o café, como un color de leyendas, símbolos como flechas y orlas.
	Se usa como color de fondo para señales preventivas y complementarias velocidad, distancias y leyendas, señales de riesgo, además en señales especiales delineadoras.

Fuente: (AGENCIA NACIONAL DE TRÁNSITO 2018)

Realizado por: Oscar Segura, 2018.

1.3.5. Señales luminosas.

Existen dos tipos de señales luminosas, las peatonales y vehiculares estas “se perciben por el sentido de la vista a través de luces y colores: baliza (luz rotatoria de emergencia), semáforos, direccionales, etc”. (TRÁNSITO, 2017 pág. 80)

1.3.5.5. Semáforos.

Tanto los semáforos peatonales como los vehiculares se perciben por el sentido de la vista:

Se encuentran en las vías como semáforos y direccionales. El semáforo es un dispositivo de señalización luminosa útil para el control y la seguridad vial, mediante el cual se regula los movimientos de peatones y vehículos en las calles y carreteras. (TRÁNSITO, 2017 pág. 81)

1.3.5.6. Semáforos peatonales.

Es un dispositivo óptico luminoso de dos colores que permite y sirve dirigir la circulación peatonal, para detenerlo o ponerlo en movimiento, se los identifica en la Figura 6-1.



Figura 6-1: Semáforo peatonal.

Fuente: AGENCIA NACIONAL DE TRANSITO

1.3.5.7. Semáforo peatonal rojo fijo.

Todos los peatones están prohibidos de ingresar a la calzada para cruzar. Si se utiliza en forma intermitente, significa que los peatones si ya han empezado a cruzar la calzada pueden continuar hasta la otra acera; caso contrario deben esperar, como se puede visualizar en la Figura 7-1.

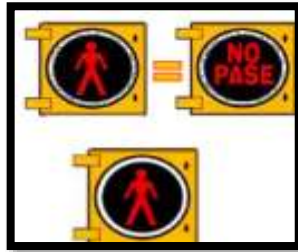


Figura 7-1: Semáforo peatonal en rojo

Fuente: AGENCIA NACIONAL DE TRANSITO

1.3.5.8. Semáforo peatonal verde.

Significa que los peatones, pueden cruzar la calzada, en la Figura 8-1 se puede identificar el semáforo en verde.




Figura 8-1: Semáforo peatonal en verde

Fuente: AGENCIA NACIONAL DE TRANSITO

Para el trabajo de titulación presente las señales peatonales de mayor importancia para los peatones, específicamente los que guían a las personas, al momento de cruzar la calle y los lugares por donde puede cruzar con seguridad, además estas se instalan en lugares específicos de la vía se detallan a continuación en la siguiente tabla:

Tabla 4-1: Señales usadas para la programación

	DETENCIÓN OBLIGATORIA - PARE Se instala en las aproximaciones a las intersecciones, donde una de las vías tiene prioridad con respecto a otra, y obliga a parar al vehículo frente a ésta señal antes de entrar a la intersección.
---	--

	<p>APROXIMACIÓN A SEMÁFORO Notifica al conductor de la existencia más adelante de un cruce controlado con semáforo, por lo que deberá tomar las precauciones para detener el vehículo en caso de que dicho dispositivo indique luz roja.</p>
	<p>PEATONES EN LA VÍA Esta señal advierte la aproximación a un tramo de vía en donde hay posibilidades que se encuentren peatones cruzando la vía.</p>
	<p>DEMARCADO TIPO CEBRA Se pintan en intersecciones con señal de pare o cerca de establecimientos educativos. Los conductores deben ceder el paso a los peatones que se encuentren cruzando la calzada.</p>
	<p>DOS LÍNEAS TRANSVERSALES En las interacciones semaforizadas, se pintan dos líneas transversales.</p> <p>LÍNEA DE PARADA Es una línea blanca continua que se pinta transversalmente a la calzada para indicar a los conductores el sitio donde deben detener sus vehículos momentáneamente. Debe estar complementada por un semáforo, una señal de PARE o una senda peatonal.</p>
	<p>Semáforo peatonal.</p> <p>Es un dispositivo óptico luminoso de dos colores que permite y sirve dirigir la circulación peatonal, para detenerlo o ponerlo en movimiento.</p>

Fuente: (AGENCIA NACIONAL DE TRÁNSITO 2018)

Realizado por: Oscar Segura, 2018.

1.4. Sistemas embebidos.

Un sistema embebido se lo define como cualquier equipo computacional programable, que cumple con una o varias funciones específicas en tiempo real.

“Un sistema embebido es un sistema cuya función principal no es computacional, pero es controlado por un computador integrado. Este computador puede ser un microcontrolador o un microprocesador. La palabra embebido implica que se encuentra dentro del sistema general, oculto a la vista, y forma parte de un todo de mayores dimensiones.” (PEREZ, y otros, 2009 pág. 4)

Todo tipo de sistema computacional está combinado por dos partes que son el hardware y el software embebido como los elementos más importantes. El sistema es independiente puesto que su software está embebido, suele alojarse en memoria solo de lectura (ROM) por lo que prescinde de una memoria volátil. (PEREZ, y otros, 2009 pág. 4)

En un sistema incrustado se distinguen tres características que definen y distinguen a este de otros sistemas computacionales.

- En este sistema se ejecutan tareas específicas de forma repetitiva, a diferencia de un sistema computacional ordinario que ejecuta cierta cantidad de programas.
- Guarda ciertas limitaciones en su implementación como: diseño, tamaño, costo desempeño, consumo energético. *“Los sistemas embebidos generalmente deben ser poco costosos, poseer un tamaño reducido, tener un buen desempeño para procesar datos en tiempo real, y además consumir un mínimo de energía para extender el tiempo de vida de las baterías o prevenir la necesidad de elementos adicionales de enfriamiento”* (PEREZ, y otros, 2009 pág. 5)
- Capacidad de reacción ante cambios ambientales, sin interrumpir su operación ni retrasar las funciones otorgadas.

1.4.2. Componentes.

Todo sistema embebido se compone de tres partes principales que son las siguientes:

1.4.2.5. Hardware.

Es la parte física que constituye en sí mismo, que en conjunto con el software realizan tareas específicas. Estos componentes se distan de los sistemas tradicionales en diversos aspectos como: consumo energético, funcionalidad, tamaño, capacidad de procesamiento, y otras características. (PEREZ, y otros, 2009 pág. 5)

El hardware que comúnmente compone un sistema incrustado se describe a continuación:

- **Microprocesador / Microcontrolador:** Componente LSI (Large Scale Integration) que brinda la capacidad de cómputo al sistema y se encarga de realizar tareas o funciones en una sola pieza de circuito integrado, el mismo que se forma por miles de transistores y otros componentes electrónicos.
- **Sensores:** Dispositivo que se encarga de percibir las señales físicas de un fenómeno para traducirla a otro tipo de señales generalmente eléctricas.
- **Memoria:** Espacio de almacenamiento para la retención de datos y su posterior uso.

1.4.2.6. Software.

Se define al software de un sistema embebido como la aplicación que ejecuta un trabajo en específico de forma repetitiva.

“El software que se ejecuta en un sistema embebido es diseñado bajo algunas restricciones importantes: (i) cantidades pequeñas de memoria, generalmente en el orden de los KB, (ii) capacidades limitadas de procesamiento, generalmente los procesadores poseen velocidades que no superan los Mhz, (iii) la necesidad de limitar el consumo de energía en cualquier instante, bien sea en estado de ejecución o no.” (PEREZ, y otros, 2009 pág. 6)

1.4.2.7. Sistemas operativos embebidos.

Los sistemas operativos proveen de una interacción agradable entre el usuario y el sistema, y a su vez le permite al usuario administrar todos los recursos del mismo. Además, un sistema operativo embebido es aquel que se ejecuta bajo un sistema incrustado con

características en tiempo real.

Pérez lo define como un “*software muy pequeño desarrollado específicamente para ser usado con un algún sistema embebido en particular, o en ocasiones puede ser una versión reducida de algún sistema operativo que se utiliza en una computadora de propósito general.*” (PEREZ, y otros, 2009 pág. 6)

1.5. Visión artificial.

Se puede determinar la a visión artificial como, la capacidad de transmitir al ser humano la capacidad de mejorar la calidad de interpretación, en el procesamiento de la información para los sistemas o máquinas autónomas. En la figura 9-1 se identifica cada una de las etapas dentro de un proceso de visión artificial.

“La visión artificial puede ser definida como los procesos de obtención, caracterización e interpretación de información de imágenes tomadas de un mundo tridimensional.” (MARTINSANZ, y otros, 2007 pág. 11)

Como un concepto, Pajares y De la Cruz lo delimitan como “la capacidad de la máquina para ver el mundo que le rodea, más precisamente para deducir la estructura y las propiedades del mundo tridimensional a partir de una o más imágenes bidimensionales” (MARTINSANZ, y otros, 2007 pág. 20)

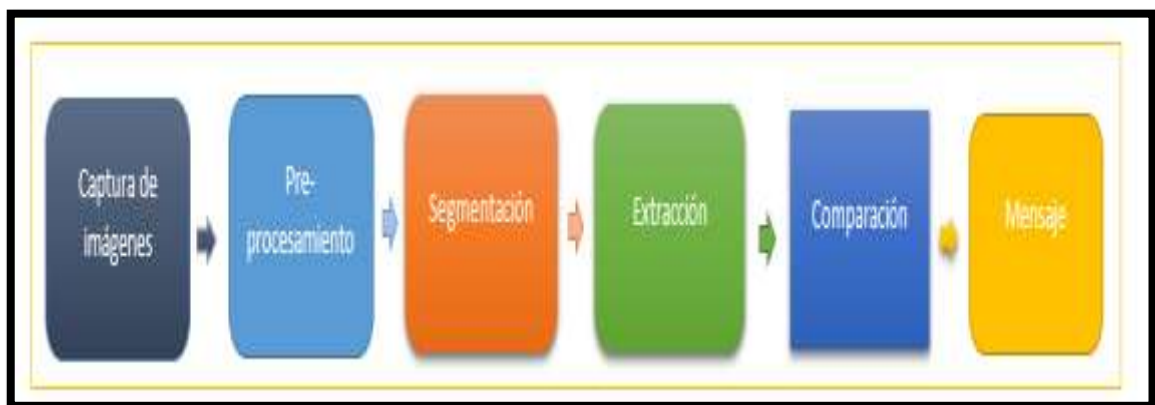


Figura 9-1: Proceso de un sistema de visión artificial

Realizado por: Oscar Segura, 2018.

1.5.2. *Arquitectura de un sistema de visión artificial.*

Un sistema de visión artificial está conformado de un mínimo de cinco elementos:

- **Dispositivo de captura:** Unidad óptica que recepta las señales analógicas del mundo real.
- **Conversión A/D:** La señal analógica obtenida se convierte a una o varias señales digitales.
- **Memoria:** Donde se almacena la información obtenida.
- **Procesador:** Trabaja en conjunto con la memoria para operar sobre la imagen digitalizada.
- **Monitor:** Permite la visualización de los procesos realizados, aunque se puede prescindir del mismo.

1.5.2.5. *Dispositivos de captura de imágenes.*

Para adquirir una imagen a partir de una escena, es fundamental un dispositivo físico que sea sensible a cierta banda del espectro electromagnético, que produzca una señal eléctrica en proporción a la energía percibida por el sensor, para luego ser digitalizada, en la Figura 10-1 se describe el proceso espectro electromagnético. (MARTINSANZ, y otros, 2007 pág. 23)

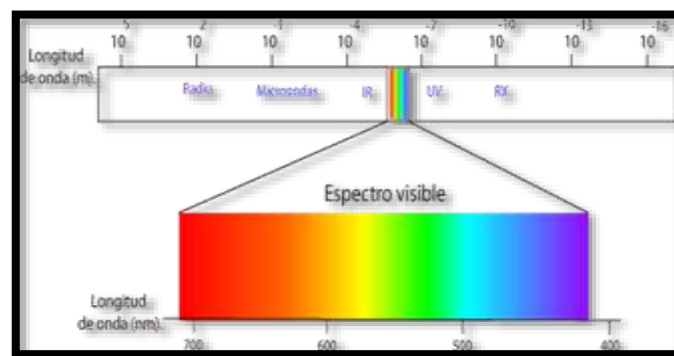


Figura 10-1: Espectro electromagnético

Fuente: <http://www.aulaclie.es/fotografia-photoshop/graficos/espectro1.gif>

Entre los principales dispositivos de captura se tienen: cámaras fotográficas, scanner, sensores de rango y ultrasonido, rayos X, imágenes de tomografías y resonancias, etc.

1.5.2.6. Digitalización de una imagen.

Es el proceso por el cual se convierte las señales analógicas obtenidas por el sensor a una señal continua, es decir a partir del muestreo de la imagen obtenida se convierte a una matriz discreta de $M \times N$ píxeles.

