



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

“IMPLEMENTACIÓN DE UN PROTOTIPO DE DISPOSITIVO ELECTRÓNICO QUE MEDIANTE EL PROCESAMIENTO DE IMÁGENES CON VISIÓN ARTIFICIAL MUESTRE LA INFORMACIÓN DE CIRCUITOS TURÍSTICOS DEL CENTRO HISTÓRICO DE LA CIUDAD DE RIOBAMBA”

Trabajo de titulación

Tipo: Dispositivo tecnológico

Presentado para optar el grado académico de:

INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORES: BERRONES ORTIZ ALVARO ALEJANDRO

CRESPATA ALMACHI CRISTIAN SANTIAGO

DIRECTOR: ING. LOZADA YÁNEZ PABLO EDUARDO

Riobamba – Ecuador

2021

© 2021, Cristian Santiago Crespata Almachi; & Alvaro Alejandro Berrones Ortiz

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

Nosotros, Cristian Santiago Crespata Almachi Y Alvaro Alejandro Berrones Ortiz, declaramos que el presente trabajo de titulación es de nuestra autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autores asumimos la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica De Chimborazo

Riobamba, 09 de septiembre de 2021



Cristian Santiago Crespata Almachi

050418666-9



Alvaro Alejandro Berrones Ortiz

060583408-4

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

El Tribunal del Trabajo de Titulación certifica que: El trabajo de titulación: Tipo Dispositivo Tecnológico **“IMPLEMENTACIÓN DE UN PROTOTIPO DE DISPOSITIVO ELECTRÓNICO QUE MEDIANTE EL PROCESAMIENTO DE IMÁGENES CON VISIÓN ARTIFICIAL MUESTRE LA INFORMACIÓN DE CIRCUITOS TURÍSTICOS DEL CENTRO HISTÓRICO DE LA CIUDAD DE RIOBAMBA”**, realizado por los señores **CRISTIAN SANTIAGO CRESPATA ALMACHI Y ALVARO ALEJANDRO BERRONES ORTIZ** , ha sido minuciosamente revisado por los Miembros del Trabajo de Titulación, el mismo que cumple con los requisitos científicos, técnicos, legales, en tal virtud el Tribunal Autoriza su presentación.

	FIRMA	FECHA
Ing. José Enrique Guerra Salazar PRESIDENTE DEL TRIBUNAL	_____	2021-09-09
Ing. Pablo Eduardo Lozada Yáñez DIRECTOR DE TRABAJO DE TITULACIÓN	_____	2021-09-09
Ing. Fausto Ramiro Cabrera Aguayo MIEMBRO DEL TRIBUNAL	_____	2021-09-09

DEDICATORIA

A mis padres por ser esas personas que han estado a lo largo de este trayecto de mi vida, brindándome su apoyo, consejos y formándome con valores éticos y morales los cuales han sido los cimientos para llegar hacer la persona que soy actualmente, por mostrarme el esfuerzo necesario que se necesita para cumplir las metas que uno se propone, por enseñarme a que con esfuerzo y sacrificio podemos llegar a lo más alto.

Cristian

Este trabajo se lo dedico a mis padres Irma y Cesar, quienes fueron un pilar fundamental para mi formación, tanto como persona y como futuro profesional, con sus consejos y su apoyo incondicional. A mi hermana Samanta que siempre me animaba a no rendirme, a mi hermano Cristian, que con sus ocurrencias me motivaba a seguir adelante. A mi abuelito Benigno y a mi abuelita Cleo que nunca dudaron en apoyarme y siempre estaban en todo momento, a mis familiares y amigos que siempre con su aliento me incentivaban a seguir adelante para poder culminar mi carrera como ingeniero y con este trabajo de titulación.

Alvaro

AGRADECIMIENTO

Quiero expresar un infinito agradecimiento a mis padres quienes inculcaron en mí valores éticos y morales, los mismos que me han guiado en toda mi trayectoria como estudiante, de la misma manera quiero expresar un eterno agradecimiento a la ilustre Escuela Superior Politécnica de Chimborazo que me abrió las puertas y me dio la oportunidad de formarme como un profesional.