1.5.3. Aplicaciones de la visión artificial.

Tabla 5-1: Aplicaciones de la visión artificial.

Inspección industrial y control de calidad	<ul style="list-style-type: none">▪ Verificación de etiquetado y códigos.▪ Inspección de soldaduras, circuitos impresos, motores.▪ Clasificación de piezas.
Vigilancia y seguridad	<ul style="list-style-type: none">▪ Control de accesos.▪ Control de abandonos.
Identificación	<ul style="list-style-type: none">▪ Identificación biométrica: huellas, pisadas, firmas, iris.▪ Reconocimiento de caras, de gestos, caracteres.
Control de tráfico	<ul style="list-style-type: none">▪ Reconocimiento de matrículas, peaje por volumen, control de flujo.▪ Sistemas de ayuda a la conducción.
Guiado de robots	<ul style="list-style-type: none">▪ Industriales▪ Vehículos autónomos
Análisis de imágenes	<ul style="list-style-type: none">▪ Satélite
Aplicaciones militares	<ul style="list-style-type: none">▪ Detección de objetivos▪ Guiado balístico
Bioingeniería	<ul style="list-style-type: none">▪ Ayuda al diagnóstico (SALGADO, 2007)

Fuente: http://arantxa.ii.uam.es/~jms/seminarios_doctorado/abstracts2006-2007/20070503LSalagado.pdf.

Realizado por: Oscar Segura, 2018.

En la actualidad existe una gran variedad de aplicaciones en cuanto a visión artificial se refiere, como se puede apreciar en la tabla 5- 1.

1.5.4. Representación de una imagen.

La imagen obtenida por cualquier dispositivo de captura se presenta digitalizada en forma de una matriz de M x N elementos; representado a través de un sistema coordenado $f(x,y)$; mostrando su origen en el extremo superior izquierdo, como se puede apreciar en la Figura 11-1. A cada elemento de este arreglo se lo denomina pixel (picture element). (MARTINSANZ, y otros, 2007 pág. 30)

Todos los píxeles guardan cierta información, es decir en el caso de que la imagen esté en niveles de grises dicha información será el brillo.

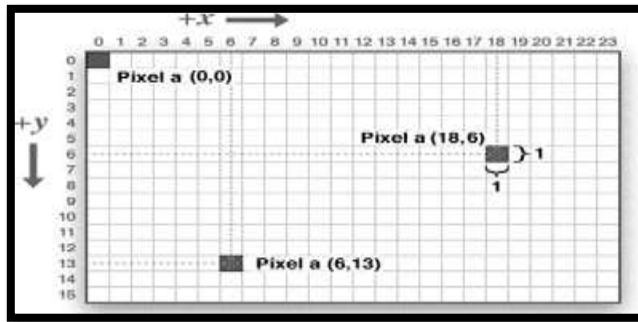


Figura 11-1: Sistema coordenado de una imagen
Fuente: <http://www.diarioelectronico hoy.com/blog/lcds-graficos-pcd8544>

Las coordenadas de un píxel (y) tiene dos vecinos horizontales y verticales, a este conjunto se le denomina, vecindad a cuatro de un píxel, en la Figura 11-1 se puede apreciar la vecindad de un pixel.

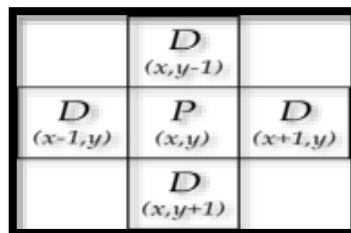


Figura 12-1: Vecindad a cuatro de un píxel.
Fuente: <https://upload.wikimedia.org/wikipedia/commons/thumb.png>

Cada uno de ellos tiene una distancia unitaria desde su posición original. Las coordenadas vienen dadas por la Ecuación 1-1.

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1) \quad \text{Ecuación 1-1}$$

1.5.5. Segmentación.

La imagen que provee un dispositivo de captura habitualmente no es lo suficientemente buena para extraer información de manera adecuada, por lo que se tiene que apartar los fragmentos o características de la imagen para facilitar su análisis o reconocimiento.

“La segmentación es el proceso por el cual a través de técnicas y procedimientos se extrae información útil de una imagen, para su posterior uso. Este proceso se basa en la similitud (se orienta hacia regiones) y discontinuidad (se orienta hacia bordes).” (MARTINSANZ, y otros, 2007 pág. 8)

1.5.5.5. Binarización.

La binarización es el proceso por el cual se compara los niveles de grises de una imagen con un valor determinado conocido como umbral (threshold). Si el nivel de gris presente en el píxel es menor al umbral, a este se le asigna un valor de cero lo que supone el color negro (Ecuación 2-1), de manera análoga, si el píxel es mayor al umbral esta toma el valor de uno o blanco (Ecuación 3-1).

$$S [x, y] = 0 \quad (0), E [x, y] \leq T \quad \text{Ecuación 2-1}$$

$$S [x, y] = 1 \quad (255), E [x, y] > T_m \quad \text{Ecuación 3-1}$$

El problema principal de la binarización es el de hallar el valor de umbral adecuado para que se tome valores de cero o uno, ya que el nivel de gris de una imagen puede variar por diversos factores y eliminar información que podría ser útil para el sistema, lo cual podría provocar ruido al momento de hacer un análisis, en la Figura 13-1 se puede apreciar la binarización de una imagen y además la inversión de los colores.

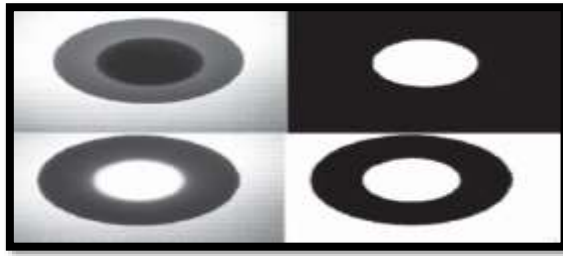


Figura 13-1: Binarización

Fuente: <https://www.researchgate.net/figure/Figura-13-Resultados-del-algoritmo-de-binarizacion>

1.5.5.6. *Detección de contornos.*

La detección de contornos es una técnica aplicada al procesamiento de imágenes de donde se extrae información de los límites de una forma o un objeto, para ser usados después para la detección y análisis del mismo. Se basa en las variaciones de la intensidad de un pixel, si éste pasa de un valor de 255 (blanco) a 0 (negro) o viceversa, representa un contorno.

De acuerdo con, Vélez mencionan que:

“El contorno de un objeto en una imagen digital corresponde al mínimo conjunto de píxeles que separa ese objeto del fondo o background de la imagen. Normalmente estos contornos se corresponden con los puntos donde se producen discontinuidades en los valores de píxeles adyacentes (cambios en el matiz o el brillo) o con los puntos donde cambia un patrón que se repite (cambios de textura).” (VÉLEZ, 2007 pág. 128)

1.5.6. *Operaciones morfológicas.*

Dentro del campo de del procesamiento de imágenes, la morfología se entiende como de la topología o la estructura de objetos a partir de su respectiva imagen. Las operaciones morfológicas se encargan del realce de la geometría de los pixeles especialmente en la teoría de conjuntos.

1.5.6.5. *Erosión.*

La erosión es la transformación morfológica que combina dos conjuntos, usando el vector resta de los elementos del conjunto. Esta operación degrada la imagen fuente, eliminando los píxeles vecinos que toman valor de uno en la imagen fuente, como se aprecia en la Figura 14-1. De acuerdo con Vélez define la erosión como “el conjunto de todos los elementos x para los cuales B trasladado por x está contenido en A ” (VÉLEZ, 2007 pág. 130)

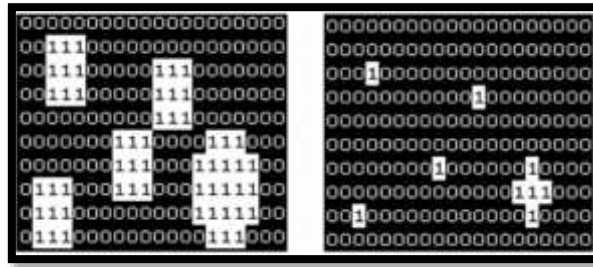


Figura 14-1: Erosión de una imagen binaria.

Fuente: <http://eia.udg.es/~rafa/docencia/morfologia.pdf>

Matemáticamente la erosión se define como:

$$A \ominus B = \{x / Bx \subseteq A\} \quad \text{Ecuación 4-1}$$

1.5.6.6. Dilatación.

La dilatación es la transformación morfológica que combina dos conjuntos usando adición de vectores lo contrario de la erosión, de los elementos del conjunto. La dilatación representa una ampliación de la imagen fuente, donde todos los píxeles de la vecindad toman un valor de uno, como se aprecia en la Figura 15-1.

Esta operación da como resultado a un conjunto de elementos, donde al menos un elemento de B es contenido en A (Ecuación 5-1).

Matemáticamente la dilatación se define como:

$$A \oplus B = \{x / (B^{\wedge})_x \cap A\} \quad \text{Ecuación 6-1}$$

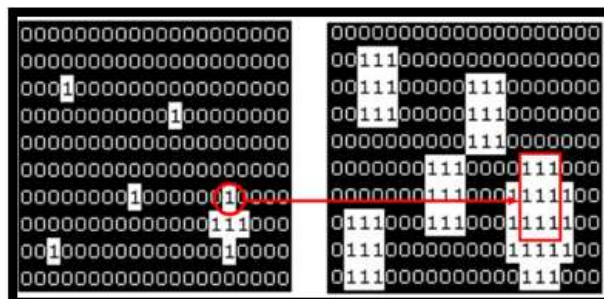


Figura 15-1: Dilatación de una imagen binaria

Fuente: <http://eia.udg.es/~rafa/docencia/morfologia.pdf>

1.6. Software para el desarrollo del sistema.

Este software permitirá el desarrollo del sistema a implementar en el trabajo de titulación, usando todos los recursos necesarios para el mismo.

1.6.2. Open CV.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Provee una infraestructura para aplicaciones de visión artificial y también es muy usada a nivel comercial, Sus librerías ofrece más de 2500 algoritmos que incluye algoritmos de machine learning y de visión artificial.

Estos algoritmos permiten identificar objetos, caras, clasificar acciones humanas en vídeo, extraer modelos 3D, encontrar imágenes similares, seguir el movimiento de los ojos, etc. Este software está desarrollado en C++ y tiene interfaces en C++, C, Python, Java y MATLAB, es multiplataforma. (OPENCV, 2017)

1.6.2.5. OpenCV – Python.

Python es un lenguaje de programación de alto nivel multipropósito, sencillo de aprender debido a su fácil codificación y sintaxis, evitando errores que son comunes al programar en diferentes lenguajes.

Su diseño visual es de baja calidad, usando palabras claves inglesas que a diferencia de otros lenguajes de programación lo hacen con el uso de puntuación. Del mismo modo para delimitar un bloque de programa se omite el uso de caracteres como paréntesis, llaves u otros.

A comparación con otros lenguajes como C o C++, Python suele ser relativamente lento debido a que se orienta a ser un lenguaje comprensible, es decir de alto nivel, por lo que toma más tiempo pasar desde el código fuente en el que está escrito hacia el lenguaje que es comprensible por la máquina o conocido como lenguaje de bajo nivel. (OPENCV, 2017)

1.7. Hardware para el desarrollo del sistema embebido.

Como se requiere reducir costos para el prototipo a construir, se buscan tarjetas de desarrollo de bajo costo, se busca un hardware que permita el desarrollo del sistema a implementar en el trabajo de titulación, usando todos los recursos necesarios para el mismo.

1.7.2. Raspberry Pi.

En la Figura 16-1, se puede visualizar el ordenador al cual se lo puede determinar como un ordenador de tamaño reducido de crédito de bajo costo, diseñado por la fundación Raspberry Pi con el fin de ser usado para muchas cosas como la enseñanza, programación, hoja de cálculo etc. Se conecta a su televisor y un teclado. Es una placa que soporta varios componentes necesarios en un ordenador común. (RASPBERRY, 2018)



Figura 16-1: Raspberry Pi

Fuente: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Existe una variedad de estas tarjetas de desarrollo que se las detalla en la siguiente tabla:

1.7.2.1. Características principales de las tarjetas de desarrollo.

Modelo	Raspberry Pi 1 B+	Raspberry Pi 2 B	Raspberry Pi 3 B+	Raspberry Pi Zero	Raspberry Pi Zero W
Fecha	2012 Feb 15	2015 Feb 1	2016 Feb 28	2015 Nov 20	2017 Feb 15
Procesador	30	35	40	5	10
SOC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2837	Broadcom BCM2835	Broadcom BCM2835
CPU	ARM1176JZF-S	Cortex-A7	Cortex-A53-64-bit	ARM1176JZF-S	ARM1176JZF-S
RF Chips	1	2	2	1	1
GPU	VideoCore IV				
CPU Clock	700 MHz	900 MHz	1.2 GHz	1 GHz	1 GHz
RAM	512 MB	1 GB	1 GB	512 MB	512 MB
Memoria	Micro SD	Micro SD	Micro SD	Micro SD	Micro SD
USB	2	2	2	1 microUSB	1 microUSB
Ethernet	SI	SI	SI	No	No
WiFi	NO	NO	SI	NO	SI
Bluetooth	NO	No	SI	No	SI
HDMI	SI	SI	SI	Mini	Mini
GPIO	26	40	40	26	26
UART	SI	SI	SI	SI	SI
SPI	SI	SI	SI	SI	SI

Figura 17-1: Características Raspberry Pi

Fuente: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

1.7.2.2. Selección de tarjeta de desarrollo mediante una comparación de características.

A continuación se muestra la comparación más alta entre Raspberry Pi y Banana Pi

Tabla 2-2: Comparación entre tarjetas de desarrollo.

Comparaciones entre Raspberry Pi y Banana Pi	Raspberry Pi	Banana Pi
Procesador	ARM Cortex A7-900 MHz - quad-core	Allwinner A20 - Cortex A7 - 1 GHz - doble núcleo
RAM	1 GB LPDDR2	1 GB DDR3
USB	4 x USB 2.0	2 x USB 2.0, 2 x MicroUSB
GPIO	Encabezado de expansión de 40 pines	Encabezado de expansión de 26 pines
Red	1 x 10/100 Ethernet	1 x 10/100/1000 Ethernet
Dimensiones	85.6 x 56 x 21 (mm)	92 x 60 (mm)
Audio	3.5 mm	3,5 mm
Vídeo	1 x HDMI - 1.3 o 1.4	1 x HDMI, 1 x compuesto

Realizado por: Oscar Segura, 2018.

En conclusión el diseño de la Raspberry Pi está girando en torno a su presupuesto y facilidad de uso, que es la razón principal de su enorme demanda en el mercado. El soporte generalizado de otro software solo complica la utilidad del dispositivo, con mucha información sobre cómo escribir y compilar software, hardware creado para Raspberry Pi. Si bien también existe la misma información para Banana pi, pero no es tan común, lo cual es complicado ya que Banana Pi es un poco más difícil de configurar que Raspberry Pi.

1.7.3. Módulo de cámara.

Esta cámara tiene un sensor Omnivision de 5 Mpíxeles que permite realizar capturas de 2.592 x 1.944 píxeles y que también captura vídeo a 1080p con una tasa de 30 fotogramas por minuto.

En la Figura 18-1 se muestra la cámara Raspicam que es compatible en un 100 % con

cualquier versión o modelo de tarjeta Raspberry pi, ya que trae sus librerías ya preinstaladas en el sistema operativo de la tarjeta Raspberry pi. (HARDZONE, 2018)

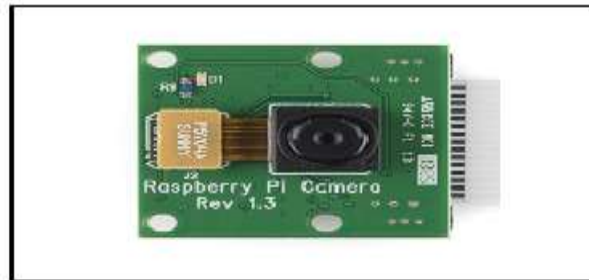


Figura 18-1: Cámara Raspicam
Fuente: (SEGURA, 2018)

1.7.4. Batería Portátil.

La batería portátil o power bank es básicamente una batería externa dentro de una carcasa. Puede cargarse a través de una fuente de energía tipo computador portátil o un cargador normal de corriente, siempre utilizando un cable con entrada USB. En la figura 19-1, se muestra el modelo de batería existente en el mercado, estas son de un modelo de fácil manipulación.



Figura 19-1: Batería Portátil
Fuente: www.ohmygeek.net/2014/11/24/genius-eco-u267/

CAPÍTULO II

2 MARCO METODOLÓGICO

En el siguiente capítulo se detalla el diseño de *hardware* y *software* para el prototipo a implementar, donde se da detalle de cada elemento con sus características que se usaron en el prototipo. También se detalla las etapas del proceso, así como los esquemas del sistema.

Es un sistema fundamentado en visión artificial, se considera la adquisición de una cámara la cual permita adquirir imágenes del sitio de interés.

2.1 Arquitectura general del sistema.

Como se muestra en el Grafico 1-2, la arquitectura general del prototipo a implementar, se describe de manera universal el funcionamiento de las diferentes etapas del sistema, desde el primer bloque la cual es adquirir imágenes por medio de una cámara para posteriormente pasar al bloque de procesamiento de imágenes, donde se realizará el tratamiento o procesamiento de las mismas mediante un algoritmo para determinar las señales peatonales. Y en el último bloque se obtendrá la información mediante un mensaje de voz.

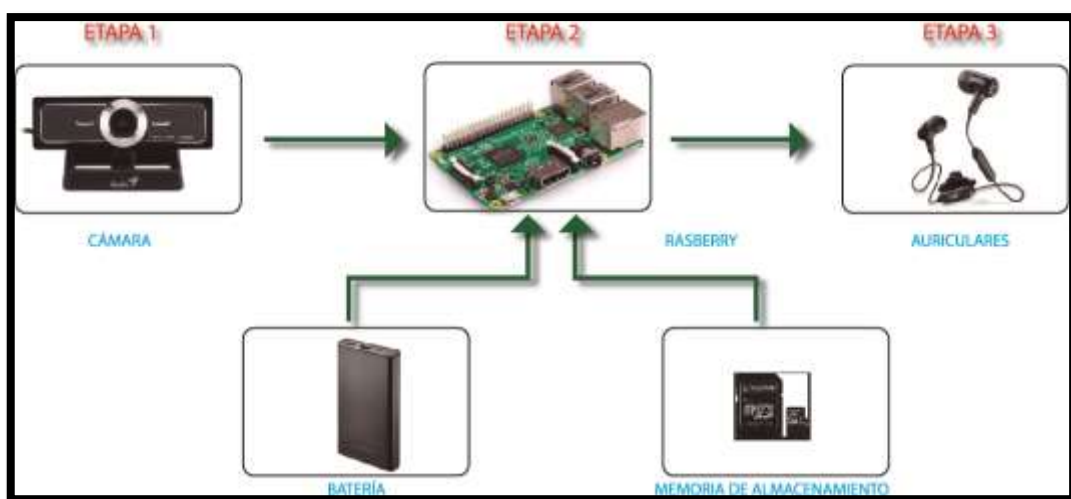


Figura 1-2: Arquitectura general del sistema

Realizado por: Oscar Segura, 2018

De acuerdo con la arquitectura del prototipo el cual está constituido por tres partes principales las cuales se describen a continuación cada una de ellas:

En el primer bloque se tiene la adquisición imágenes mediante una cámara WideCam F100, en el segundo bloque se procesa todas las imágenes adquiridas por la etapa anterior, mediante una tarjeta de desarrollo Raspberry Pi 3 y por último en el tercer bloque se tiene el envío de mensajes de voz con la información detectada en la imagen.

2.2 Diagramas de bloques del procesamiento de las imágenes.

Dentro del segundo bloque es donde se realizará el sistema de visión artificial la cual tiene varias etapas para llegar al resultado deseado cada una de las etapas son importante por eso se detallará cada una de ellas:

- La primera fase, es guardar las imágenes provenientes de la cámara.
- La segunda fase consiste en el tratamiento digital de las imágenes, con objeto de facilitar las etapas posteriores. En la etapa de procesamiento previo es donde, mediante filtros y transformaciones geométricas, se eliminan partes indeseables de la imagen o se realzan partes interesantes de la misma.
- La tercera fase junto con la cuarta se conoce como segmentación y extracción, y consiste en aislar los elementos interesados de una escena para comprenderla.
- Después, se llega a la etapa de comparación. En ella se pretende distinguir los objetos segmentados, gracias al análisis de ciertas características establecidas previamente.
- Una vez hecha la comparación se envía un mensaje el cual sería la última etapa del procesamiento de cada imagen obtenida de la primera etapa.



Figura 2-2: Bloques del procesamiento de las imágenes

Fuente: Oscar Segura, 2018.

2.3 Requerimiento funcional del prototipo.

Para el prototipo se deben satisfacer los siguientes requerimientos:

- Detección de señales peatonales por medio de una cámara.
- Generar alertas auditivas una vez detectada la señal a través de un auricular.

2.4 Requerimientos del sistema:

- Dar avisos con alertas auditivas
- Alimentación portátil
- Consumo de corriente máximo 1 amperio
- Comunicación alámbrica

2.5 Selección de los elementos electrónicos para la construcción del prototipo.

A continuación, se detallará las características de cada uno de los elementos seleccionados para el prototipo que forman parte del sistema.

2.5.1 Elementos hardware del prototipo.

Dentro del mercado se puede encontrar una gran gama de dispositivos electrónicos para diversas aplicaciones, para el prototipo implementado se eligieron los siguientes elementos, por sus características, su uso, conexión, tipo de sensor y demás peculiaridades.

2.5.1.1 Selección de dispositivo de captura.

Para seleccionar el dispositivo de captura, de imágenes o la cámara se fundamenta en 2 requerimientos importantes, la resolución y el campo de visión de la cámara, para ello la primera opción fue escoger la cámara Raspicam, pero no cumplió los requerimientos necesarios al momento de realizar las pruebas, el modelo de cámara escogido fue una genius WideCam F100 cuyas características fueron las siguientes.

La cámara tiene un lente de con un campo de visión, hasta 120 grados que le permite ver un ángulo con mayor amplitud, la WideCam F100 incluye un micrófono estéreo incorporado que también le permitirá hablar con claridad. El lente de cristal brinda una mejor calidad de la imagen, en la Figura 3-2 se puede apreciar la cámara y en la Tabla 1-2 sus características



Figura 3-2: Cámara WideCam F100
Fuente: Oscar Segura, 2018.

Especificaciones técnicas:

Tabla 1-2: Características de cámara WideCam F100.

Sensor de imagen:	CMOS de píxeles Full HD 1080p
Tipo de lente:	objetivo de enfoque manual
Formato de archivo:	MPEG/WMV/JPG
Consumo energético	100mA y 0.2W
Con Micrófono:	SÍ
Resolución (DPI):	12MP (interpolación), 1920 x 1080, 1280 x 720, 640 x 480 píxeles
Peso:	82 g
Dimensiones (A x A x P):	150 x 49 x 48 mm

Realizado por: Oscar Segura, 2018.

El análisis de comparación es el siguiente se tomó dos fotografías a la misma distancia y se comparó el ángulo de visión que se puede constatar en la siguiente Figura 4-2.



Figura 4-2: Comparación de imágenes izquierda f 100 derecha cámara Raspberry
Fuente: Oscar Segura, 2018.

2.5.1.2 Tarjeta de desarrollo Raspberry pi 3.

La Raspberry Pi 3, fue seleccionada como parte del prototipo porque sus características se convenientes para los requerimientos del prototipo. La tarjeta es más rápida que sus antecesores, por ese motivo fue elegida para el desarrollo del proyecto. Cuenta con un procesador de cuatro núcleos el cual es más poderoso a los primeros, se puede apreciar en la Figura 5-2.



Figura 5-2: Tarjeta de desarrollo Raspberry pi 3
Fuente: Oscar Segura, 2018.

Tabla 3-2: Características Tarjeta de desarrollo Raspberry pi 3.

Unidad central de proceso	1.2 GHz Arquitectura de 64 bits Cuatro núcleos ARMv8
Juego de instrucciones	RISC de 32 bits

Memoria (SD RAM)	1 Gb
Puertos USB 2.0	4
Entradas de video	Conector MIPI CSI, permite alojar el módulo de cámara.
Salidas de video	Conector RCA High Definition Multimedia Interface Interfaz DSI para panel LCD
Almacenamiento	A través de tarjeta Micro SD
Conectividad a la red	Ethernet RJ45 WIFI 802.11n Bluetooth 4.1
Periféricos de bajo nivel	40 x GPIO
Consumo energético	700 mA 3.5W
Fuente de alimentación	5V vía micro USB
Dimensiones	2.5 m x 53.98 mm

Fuente: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

Realizado por: Oscar Segura, 2018.

2.5.1.3 Alimentador portátil.

Se seleccione este dispositivo conocido como Power Bank es una batería portátil que tiene la capacidad de alimentar a los tres bloques del sistema por un tiempo determinado, para que puedan funcionar todos sus componentes de forma alámbrica, en la Figura 6-2 se puede apreciar la batería portátil a utilizar.