Cristian

Expreso un profundo agradecimiento a Dios por darme la oportunidad de poder vivir para lograr cumplir con mis metas y objetivos. A mis padres por ser unas personas de bien, que me incentivaron a seguir hacia adelante y brindarme todo su amor y paciencia. A mis hermanos, que siempre pude contar con ellos y a toda mi familia. Les agradezco a mis amigos de San Andrés, a mis amigos del colegio en el que me gradué y los que pude hacer durante la carrera, con quienes siempre pude confiar y estaban presentes en las situaciones que más necesitaba. Mi agradecimiento también lo dirijo a la Escuela Superior Politécnica de Chimborazo, a los docentes de la carrera de Ingeniería Electrónica en Control y Redes Industriales, quienes mediante su enseñanza y sabiduría fueron un pilar fundamental para poder lograr mi objetivo como ingeniero.

Alvaro

TABLA DE CONTENIDO

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE GRÁFICOS.....	xv
ÍNDICE DE ANEXOS.....	xvi
ÍNDICE DE ABREVIATURAS.....	xvii
RESUMEN.....	xviii
SUMMARY	xix
INTRODUCCIÓN	1
CAPÍTULO I	
1 MARCO TEÓRICO	6
1.1 Circuitos turísticos	6
<i>1.1.1 Lugares turísticos o culturales en la ciudad de Riobamba</i>	<i>7</i>
<i>1.1.2 Señalética turística.....</i>	<i>9</i>
<i>1.1.2.1 Pictogramas</i>	<i>10</i>
1.2 Visión artificial	11
1.3 Procesamiento Digital de Imágenes (PDI)	12
<i>1.3.1 La imagen digital</i>	<i>13</i>
<i>1.3.2 Etapas de procesamiento digital de imágenes.....</i>	<i>13</i>
<i>1.3.2.1 Captura</i>	<i>13</i>
<i>1.3.2.2 Pre-Procesamiento.....</i>	<i>14</i>
<i>1.3.2.3 Segmentación</i>	<i>17</i>
<i>1.3.2.4 Extracción de características.....</i>	<i>18</i>
<i>1.3.2.5 Identificación de la señalética</i>	<i>20</i>
1.4 Hardware	21
<i>1.4.1 Tarjetas controladoras.....</i>	<i>22</i>
<i>1.4.2 Cámaras</i>	<i>23</i>

1.4.3	<i>Pantalla táctil</i>	24
1.4.4	<i>Batería</i>	25
1.4.5	<i>GPS</i>	27
1.5	Software	28
1.5.1	<i>Sistema operativo</i>	29
1.5.2	<i>Lenguaje de programación</i>	30

CAPÍTULO II

2	PROPUESTA Y DISEÑO DE PROTOTIPO	32
2.1	Requerimientos del sistema	32
2.2	Concepción general del sistema	33
2.3	Selección de los componentes necesarios para el hardware del prototipo	34
2.3.1	<i>Hardware del prototipo</i>	34
2.3.1.1	<i>Tarjeta Raspberry Pi 4B</i>	34
2.3.1.2	<i>Cámara Pi NoIR V2</i>	35
2.3.1.3	<i>Pantalla LCD táctil Kuman 3,5" 480X320 HDMI</i>	36
2.3.1.4	<i>Módulo GPS NEO-6M</i>	37
2.3.2	<i>Estructura del dispositivo</i>	38
2.3.3	<i>Esquema general de conexiones del prototipo</i>	41
2.4	Diseño del software	42
2.4.1	<i>Etapa de adquisición</i>	42
2.4.2	<i>Etapa de procesamiento</i>	42
2.4.3	<i>Etapa de interpretación de resultados</i>	43
2.5	Software	44
2.5.1	<i>Software del prototipo</i>	44
2.6	Diseño de la interfaz del prototipo	51
2.6.1	<i>Diseño de interfaz y botones para las pantallas</i>	51

CAPÍTULO III

3	VALIDACIÓN DEL PROTOTIPO	56
3.1	Descripción de las pruebas realizadas.....	56
3.2	Prueba de la etapa de adquisición de imágenes.....	56
3.2.1	<i>Pruebas a diferentes niveles de iluminación</i>	<i>56</i>
3.2.2	<i>Pruebas a diferentes distancias del prototipo a la señalética.....</i>	<i>62</i>
3.3	Pruebas del trazado de rutas y de la información del lugar de destino.	68
3.3.1	<i>Trazado de rutas</i>	<i>68</i>
3.3.2	<i>Evaluación de la información obtenida.....</i>	<i>69</i>
3.4	Tiempo de respuesta	71
3.4.1	<i>Pruebas de campo con las señaléticas del entorno.....</i>	<i>72</i>
3.5	Análisis de autonomía del dispositivo y batería.....	76