Primero se sumó las corrientes eléctricas de los dispositivos que se van a alimentar de la batería, como se muestra en la Tabla 3-3.

Tabla 4-2: Características de la batería portátil.

Cámara WideCam F100	100 mA
Tarjeta de desarrollo Raspberry pi 3	700 mA
Auriculares	50 mA
Consumo eléctrico total de las cargas	850mA

Realizado por: Oscar Segura, 2018

Con el valor total de las cargas que estarán conectadas a la batería se puede determinar que batería será la indicada, para que abastezca dicho consumo y para poder determinar la durabilidad del alimentador portátil con el dispositivo conectado se emplea la siguiente fórmula, que es para determinar el tiempo de descarga de la batería:

$$T_{descarga} = \frac{\text{capacidad_bateria}}{\text{consumo_electronico}}$$

$$T_{descarga} = \frac{2600mAh}{850mA} = 3h$$

Esto nos quiere decir que la batería va a durar 3 horas, luego de este tiempo se debe cargar el alimentador del prototipo, en la Figura 6-2 se puede ver el tipo de batería portátil que se está usando.



Figura 6-2: Alimentador portátil.

Fuente: Oscar Segura, 2018.

Tabla 5-2: Características de la batería portátil.

Voltaje de salida	5 Voltios (Corriente Directa)
Corriente de salida	1,5 Amperios
Capacidad en máxima	5200 (mili amperio hora).
Puertos micro USB	1 micro USB para recargar el alimentador.
Puertos USB	1 USB para entregar la energía
Indicador de carga	<i>I</i> ndicador, para saber si necesita ser cargado.

Fuente: <https://www.regalopublicidad.com/bateria-portatil-de-2200mah>

Realizado por: Oscar Segura, 2018

2.5.1.4 Memoria micro SD.

Como se muestra en la Figura 7-2. La memoria micro SD es un elemento de almacenamiento el cual tiene un papel muy importante, en la memoria se va a grabar el sistema operativo, el cual proveerá el funcionamiento a la tarjeta Raspberry pi 3, una vez instaló el sistema operativo en la memoria se inserta en la ranura de la tarjeta Raspberry pi 3 para luego iniciar con las configuraciones en la tarjeta.



Figura 7-2: Alimentador portátil.

Fuente: Oscar Segura, 2018

2.5.1.5 Auriculares.

Como se muestra en la Figura 8-2 el auricular permite transmitir un mensaje de voz hacia el usuario y él pueda recibir la información proveniente de la tarjeta de desarrollo Rasp



Figura 8-2: Auriculares.

Fuente: Oscar Segura, 2018.

2.5.1.6 Esquema de conexión electrónico.

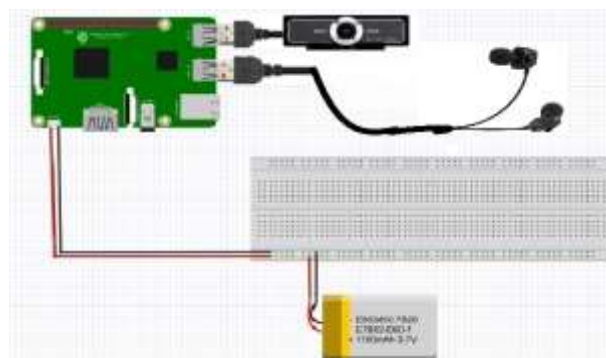


Figura 9-2: Esquema de conexión electrónico

Fuente: Oscar Segura, 2018.

2.6 Requerimiento para el diseño del algoritmo.

El prototipo debe cumplir los siguientes requerimientos:

- Procesar imágenes en tiempo real.
- Funciones para pre procesar y procesar las imágenes adquiridas.
- Algoritmos y métodos de comparación de imágenes para emitir un resultado.
- Métodos de indexación de imágenes que sirvan como base de datos para comparar imágenes.

2.6.1 Instalación de Programas en la Raspberry Pi 3.

El primer paso que se realizó fue obtener una tarjeta de memoria micro SD con capacidad de 16 gigabytes clase 10 de la marca Sony, eso es lo recomendado por Raspberry con una capacidad de almacenamiento de entre 8 a 16 gigabytes con una velocidad de escritura de 10 MB/s. esta memoria albergara el sistema operativo Raspbian Stretch with Desktop.

2.6.1.1 Sistema operativo Raspbian Stretch.

Raspbian es una distribución del sistema operativo GNU/Linux y por lo tanto libre basado en Debian Stretch (Debian es una comunidad conformada por desarrolladores y usuarios) para la placa computadora Raspberry Pi en sus diferentes versiones hasta la fecha presente. (SALAZAR, 2018)

2.6.1.2 Características del sistema operativo Raspbian Stretch.

- Coste: Debian es un sistema operativo (S.O.) de libre distribución (es decir sin coste alguno).
- Multiusuario: permite a varios usuarios acceder al mismo tiempo a través de terminales, y distribuye los recursos disponibles entre todos.
- Multiplataforma: Es decir que puede correr en la mayoría de plataformas del mercado (procesadores de la gama Intel y AMD, Motorola, Sun, Sparc, etc.).
- Kernel: Los sistemas Debian actualmente usan el núcleo de Linux.
- Memoria: La memoria se gestiona como un recurso unificado para los programas de usuario y para el caché de disco, de tal forma que toda la memoria libre puede ser usada para caché y ésta puede a su vez ser reducida cuando se ejecuten grandes programas.

- Licencia: Debian nace como una apuesta por separar en sus versiones el software libre del software no libre, para esto debe respetar 4 libertades: 1. libertad para usarlo.2. libertad para modificarlo.3. libertad para copiarlo.4. libertad para distribuir las modificaciones.
- Estabilidad: como Debian es una distribución que ha probado su estabilidad y utilidad, muchos desarrolladores la han tomado para crear otras nuevas como: Knoppix, Ubuntu, Sidux, etc.
- Seguridad: los problemas de seguridad se solucionan rápidamente con parches de seguridad que se actualizan en internet.

Posteriormente se procedió a descargar el sistema operativo Raspbian de la página oficial, www.raspberrypi.org luego de obtener el sistema operativo, se procedió a formatearlo la memoria bajo FAT-32 que es el formato para manejo de la Raspberry Pi 3. En la figura 10-2 se puede apreciar la imagen del sistema operativo seleccionado a descargar.



Figura 9-2: Imagen a descargar del SO Raspbian.

Fuente: Descargas. www.raspberrypi.org/downloads/raspbian

2.6.1.3 Instalación del sistema operativo en la Raspberry PI 3.

Luego de la descarga del sistema operativo Raspbian, se procedió a guardar la descarga en la tarjeta de memoria micro SD, mediante el programa Win32 Disk Imager, el cual permitió seleccionar el archivo de imagen y la ubicación del dispositivo en donde se desea guardar la descarga correspondiente, la cual fue descargada en la tarjeta SD. En la Figura 11-2 se muestra la pantalla principal del programa Win32 Disk Imager

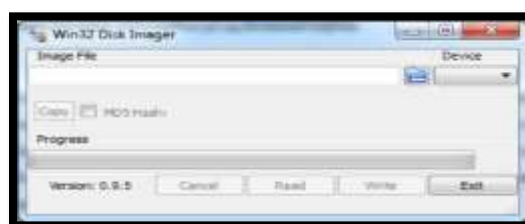


Figura 10-2: Programa Win32 Disk Imager.

Realizado por: Oscar Segura, 2018

Con esto la imagen almacenada en la tarjeta micro SD se conectó a la Raspberry junto con todos los periféricos de entrada y salida (teclado, pantalla, mouse y cable ethernet), para el manejo de la placa. Al energizar la Raspberry automáticamente arrancó el sistema operativo y mostró la pantalla de inicio del escritorio, como en la Figura 12-2.



Figura 11-2: Escritorio de la Raspberry Pi 3

Realizado por: Oscar Segura, 2018

2.6.1.4 Descarga e instalación de Python 3.0 y librerías para uso de imágenes y video.

Después de que el sistema operativo funcionaba correctamente, se pasó a la instalación de las librerías usadas en este prototipo, que permitieron realizar tareas de visión artificial por computador.

De acuerdo a la guía de instalación de OpenCV por parte de pyimagesearch, que se la puede encontrar en la siguiente dirección web, <https://www.pyimagesearch.com/2017/09/04/raspbian-stretch-install-opencv-3-python-on-your-raspberry-pi/> explica paso a paso la instalación y además se pueden encontrar otras guías, para la instalación de otras librerías complementarias.

2.6.2 Programación del prototipo.

Una vez realizado todo el proceso de instalación de librerías y complementos, se procedió a empezar la programación para lo cual se explica a detalle, las partes más importantes, el primer paso fue la captura de imágenes como se explica continuación.

2.6.2.1 Captura de imagen.

Como se sabe la cámara capta la imagen a través del lente convierte las fotos que se proyecta en ella mediante un converso ACD analógico digital, lo hace un conjunto de matrices y luego lo envía mediante comunicación serial pasando por el puerto serial.

Al describirse este proceso el software Python y las librerías de OpenCV deben hacer una captura a la vez y procesarla, para hacer esto se realiza un conjunto de comando requerido para su correcto funcionamiento.

El primer comando es establecer un identificador en este caso la identificación será el 0 para el dispositivo conectado el cual será una cámara web, en este caso solo será una cámara así que el id siempre será cero. Luego se establece las dimensiones de la imagen proveniente de la cámara que serán las máximas que puede abarcar la misma, estas son 1920 x 1080 con el comando set.

Para finalizar sea uso del comando read para obtener las capturas de la cámara y se almacenan en una variable como se muestra en la Figura 13-2.

```
#!/usr/bin/env python
...
"""
En este caso, si quiere decir que queremos acceder
a la cámara 0. Si hay más cámaras, puede ir probando
con 1, 2, 3...
"""
cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    if ret == True:
        cv2.imwrite("foto.jpg", frame)
        print("Foto tomada correctamente")
    else:
        print("Error al acceder a la cámara")
...
"""
Finalmente liberamos o soltamos la cámara
"""
...

```

Figura 12-2: Comando para captura
Realizado por: Oscar Segura, 2018

2.6.3 Selección de librerías para el desarrollo del programa.

Las librerías implícitas en este programa están entrelazadas más con el procesamiento de imágenes, así como el manejo de matrices, la primera y la principal es OpenCV con ella se realizó la programación en su mayoría, ya que este abre un conjunto de funciones y brinda una gran facilidad al momento de programar. Una de las principales tareas de esta función es la utilización de la segmentación mediante el uso de la detección de contornos y la conversión a distintos formatos de color, como se puede apreciar los dos iconos de la Figura 15-2.

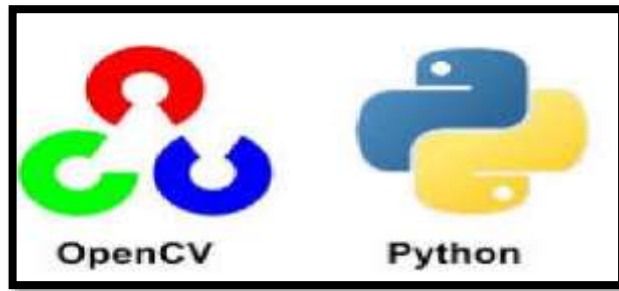


Figura 13-2: OpenCV y Python.
Realizado por: Oscar Segura, 2018

La segunda herramienta es NumPy, esta librería abre un conjunto de funciones que permiten crear insertar, eliminar y seleccionar elementos en una matriz, así como también hacer análisis estadísticos básicos como la media la mediana y la varianza, también la conversión de un vector a matriz y viceversa.

La tercera librería es imutils es un complemento de OpenCV que permite realizar técnicas de visión artificial, como resize para cambiar las dimensiones de una imagen, rotarla a un determinado ángulo, recortarla, acercarla y para esta implementación se hizo uso de la herramienta que permite redimensionar la imagen.

La cuarta librería es skimage, un complemento de visión que para algunos es mucho mejor que OpenCV tiene una gran gama de funciones tanto para análisis de segmentación como de morfología en visión artificial, para esta programación se usó la función que permite hacer una transformación probabilística de Hough y también para hacer un redimensionamiento de la imagen. Como funciones complementarias se usó de la librería math, scipy que permiten hacer operaciones aritméticas, geométricas y trigonométricas necesarias para esta aplicación.

La última librería es pygame que es necesaria para la reproducción de audios al finalizar cada uno de los procesos correspondientes en cada señal. En la Figura 15-2 se muestran estas librerías.



Figura 14-2: NumPy, scipy scikit image
Realizado por: Oscar Segura, 2018

2.6.4 Detección de líneas paralelas, semáforo y paso cebra.

Para la detección de líneas paralelas se necesita realizar una captura de imagen desde la cámara la cual debe ser en color o entro formato que contenga tres matrices para su correcto procesamiento. Luego de eso se hace una extracción del brillo de la imagen esto se hace usando el formato de color HSV, analizando la última matriz que lo compone en este caso del brillo.

Al trabajar con una imagen hay q tener en cuenta si la componen una matriz o un conjunto de matrices, para analizar el brillo solo se hace uso de una sola matriz, con la cual se debe calcular un único valor del conjunto de valores que componen la matriz de la imagen, por lo cual se debe convertir esa matriz en un array con el comando apropiado en Python al obtener este valor se compensa el nivel de brillo necesario para el correcto proceso posterior.



Figura 15-2: Filtro de color HSV

Realizado por: Oscar Segura, 2018

2.6.5 Filtrado para colores primarios.

Después de compensar el nivel de brillo de la imagen se procese a aislar un determinado espacio de color, es decir los colores cuyas componentes sean más apegadas o lo más próximas al color primario que componente el formato RGB (rojo, verde, azul), ya que de antemano se sabe que las señales de transistor a analizar se componen de colores vivos con tonalidades rojas verdes y amarillas eso permite filtrar color con tonalidades grises o muy suaves. Para realizar esto se elabora un formato específico de color, el formato YCrCb utilizando para esto el comando correcto que permita la conversión de color RGB a dicho formato deseado. Luego se hace uso de una inversión de color a BGR procediendo a repetir el proceso para la siguiente imagen.



Figura 16-2: Filtro de RGB invertido

Realizado por: Oscar Segura, 2018

Para seguir procesando la imagen anteriormente mencionada, se debe tomar solo una matriz o componente ya que esta tiene 3 matrices, se aísla los tres canales y se escoge una que tenga un nivel de umbralización aceptable, dando como resultado una imagen binaria compuesta por dos colores básicos negro y blanco cuyos valores únicamente serán 0 o 255 ya que son datos de tipo byte.



Figura 17-2: Resta de colores invertidos

Realizado por: Oscar Segura, 2018

1.3.7. Segmentación.

Una vez obtenida la imagen binaria o umbralizada el siguiente paso es aislar el ruido adyacente de lo que se va a filtrar para ello hay que segmentar dicha imagen, es decir aislar cada objeto por separado y analizarlo, en Python este proceso se hace con el comando `findcontours` para contornos cerrados. Cada contorno se analiza dentro de un bucle en una estructura `for` para aplicar los filtros de morfología correspondientes.



Figura 18-2: Segmentación
Realizado por: Oscar Segura, 2018

2.6.5.1 Filtro 1.

Primer filtro es para eliminar objetos que estén en el contorno de la imagen, ya que las señales de tránsito estén casi en el centro de la misma así cualquier objeto cuyas compenretres de los extremos estén el contorno se eliminaran. Como parámetro de control se escogió mayor a 10 pixeles en el origen y menos 10 pixeles con respecto al ancho y largo de la imagen, es necesario almacenar en una nueva imagen para ir almacenando los contornos deseados y que hayan pasado el filtro mencionado.

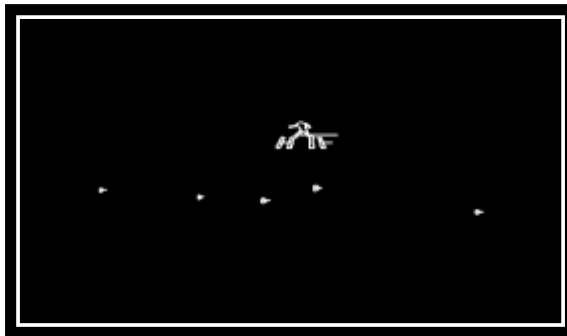


Figura 19-2: Segmentación total
Realizado por: Oscar Segura, 2018

2.6.5.2 Filtro 2.

Para comenzar con el segundo filtro debe procesarse lógicamente el primero, en un orden ya establecido, el segundo filtro es el encargado de filtro por morfología, es decir solo para formas rectangulares o cuadradas y que la señal a analizar tiene dicha forma, para lograr esto se hace uso de una función que permite calcular la longitud de arco del contorno y hacer una aproximación geométrica de la misma en su esquina ,al tener 4 esquinas o bodes con una forma rectangular pasara y se almacenara en segunda variable llamada filtro .

Comparación con base de datos:

Con el segundo filtro listo se procese a rellenar dichas imágenes y que solo son contornos vacíos para ellos se hace uso del comando floodfill y luego de una inversión de la imagen binaria.

2.6.5.3 Filtro 3.

El tercer y último filtro, se analiza y se compara con una base de datos, imágenes que se han procesado anteriormente y las que sirven para hacer una correlación con la imagen actual, para ello debe hacerse un redimensionamiento de cada objeto que se va a hacer ese momento, para ello se necesita una vez más de la segmentación con el comando findcontours. al dimensionar nuevamente se calcula un valor relativo de dimensión el cual es ancho sobre largo del contorno de la imagen, si ese valor está dentro de un rango previamente establecido basado en el método experimental en ese momento se hace la comparación con la base de datos, y este proceso se efectúa por la sencilla razón de optimizar el tiempo de ejecución. El mismo proceso se hace tanto para las tres señales.

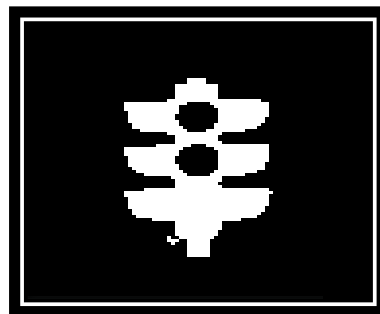


Figura 20-2: Base de datos

Realizado por: Oscar Segura, 2018

Al comparar con la base de datos y haber pasado un valor umbral se toma la decisión que la señal está en un punto determinado y se avisa por medio de un audio pregrabado que la señal este ahí. Cabe recalcar que todo esto se hace con una función llama matchTemplate que una correlación entre las dos matrices la base y el objeto que se analiza en ese momento.

2.6.6 Detección del pare.

Para la detección del pare se deben analizar 2 parámetros importantes el color y la morfología, así que el primer filtro que debe analizarse el de color, el color principal que compone esta señal es el rojo y el blanco. Primeramente, se analiza el color rojo, para ello se hace uso del filtro que se mencionó anteriormente en donde se restaban 2 imágenes con sus componentes invertidas obteniendo tonalidades más cercanas a los colores primarios rojo, verde y azul, una vez obtenida

esta imagen se hace un filtro basándose en el espacio de color rojo en las tres componentes de color RGB.

2.6.6.1 Filtrado de color rojo.

El espacio se desplaza en estos dos intervalos 17,15,75 a 175, 165,255, cabe recalcar que estos espacios de color esta distribuidos de la siguiente manera azul, verde y rojo. El filtro como tal no es ciento por ciento efectivo tiene sus falencias ya que puede filtrar componentes amarillas o tomates pero que luego se irán eliminando, en la presente imagen se muestra los resultados del filtro de color.



Figura 21-2: Filtros de color

Realizado por: Oscar Segura, 2018

Para la creación de la base de datos hay que analizar lo siguiente, esta señal tiene un grado de dificultad ya que la cada palabra que la compone de 4 letras que están separadas, así que al segmentar serán cuatro objetos separados, al que para analizarlos se debe encontrar una forma de unirlos para ello se hace uso de una función que permita encontrar el centro de cada objeto y prolongar una línea hacia la derecha para unirlos de esa forma se adquiere un solo contorno listo para analizar y almacena como la respectiva base de datos .



Figura 22-2: Símbolo del pare

Realizado por: Oscar Segura, 2018

2.6.7 Comparación de base de datos.

Debe recalarse que la base de datos debe almacenarse como una imagen binaria o en escalas de grises debido a que la función para hacer la correlación, trabaja con imágenes que solo tenga una componente.



Figura 23-2: Creación de base del pare

Realizado por: Oscar Segura, 2018

El procedimiento posterior ya teniendo el filtro de color y umbralizada es el mismo se segmenta con la función ya mencionada, al entrar al ciclo for se hace un nuevo dimensionamiento del contorno en base a los parámetros tanto largo como ancho de la imagen empleada como base de datos correspondiente. si su valor de coincidencia es superior al 0.59 se da como resultado que la señal de pare ha sido detectada en caso contrario no lo hará. Cabe mencionar que la base de diseño tomando en cuenta una distancia aproximada de 5 a 6 metros de esa forma se evita de que la detección sea a cada momento y mejora el tiempo de ejecución del programa, para todas las señales se toma en cuenta el mismo parámetro.



Figura 24-2: Resultado obtenido

Realizado por: Oscar Segura, 2018

2.6.8 Detección de semáforo rojo.

Para la detección de esta señal se tomó en cuenta las características de la misma como obviamente el color de la misma y su morfología, tanto de la señal de alto como las formas adyacentes. Para este proceso se usó el mismo filtro de color como el empleado en la señal de pare, pero con ciertas diferencias.



Figura 25-2: Resultado obtenido

Realizado por: Oscar Segura, 2018

2.6.8.1 Creación de mascara para filtración de color.

Para hacer la detección de color apropiada se tuvo que hacer una máscara para el filtrado de color rojo en una imagen RGB esto se logró usando la función bitwise, esta tiene como misión hacer operaciones binarias entre imágenes cuya característica principal, es tener las mismas dimensiones tanto de largo como de ancho. Entre las operaciones más comunes están la operación not and y or también puede usar la or exclusiva, pero la que se va a utilizar es la operación and esta permite generar una máscara necesaria para la umbralización, para ello se necesita de una imagen como variable de almacenamiento.



Figura 26-2: Creación de mascar

Realizado por: Oscar Segura, 2018

Sebe mencionar que no solo se toma en cuenta la tonalidad roja para el análisis sino también las claras ya que si es cierto que el color más dominante o abundante de esta señal es el rojo el color de las letras que van a hacer comparadas con la base de datos es blanco, pero no se hace un filtro de color del mismo ya que se toma como un espacio vacío.

Al realizar la resta del formato de color previamente descrito con sus componentes de color claramente invertidas es decir un objeto blanco con caracteres de color negro con ciertas dimensiones y área serán procesados adecuadamente.

2.6.8.2 Umbralización para color rojo.

Luego de obtener una umbralización con el filtro de color, el segundo filtro consto de un coeficiente de relación de área y dimensiones, como se sabe al estar a 6 metros la señal de rojo del semáforo tiene ciertas dimensiones y cierta área con lo que se puede, al trabajar con esta cámara crear un rango determinado.

Al pasar por los filtros anteriormente mencionados se debe almacenar en variables para analizar los bordes correspondientes por segmentación. Pero antes de eso se pasa por otro filtro que permite saber con más certeza si se trata de un semáforo o no.

Se analiza el semáforo detalladamente se puede contrastar que tiene un nivel de brillo alto y que a su alrededor el nivel de brillo en bajo ya que está dentro de una forma rectangular de un color verdoso opaco y que se encuentra siempre en una esquina. Basándose en este análisis se puede hacer los filtros correspondientes.



Figura 27-2: Umbralización color rojo

Realizado por: Oscar Segura, 2018

La morfología es importante para analizar o examinar las formas dentro de las imágenes y más en este caso como se mencionó anteriormente se hizo una toma del área, largo y ancho al pasar por el filtro es la correlación más importante en la comparación de las matrices en las imágenes, en este caso se comparó con la base de datos correspondiente, así como el nivel de brillo adyacente para detener minar si ha detectado un semáforo en rojo.

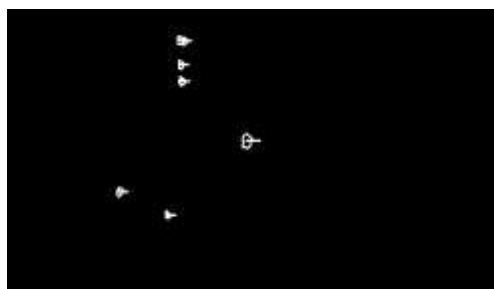


Figura 28-2: Filtración por morfología

Realizado por: Oscar Segura, 2018

Debe recalcar que la base de datos es decir las figuras guardadas son para comparar con las figuras que están dentro de las imágenes capturadas con la cámara. Estas imágenes de la base de datos fueron tomadas a una distancia de 3.5 a 5 metros, así como en las demás señales.



Figura 29-2: Resultado obtenido

Realizado por: Oscar Segura, 2018

2.6.9 *Detección de semáforo en verde.*

Para esta detección no se hizo uso de filtro de color como en las señales anteriores sino se partió del análisis de la detección de color rojo para el semáforo detallado anteriormente al tener esta imagen que servirá como mascarará posteriormente, así que se partió de una lógica muy simple al tener los colores más notables y apegados a los colores primarios, el color verde y tonalidades azules serán las más separadas de las rojas y tomates por tanto lo que no sea con tonalidades rojiza será de color verde, al invertir esta imagen se hizo el mismo proceso se creó una máscara de la misma forma y siguiendo el mismo proceso.



Figura 30-2: Color verde en formato HSV

Realizado por: Oscar Segura, 2018

2.6.9.1 *Análisis morfológico.*

Al pasar por esto se almacena en una variable para su análisis posterior, el segundo filtro es de morfología tanto área como dimensiones de la imagen para esta imagen cambian dichos valores y que entran en un rango establecido.

Al comparar con la base de datos se lo hace de la misma forma primero dimensiona cada contorno en un bucle for, después se convierte nuevamente en imagen binaria y si pasa por el coeficiente establecido en un intervalo determinado se procese hacer la correlación con el matchtemplate, esta función da como resultado un solo valor escalar que si es superior a un valor de control pasa al siguiente filtro.

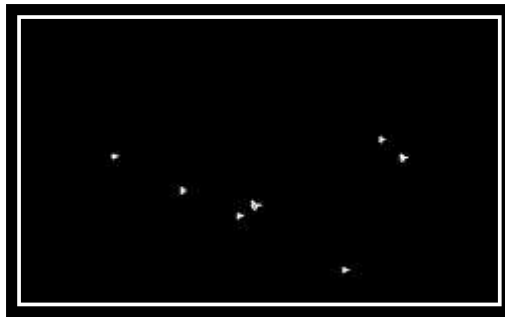


Figura 31-2: Análisis por morfología

Realizado por: Oscar Segura, 2018

Lo último sería establecer un área de interés ya que como se mencionó anteriormente el semáforo tiene a su alrededor un color verdoso oscuro que permite contrastar con el nivel de brillo y de esa forma ser más perceptible visualmente a los transeúntes y conductores, lo mismo hace este proceso establece la región de interés y si la zona de interés coincide con la base de datos hará la detección.

2.6.10 *Detección de líneas laterales y paso cebra.*

Primeramente, se hace un análisis de la característica de esta señal lo que la diferencia de la demás es que no es una señal con una forma definida y no es una morfología tan estandarizada con las demás señales peatonales por lo que está marcada en la calle, por tanto, está expuesta a más desgaste por parte del entorno como lluvia, humedad, sol el paso de vehículos y peatones por lo que sus colores característicos tiende a perderse por lo tanto se dificulta a un más su detección.

2.6.10.1 Creación de matrices interlineadas.

El primer paso es crear 2 matrices que servirán para rellenar espacios vacíos debido a que esta señal es muy vulnerable a perder su color característico por razones y argumentos anteriormente mencionados. Para crear estas matrices se necesita de una imagen en cuyos valores sean cero es decir una imagen totalmente oscura, esta imagen se crea haciendo uso de la librería `numpy` que contiene un conjunto de funciones necesarias para analizar y procesar matrices.

La forma en cómo se usa esta función es la siguiente primeramente se establece que será una matriz de ceros, se establece el nombre de la variable y el tipo de datos que la contiene. También se debe obtener la dimensión de largo y ancho de la imagen para su análisis posterior.

Esto se efectúa con la función `shape` y se coloca después de la imagen seguida de un punto y debe almacenarse en dos variables `w` como ancho y `h` como largo, cabe mencionar que esta imagen debe estar en un solo canal es decir que debe estar compuesta por una sola matriz.

La matriz con líneas horizontales se realiza de la siguiente manera: Primero se crea un bucle con la estructura `for` que en este caso serán las filas luego dentro de este se crea otro que serán las columnas, al hacer esto se establece un control con la variable `if` el cual dejara pasar solo las filas cuyo número sea impar generando así un espacio en toda la fila dándole un color negro por cada iteración de fila hasta llegar a la variable `h` de largo. Por tanto, generara la siguiente imagen.



Figura 32-2: Interlineado horizontal
Realizado por: Oscar Segura, 2018

La matriz de líneas verticales se realiza tomando en cuenta las columnas de la imagen tomando en cuenta la variable de columna dentro de los 2 bucles de iteración `for` y de la misma forma haciendo una interacción impar generando espacios en las columnas.

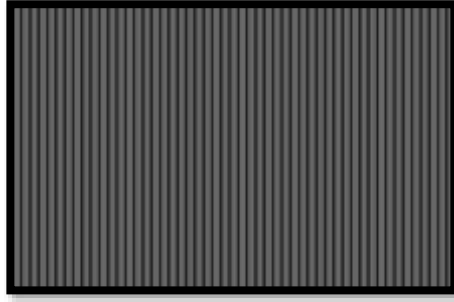


Figura 33-2: Interlineado vertical
Realizado por: Oscar Segura, 2018

2.6.10.2 Relleno de espacio para reconstrucción de imagen

Como sucedió con las imágenes anteriores se empieza con el filtro que permita extraer con más detalle la sección a analizar, para este caso específico se hará un filtro en tonalidades claras o blancas ya que líneas trazadas en la calle tiene este color. El tono al igual que con el rojo se puede establecer un rango en las tres componentes de color RGB dando un parámetro mínimo a máximo cuyo valor ya están contemplados. Luego de esto se extrae los canales más pronunciados y se hace la respectiva resta, y por último la umbralización será el punto de partida para desarrollar más procesos que permitan filtrar la imagen.



Figura 34-2: Reconstrucción de espacios
Realizado por: Oscar Segura, 2018

2.6.10.3 Filtrado por morfología.

El paso siguiente es eliminar objetos adyacentes o coordenadas de sus extremos muy cercano al borde de la imagen el comando para esto es el tuple que permite acceder a los elementos de cada contorno una vez hecho la extracción de los mismos, también se hace uso comando argmin y argmax para establecer las coordenadas de los extremos de cada contorno y estas son 4 el extremo superior e inferior y extremo izquierda derecha.

Estas deben estar dentro de una región previamente analizada. Luego se rellena la imagen y se invierte para no tener contornos sino objetos con relleno.



Figura 35-2: Filtrado pro morfología
Realizado por: Oscar Segura, 2018

Luego de esto se procese a encontrar patrones con forma lineales mediante la transformada de hough probabilística esta función no viene dentro de la librería de OpenCV sino dentro de otra skimage. Una vez encontradas las líneas se manda a dibujar en una segunda imagen por medio de la librería skimage, y dado que previamente se estableció en 0 estas imágenes servirán para hacer un análisis posterior.

Con estas 2 imágenes se extraen sus contornos y se analizan por separado cada una filtrando como en el proceso anterior, analizando sus extremos con los comandos anteriormente mencionados se rellenan con el comando floodfill y se invierten.

Al pasar por el siguiente filtro es donde se rellenan los espacios con las 2 imágenes que se generaron anteriormente tanto como en filas como en columnas haciendo un interlineado descrito anteriormente, esto se realiza haciendo uso de la operación nord, primero haciendo una operación or y luego negándola. El resultado serán los espacios entre esas 2 imágenes.



Figura 36-2: Reconstrucción total
Realizado por: Oscar Segura, 2018

Las imágenes resultantes serán filtradas y comparadas como se realizó anteriormente. Es importante mencionar que este espacio tiene una dimensión y área determinada ya que se crearía demasiado ruido en la imagen resultante. Es importante mencionar que no se hace análisis de morfología en esta parte debido a que el ruido no está totalmente eliminado.

Una vez filtrados los espacios para la reconstrucción total de la imagen se procese a hacer una operación or entre la imagen obtenida por filtro de color y la imagen obtenida en el proceso actual.

La imagen resultante pasara por último filtro para tomar una decisión si existe o no un paso cebra o a su vez un paso lateral. En la mayoría de las señales peatonales se usó el mismo proceso, con unas pequeñas variaciones por sus diferentes formas no tiene el mismo diagrama de flujo.



Figura 37-2: Resultado obtenido
Realizado por: Oscar Segura, 2018

2.6.11 Diagramas de flujos para el reconocimiento de las señales peatonales.

Los diagramas de flujo son de gran ayuda para una mejor guía al momento de realizar y entender los algoritmos usados para este programa. Un diagrama de flujo es una representación gráfica de un proceso.

2.6.11.1 Detección de próximo a paso lateral, semáforo y paso cebra.



Gráfico 2-2: Detección de próximo a paso lateral, semáforo y paso cebra.
Realizado por: Oscar Segura, 2018

2.6.11.2 *Detección de pare y semáforos.*



Gráfico 2-2: Detección de pare y semáforos.

Realizado por: Oscar Segura, 2018.

2.6.11.3 *Detección de líneas laterales y paso cebra.*



Gráfico 3-2: Detección de líneas laterales y paso cebra

Realizado por: Oscar Segura, 2018.

CAPÍTULO III

3. MARCO DE RESULTADOS

En este último capítulo se presentaron las pruebas y los resultados del prototipo construido. Donde se evaluó el algoritmo de detección de señales peatonales y su funcionamiento, se realizaron pruebas con imágenes de caracteres que se asemejen a las de una intercepción de esquinas, donde posteriormente fueron evaluadas con imágenes reales, en la cual el algoritmo debe evaluar si existen dichas señales para emular al prototipo.

3.1. Tiempo de respuesta del algoritmo.

Para determinar el tiempo de respuesta del algoritmo o la ejecución del mismo, se usó la función **time ()** del módulo estándar que lleva el mismo nombre (el **módulo time**), esta función nos retorna un valor de tipo flotante con el tiempo de ejecución del algoritmo, esta es una función de Python, permite medir el tiempo transcurrido durante la ejecución de un determinado código.

El tiempo de respuesta del algoritmo es de 4.3 segundos en promedio.

```
tiempo_final = time()
tiempo_ejecucion = tiempo_final - tiempo_inicial
print 'El tiempo de ejecución fue:', tiempo_ejecucion #En segundos
```

Figura 1-3: Tiempo de respuesta de algoritmo.

Realizado por: Oscar Segura, 2018

3.2. Desarrollo de pruebas del prototipo.

A continuación, se detallan las pruebas que se realizaron al prototipo para comprobar que los objetivos establecidos sean cumplidos y los resultados sean los esperados. Para ello se evalúan cada una de las señales de manera individual que en total son 8, para cada validación se realizó 20 tomas del objeto a evaluar es decir la señal peatonal, el número de pruebas escogido es alto para tener resultados más precisos y el porcentaje de validación del prototipo sea alto.

3.2.1. Experimento de validación del prototipo implementado.

La tabla 1-3, describe las pruebas de validación del prototipo se le realizó con la ayuda de un usuario, se validó la efectividad que tiene el prototipo para el reconocimiento de señales peatonales, como se puede verificar en la tabla se califica con “**si**”, si identifica el objeto y con “**no**”, si no identifica el mismo. Esto simplifica la redundancia en la explicación de la tabla.

3.2.1.1. Prueba de identificación de aproximación a paso cebra.

Tabla 1-3: Identificación de aproximación a paso cebra.

APROXIMACION A PASO CEBRA				
Prueba	Tiempo (segundos)	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	6	2,4	no	no
2	6	2,6	no	no
3	6	2,8	no	no
4	6	3	si	si
5	7	3,2	si	si
6	8	3,4	si	si
7	7	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	6	4,2	si	si
11	7	4,4	si	si
12	7	4,6	si	si
13	6	4,8	si	si
14	6	5	Si	Si
15	7	5,2	Si	Si
16	6	5,4	Si	Si
17	6	5,6	Si	Si
18	6	5,8	Si	Si
19	8	6	No	No
20	8	6,2	No	No

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 14 \cdot (100) / 20 = 75\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 14 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 75%. De las 20 pruebas 6 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 25% de falla en las pruebas establecidas, el error que se da se debe a la distancia que reconoce la señal.

3.2.1.2. Prueba de identificación del pare.

Tabla 2-3: Identificación del pare.

PRUEBAS DEL PARE				
Prueba	Tiempo (segundos)	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	5	2,5	no	no
2	5	2,7	no	no
3	6	2,9	no	no
4	7	3	si	si
5	6	3,2	si	si
6	6	3,4	si	si
7	6	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	7	4,2	si	si
11	7	4,4	si	si
12	6	4,6	si	si
13	6	4,8	si	si
14	6	5	si	si
15	6	5,2	si	si
16	5	5,4	si	si
17	5	5,6	si	si
18	5	5,8	si	si
19	7	6	si	si
20	6	6,2	no	no

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 14 \cdot (100) / 20 = 75\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 14 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 75%.

De las 20 pruebas 6 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 25% de falla en las pruebas establecidas.

3.2.1.3. Prueba de identificación de aproximación a semáforo.

Tabla 3-3: Identificación de aproximación a semáforo.

PRUEBA APROXIMACIÓN A SEMÁFORO				
Prueba	Tiempo (segundos)	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	6	2,4	no	no
2	6	2,6	no	no
3	6	2,8	no	no
4	6	3	si	si
5	7	3,2	si	si
6	7	3,4	si	si
7	7	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	6	4,2	si	si
11	7	4,4	si	si
12	7	4,6	si	si
13	6	4,8	si	si
14	6	5	si	si
15	7	5,2	si	si
16	6	5,4	si	si
17	6	5,6	si	si
18	6	5,8	si	si
19	7	6	si	si
20	5	6,2	no	no

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 15 \cdot (100) / 20 = 80\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 15 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 80%. De las 15 pruebas 5 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 20% de falla en las pruebas establecidas.

3.2.1.4. Prueba de identificación de aproximación a esquina.

Tabla 4-3: Identificación de peatones en la vía.

PRUEBAS DE PEATONES EN LA VIA(llegando a esquina)			
Prueba	Tiempo (segundos)	Distancia(metros)	SI/NO
1	6	2,5	no
2	8	2,7	no
3	6	2,9	no
4	6	3	si
5	6	3,2	si
6	6	3,4	si
7	4	3,6	si
8	4	3,8	si
9	6	4	si
10	6	4,2	si
11	6	4,4	si
12	8	4,6	si
13	6	4,8	si
14	6	5	si
15	7	5,2	si
16	6	5,4	si
17	6	5,6	si
18	6	5,8	si
19	7	6	si
20	5	6,2	si

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 17 * (100) / 20 = 90\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 13 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 90%. De las 20 pruebas 3 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 10% de falla en las pruebas establecidas.

3.2.1.5. Prueba de identificación de líneas paralelas.

Tabla 5-3: Identificación de líneas paralelas.

LINEAS PARALELAS				
Prueba	Tiempo (segundos)	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	6	2,4	si	si
2	6	2,6	si	si
3	6	2,8	si	si
4	6	3	si	si
5	7	3,2	si	si
6	8	3,4	si	si
7	7	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	6	4,2	no	no
11	7	4,4	no	no
12	7	4,6	no	no
13	6	4,8	no	no
14	6	5	no	no
15	7	5,2	no	no
16	6	5,4	no	no
17	6	5,6	no	no
18	6	5,8	no	no
19	8	6	no	no
20	8	6,2	no	no

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 9 \cdot (100) / 20 = 45\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 9 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 45%. De

las 20 pruebas 11 tuvieron un valor de 1 es decir un valor no favorable, con un valor porcentual del 55% de falla en las pruebas establecidas.

3.2.1.6. Prueba de identificación del paso cebra.

Tabla 6-3: Identificación del paso cebra.

PASO CEBRA				
Prueba	Tiempo (segundos)	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	6	2,4	si	si
2	6	2,6	si	si
3	5	2,8	si	si
4	6	3	si	si
5	7	3,2	si	si
6	8	3,4	si	si
7	7	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	6	4,2	si	si
11	7	4,4	si	si
12	7	4,6	no	no
13	6	4,8	no	no
14	6	5	no	no
15	7	5,2	no	no
16	6	5,4	no	no
17	6	5,6	no	no
18	6	5,8	no	no
19	6	6	no	no
20	8	6,2	no	no

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 11 \cdot (100) / 20 = 55\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 11 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 55%. De las 20 pruebas 9 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 45% de falla en las pruebas establecidas.

3.2.1.7. Prueba de identificación del semáforo peatonal en rojo.

Tabla 7-3: Identificación del semáforo peatonal en rojo.

SEMAFORO PEATONAL EN ROJO				
Prueba	Tiempo (segundos)	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	6	2,4	no	no
2	6	2,6	si	si
3	6	2,8	si	si
4	6	3	si	si
5	7	3,2	si	si
6	8	3,4	si	si
7	7	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	6	4,2	si	si
11	7	4,4	si	si
12	7	4,6	si	si
13	6	4,8	si	si
14	6	5	si	si
15	7	5,2	si	si
16	6	5,4	si	si
17	6	5,6	no	no
18	6	5,8	no	no
19	8	6	no	no
20	8	6,2	no	no

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 14 \cdot (100) / 20 = 75\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 14 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 75%. De las 20 pruebas 6 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 25% de falla en las pruebas establecidas, el error que se da se debe a la distancia de la señal.

3.2.1.8. Prueba de identificación del semáforo peatonal en verde.

Tabla 8-3: Identificación del pare.

SEMAFORO PEATONAL EN VERDE				
Prueba	Tiempo (segundos)I	Distancia(metros)	Intento 1 SI/NO	Intento 2 SI/NO
1	6	2,4	no	no
2	6	2,6	no	no
3	6	2,8	si	no
4	6	3	si	si
5	7	3,2	si	si
6	8	3,4	si	si
7	7	3,6	si	si
8	7	3,8	si	si
9	6	4	si	si
10	6	4,2	si	si
11	7	4,4	si	si
12	7	4,6	si	si
13	6	4,8	si	si
14	6	5	si	si
15	7	5,2	si	si
16	6	5,4	no	no
17	6	5,6	no	no
18	6	5,8	no	no
19	8	6	no	
20	8	6,2	no	

Realizado por: Oscar Segura.

Media aritmética porcentual $x = 14 * (100) / 20 = 70\%$

En los resultados de la tabla se pudo terminar que, de un total de 20 pruebas, 14 de ellas se tuvo un valor de 1 es decir favorable, esto quiere decir que la efectividad del prototipo es de un 75%. De las 20 pruebas 6 tuvieron un valor de 0 es decir un valor no favorable, con un valor porcentual del 25% de falla en las pruebas establecidas, el error que se da se debe a la distancia que reconoce la señal.

3.3. Análisis de confiabilidad del sistema.

Se usó como técnica de recolección de datos la observación experimental ya que se realizaron pruebas en base a la experimentación final del sistema para esto se emplea como técnica de recolección de datos una hoja de registro cuando el sistema detecta la señal peatonal en la imagen.

El sistema cuenta con una población de 160 elementos de los cuales se determinó una muestra de 113 elementos mediante la siguiente fórmula estadística:

$$n = \frac{Z^2 * p * q * N}{e^2 * (N - 1) + Z^2 * p * q}$$

Donde:

n: tamaño de la muestra

N: tamaño de la población o universo

Z: constante que depende del nivel de confianza que se asigne

p: probabilidad de éxito

q: probabilidad de fracaso

e: error muestral deseado en tanto por ciento

Para determinar el valor del nivel de confianza tenemos la siguiente tabla de datos, esta tabla fue creada por Carolina Jara y Carlos Caba quienes también realizaron pruebas a su prototipo que efectúa reconocimiento de imágenes, cuya tabla de nivel de confianza sirvió de referencia para las pruebas del prototipo.

Tabla 9-3: Nivel de confianza.

Valor Z	1,645	1,96	2,24	2,576
Nivel confianza	90%	95%	97,5%	99%

Realizado por: JARA, Caroline; CABA, Carlos 2018.

Para el sistema se consideró los intentos que se hicieron para la detección de las señales, por lo que es necesario establecer un nivel de confianza del 95% considerándolo como aceptable del cual se obtiene un valor de Z de 1,96 el cual fue usado para el cálculo de la muestra.

El error que se puede aceptar en el modelo es del 5% ya se le asigna un nivel de confianza al sistema del 95%

La probabilidad de éxito de que una persona sea reconocida es del 50% lo que significa que mediante la fórmula estadística obtendremos el valor de q de la siguiente manera:

$$p=1-q$$

$$q=1-p$$

$$q=1-0,5$$

$$q=0,5$$

Obtenidos todos los datos necesarios se procede a calcular la muestra necesaria requerida para el sistema.

$$n = \frac{(1,96)^2 * (0,5) * (0,5) * (160)}{(0,05)^2 * (160 - 1) + (1,96)^2 * (0,5) * (0,5)}$$

$$n= 113,16$$

$$n=113 \text{ individuos}$$

El siguiente paso es recolectar la información necesaria para la muestra calculada para lo cual se considera como aceptable el reconocimiento hasta dos intentos y el tercero se considera como no aceptable.

En la siguiente tabla 6-3 se muestra los datos obtenidos aleatoriamente:

Tabla 10-3: Datos obtenidos del funcionamiento del sistema.

PRUEBA	RECONOCE		NO RECONOCE
	Intento 1	Intento 2	
1	si	si	
2	si		
3		si	
4			no

5	si	si	
6		si	
7			no
8	si		
9			no
10	si		
11	si	si	
12			
13	si	si	
14			no
15	si	si	
16			
17			
18			
19			no
20	si		
21			no
22			
23	si		
24	si	si	
25	si		
26	si		
27			no
28		si	
29	si		
30			no
31	si		
32		si	
33	si		
34			no
35		si	
36			no
37		si	
38			
39	si		
40	si		
41			no
42	si	si	
43	si	si	
44			no
45			no
46		si	
47	si		
48		si	
49			no

50	si		
51			
52	si		
53		si	
54	si		
55			no
56		si	
57			no
58		si	
59	si		
60	si		
61	si	si	
62			no
63	si		
64	si	si	
65			
66		si	
67	si		
68	si		
69		si	
70		si	
71	si		
72			
73	si		
74		si	
75	si	si	
76			no
77			
78	si	si	
79			
80	si		
81			
82	si	si	
83		si	
84	si	si	
85			no
86	si	si	
87	si		
88			
89	si		
90	si	si	
91			no
92			no
93		si	
94		si	

95	si	si	
96	si		
97	si		
98			no
99	si	si	
100			no
101	si	si	
102	si		
103			
104	si	si	
105			no
106	si		
107	si		
108		si	
109			
110	si		
111	si		
112	si		
113			no
Suma	49	38	26
Total	0.4336	0.3362	0.23
Total %	43.36%	33.62%	23%
	77%		23%

Realizado por: Oscar Segura.

Mediante el análisis experimental realizado al sistema se llegó a la conclusión que existe un 77% de aciertos en el reconocimiento que se considera aceptable para los sistemas de reconocimiento existentes. Se puede observar en el gráfico 1-3 el resultado obtenido del análisis del sistema.

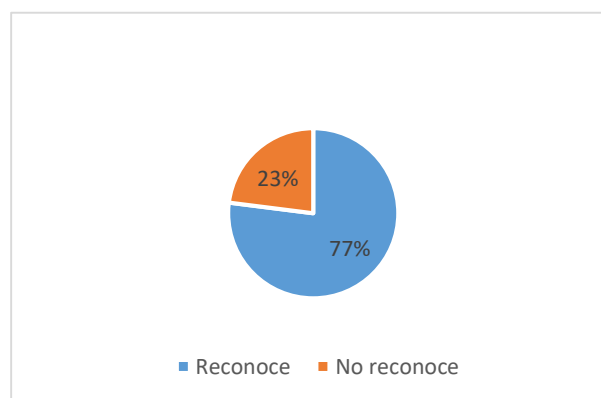


Gráfico 1-3: Porcentaje de funcionamiento del sistema.

Realizado por: Oscar Segura, 2018.

3.4. Costo del prototipo.

Dentro de los costos directos de fabricación tenemos todos aquellos que forman parte de la construcción e implementación del prototipo electrónico, todos los valores correspondientes a cada parte del sistema se informan en la Tabla 13-4:

Tabla 11-3: Costos directos de fabricación del prototipo electrónico.

Cantidad	Descripción	Precio (USD)
1	Raspberry Pi 3	45,00
1	Cable HDMI	7,00
1	Micro SD	10,00
1	Cámara WideCam F100	55,00
1	Power Bank 2600 [mAh]	25,00
1	Cargador de 5V	7,00
2	Cables USB	3,00
1	Auriculares	5,00
	Total	157,00

Realizado por: Oscar Segura, 2019

CONCLUSIONES.

- El sistema desarrollado en este trabajo de titulación, permitió implementar un dispositivo electrónico de ayuda a personas invidentes como asistente para el cruce de calles urbanas, brindando seguridad a un grupo vulnerable de nuestra comunidad.
- Durante el diseño del sistema, se definieron los requerimientos de funcionamiento que el dispositivo electrónico debía cumplir. Esto en base al criterio de personas con discapacidad visual que concluyeron en qué momento se sienten más vulnerables para desplazarse en la calle, llegándose a determinar que son las esquinas; por esa razón es este lugar de la vía pública el lugar seleccionado para el funcionamiento del dispositivo.
- Mediante la aplicación de técnicas de Visión Artificial, se creó un prototipo electrónico de alerta para el cruce de calles a personas con discapacidad visual. Se utilizaron métodos y técnicas de pre procesamiento y procesamiento de imágenes que permitieron comparar la imagen capturada en tiempo real, con una serie de imágenes indexadas al programa para que mediante un mensaje auditivo dar a conocer al usuario del sistema la señal detectada.
- Se pudo comprobar que el prototipo cumple con el reconocimiento de señales peatonales de tránsito, alertando mediante un mensaje de voz la proximidad de las mismas. Este sistema electrónico servirá a futuro para evitar accidentes de tránsito a personas con discapacidad visual.
- Las pruebas realizadas al prototipo para el reconocimiento de cada una de las señales peatonales, dieron valores porcentuales altos en su mayoría dando como resultado un valor porcentual total de 77% dando una acorde aprobación al algoritmo.

RECOMENDACIONES.

- A futuro se debe implementar un prototipo que realice una detección de las señales peatonales sin importar la calidad, limpieza o nitidez que tengan. El algoritmo debe detectar las señales sin importar esas anomalías en las señales.
- Para obtener mejores resultados en la detección de imágenes se recomienda utilizar a futuro otros métodos de reconocimiento con otros tipos de librerías y funciones. Esto implica a su vez, usar una tarjeta de desarrollo con mayor capacidad de procesamiento para obtener una respuesta más rápida del sistema.
- Se debe tener en cuenta el voltaje con el que funcionan los GPIO de la Raspberry para evitar daños en el procesador, por eso es recomendable usar baterías comerciales que abastecen las necesidades del prototipo.
- Realizar estudios para crear un dispositivo que funcione bajo condiciones adversas como la lluvia, niveles de temperatura elevada, baja luminosidad y que el prototipo funcione en la noche si detectar falsos positivos.
- Crear un algoritmo que pueda detectar de mejor forma las señales de paso cebra y líneas paralelas, que no se encuentran bien plasmadas en la calzada, de esa manera el prototipo tendrá una mayor capacidad para detectar las señales plasmadas en la calzada.

BIBLIOGRAFÍA

CAVALIERI, MARTINA. Blind. tech. [En línea] 2015. [Citado el: 12 de 10 de 2018.] <http://blind.tech/seelight-the-app-that-helping-blind-people-navigate-in-the-city/>.

CONADIS. ESTADÍSTICAS DE DISCAPACIDAD- CONSEJO NACIONAL PARA LA IGUALDAD Y DISCAPACIDAD . [En línea] 2008. [Citado el: 30 de 07 de 2018.] <https://www.consejodiscapacidades.gob.ec/estadisticas-de-discapacidad/>.

HANDBOOK. Blind People's Association India. Visual Impairment. [En línea] 2018. [Citado el: 04 de 10 de 2018.] <http://www.bpaindia.org/VIB%20Chapter-IV.pdf>.

HARDZONE. Cómo configurar y usar el módulo de cámara en la Raspberry Pi. [En línea] 2018. [Citado el: 18 de 10 de 2018.] <https://hardzone.es/2018/08/25/como-configurar-modulo-camara-raspberry-pi-3/>.

INFOTECNOVISION. Aira: acceso a la información de personas ciegas. [En línea] 2017. [Citado el: 27 de 09 de 2018.] <https://www.infotecnovision.com/aira-un-nuevo-servicio-de-asistencia-remota-para-el-guiado-y-acceso-a-la-informacion-de-personas-ciegas/>.

LENGUA, ISMAEL; et al. [En línea] 2015. [Citado el: 13 de 04 de 2018.] <http://www.scielo.org.co/pdf/dyna/v80n179/v80n179a04.pdf>.

MARTINSANZ, PAJARES; et al. *Visión por computador, imágenes digitales y aplicaciones.* MADRID : s.n., 2007.

OPENCV. BIBLIOTECA OPENCV. [En línea] 2017. [Citado el: 16 de 10 de 2018.] <https://opencv.org/>.

ORGANIZACIÓN MUNDIAL DE LA SALUD. DISCAPACIDADES FÍSICAS. [En línea] 2011. [Citado el: 07 de 06 de 2018.] <http://www.who.int/topics/blindness/es/index.html>.

ORGANIZACIÓN, DE LOS ESTADOS AMERICANOS. Departamento de Desarrollo Social y Empleo. [En línea] 2017. [Citado el: 22 de 09 de 2018.] <http://www.oas.org/es/sedi/ddse/paginas/index-4.asp>.

ORGANIZACION MUNDUAL DE LA SALUD. DISCAPACIDAD VISUAL. [En línea] 2015. [Citado el: 27 de 09 de 2018.] https://www.paho.org/uru/index.php?option=com_content&view=article&id=451:285-millones-personas-discapacidad-visual-segun-oms&Itemid=24.

PARTNERS. MWCcapital. [En línea] 2015. [Citado el: 02 de 10 de 2018.] <http://mobileworldcapital.com/es/2015/06/01/lazzus-la-app-guia-para-invidentes/>.

PEREZ, VALDEZ, et.al. Und Einem ordentlichen Register der Psalmen. [En línea] 2009. [Citado el: 17 de 09 de 2018.] https://www.academia.edu/16523506/Info_Sistemas_Embebidos.

PRONADIS. Orientación y Movilidad: estrategias frente a nuevos escenarios poblacionales. [En línea] 2017. [Citado el: 02 de 10 de 2018.] <http://pronadis.mides.gub.uy/innovaportal/v/23113/9/innova.front/orientacion-y-movilidad:-estrategias-frente-a--nuevos-escenarios-poblacionales>.

RASPBERRY, PI. Raspberry Pi - Enseña, aprende y crea con Raspberry Pi. [En línea] 2018. [Citado el: 17 de 10 de 2018.] <https://www.raspberrypi.org/>.

REVUELTA, PABLO, MEZCUA; et al. Libertad Digital. [En línea] 2014. [Citado el: 12 de 10 de 2018.] <https://www.libertaddigital.com/ciencia-tecnologia/tecnologia/2014-01-13/disenan-un-dispositivo-de-ayuda-para-los-ciegos-1276508154/>.

SALAZAR, MARCO. Centaurina. [En línea] 16 de 08 de 2018. [Citado el: 12 de 09 de 2018.]

SALGADO, LUIS. *Visión Artificial: Fundamentos y Aplicaciones*. Madrid : Universidad Politécnica de Madrid, 2007.

TRÁNSITO, AGENCIA NACIONAL DE. PRESENTACIÓN DEL MANUAL DE CONDUCCIÓN. [En línea] 2017. [Citado el: 14 de 10 de 2018.] [http://www.ant.gob.ec/phocadownload/Documentos/manual de conduccion - 2018.pdf](http://www.ant.gob.ec/phocadownload/Documentos/manual%20de%20conduccion%20-2018.pdf).

UNIVERSIA, DISCAPACIDAD. universitarios.universia.es.ed. [En línea] 2014. [Citado el: 12 de 07 de 2018.] <http://universitarios.universia.es/voluntariado/discapacidad/discapacidadvisual/discapacidad-visual.pdf>.

VÉLEZ, JOSE. INTRODUCCIÓN A LA VISIÓN ARTIFICIAL. [En línea] 2007. [Citado el: 16 de 10 de 2018.] <http://www.visionporcomputador.es/libroVision/VisionPorComputador.pdf>.

VISUAL. EDUCACIÓN INCLUSIVA: DISCAPACIDAD VISUAL SECRETARÍA DE ESTADO DE EDUCACIÓN Y FORMACIÓN PROFESIONAL DIRECCIÓN GENERAL DE FORMACIÓN PROFESIONAL MINISTERIO DE EDUCACIÓN MÓDULO 6: AUTONOMÍA PERSONAL. [En línea] 2016. [Citado el: 02 de 10 de 2018.] http://www.ite.educacion.es/formacion/materiales/129/cd/pdf/m6_dv.pdf.

WALYOU. Polymythic. [En línea] 2010. [Citado el: 12 de 10 de 2018.] <http://walyou.com/blog/2010/12/17/haptic-assistance-for-the-blind/>.

ANEXOS

Anexo A: Paso a paso la instalar del sistema operativo en la Raspberry Pi

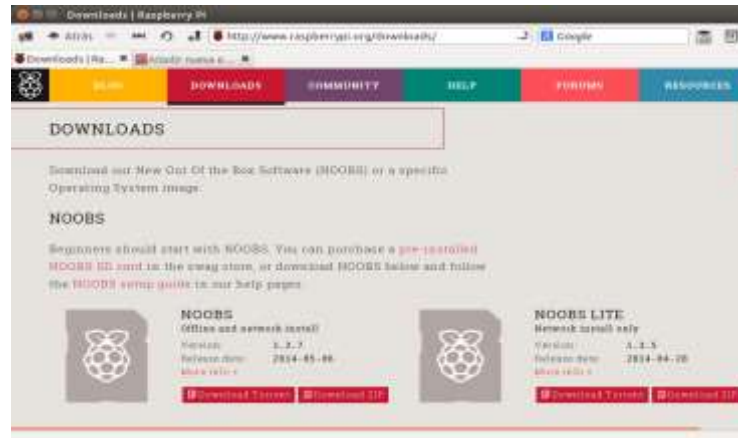


Figura 1-A: Descargar NOOBS en el PC

Realizado por: Oscar Segura, 2018.

Los pasos a dar serán:

1. **descargar NOOBS** en nuestro PC
2. insertar o **conectar la tarjeta SD** en ese mismo PC
3. **formatear la tarjeta SD**
4. **descomprimir NOOBS** que descargamos como fichero **zip** en esa SD
5. una vez **termina de copiarse** todo sacamos la tarjeta SD y **la insertamos en la Raspberry Pi**
6. conectamos el cable HDMI, teclado, ratón y cable de red. **OJO todavía no funcionará con Wi-Fi**
7. **conectamos el cable de alimentación eléctrica** y empezará a arrancar la Raspberry Pi

Figura 2-A: Pasos a seguir para instalar NOOBOS en la tarjeta SD

Realizado por: Oscar Segura, 2018



Figura 3-A: Pasos a seguir para instalar NOOBOS en la tarjeta SD

Realizado por: Oscar Segura, 2018.

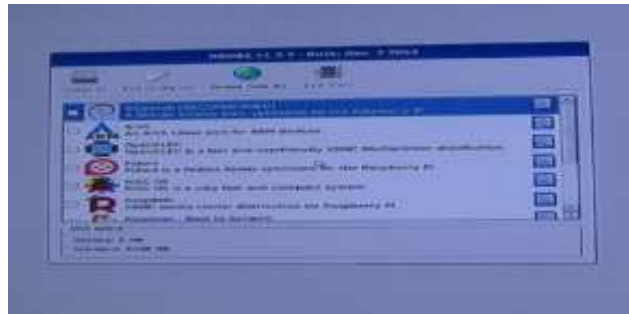


Figura 4-A: Menú con los sistemas operativos disponibles para instalar
Realizado por: Oscar Segura, 2018.



Figura 5-A: Proceso de instalación del sistema operativo
Realizado por: Oscar Segura, 2018.

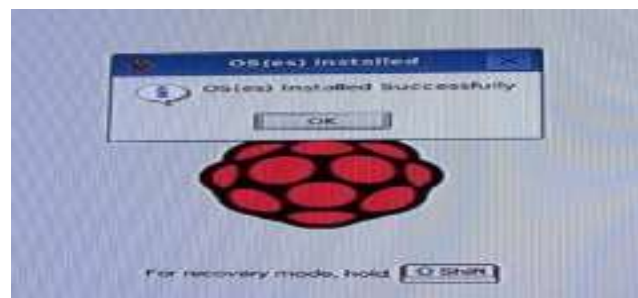


Figura 6-A: Culminación de instalación del sistema operativo
Realizado por: Oscar Segura, 2018.

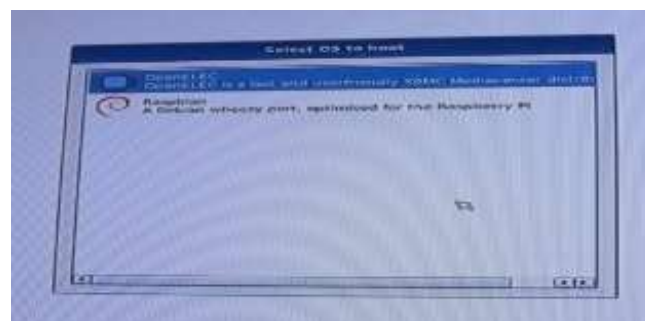


Figura 7-A: Menú de arranque del sistema operativo
Realizado por: Oscar Segura, 2018.

Anexo B: Fotografías del prototipo y de las pruebas realizadas



Figura 1-B: Prototipo

Realizado por: Oscar Segura, 2018.



Figura 2-B: Pruebas de funcionamiento

Realizado por: Oscar Segura, 2018.



Figura 3-B: Pruebas de funcionamiento y audio

Realizado por: Oscar Segura, 2018.



Figura 4-B: Pruebas de distancia.
Realizado por: Oscar Segura, 2018.