CAPÍTULO IV

4	EVALUACIÓN ECONÓMICA	80
4.1	Análisis de costos de materiales	80
4.2	Análisis de costos de equipos.....	80
4.3	Costo total del dispositivo.....	81
4.4	Tabla comparativa de <i>softwares</i> de aplicaciones y prototipo	81
	CONCLUSIONES.....	83
	RECOMENDACIONES.....	84

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

Tabla 1-1:	Límites del Centro Histórico de la ciudad de Riobamba.	7
Tabla 2-1:	Atractivos turísticos del Centro Histórico de Riobamba.	8
Tabla 3-1:	Tipos de filtros.....	16
Tabla 4-1:	Tipos de segmentación.	18
Tabla 5-1:	Algoritmos para la extracción de características.	19
Tabla 6-1:	Comparación de rendimiento entre descriptores SIFT y SURF.	20
Tabla 7-1:	Algoritmos para la comparación de características.	21
Tabla 8-1:	Características de tarjetas controladoras.....	22
Tabla 9-1:	Comparación entre familias de cámaras.	23
Tabla 10-1:	Características de pantallas táctiles Raspberry.....	25
Tabla 11-1:	Características de distintos tipos de baterías.	26
Tabla 12-1:	Características de módulos GPS.....	28
Tabla 13-1:	Ventajas y desventajas de los sistemas operativos.	29
Tabla 14-1:	Lenguajes de programación.....	30
Tabla 1-2:	Características de la tarjeta controladora Raspberry Pi 4B	34
Tabla 2-2:	Característica de la Camara Pi NoIR V2.....	36
Tabla 3-2:	Características de la pantalla LCD táctil Kuman 3.5" 480X320 HDMI.....	36
Tabla 4-2:	Características del Módulo GPS NEO-6M.....	37
Tabla 5-2:	Conexión de terminales entre la tarjeta Raspberry y elementos.....	41
Tabla 6-2:	Ejemplo de la información que contiene la tabla en Excel.....	48
Tabla 1-3:	Niveles de iluminación aproximados.	57
Tabla 2-3:	Niveles de iluminación a diferentes horas del día.	57
Tabla 3-3:	Pruebas al algoritmo con diferentes tipos de iluminación.....	60
Tabla 4-3:	Total de pruebas de identificación a diferentes tipos de iluminación.....	61
Tabla 5-3:	Diferentes distancias entre la señalética y el usuario.....	63
Tabla 6-3:	Prueba de distancia de detección de algoritmo de visión con $D_m = 3m$	64

Tabla 7-3:	Prueba de distancia de detección de algoritmo de visión con $D_m=2m$	65
Tabla 8-3:	Prueba de distancia de detección de algoritmo de visión con $D_m= 1m$	66
Tabla 9-3:	Prueba de distancia de detección de algoritmo de visión con $D_m<1m$	67
Tabla 10-3:	Pruebas de la información obtenida por el dispositivo.	70
Tabla 11-3:	Tiempo para la identificación de iglesias.	72
Tabla 12-3:	Tiempo para la identificación de museos.	73
Tabla 13-3:	Tiempo para la identificación de parques.	73
Tabla 14-3:	Tiempo para la identificación de mercados.	74
Tabla 15-3:	Tiempo para la identificación de Patrimonios Arquitectónicos.	74
Tabla 16-3:	Tiempo para la identificación de la Plaza de Toros.	75
Tabla 17-3:	Consumo de Stand-by o reposo.	76
Tabla 18-3:	Consumo de corriente en funcionamiento.	76
Tabla 19-3:	Consumo, potencia y tiempo de autonomía.	77
Tabla 20-3:	Dimensiones de las baterías.	77
Tabla 21-3:	Características de la batería LiPO 2S 1000 mAh.	78
Tabla 1-4:	Costos de materiales.	80
Tabla 2-4:	Costos de equipos.	80
Tabla 3-4:	Costo total del dispositivo.	81
Tabla 4-4 :	Tabla comparativa de <i>software</i> de aplicaciones con prototipo.	82

ÍNDICE DE FIGURAS

Figura 1-1:	Zonificación del Centro Histórico de la ciudad de Riobamba.	7
Figura 2-1:	Ruta de las iglesias.....	9
Figura 3-1:	Señalética de información turística.	9
Figura 4-1:	Pictogramas de atractivos culturales.....	10
Figura 5-1:	Elementos que constituyen un sistema de visión artificial.....	11
Figura 6-1:	Fuente de iluminación natural (Sol).....	12
Figura 7-1:	Convección de coordenadas.....	13
Figura 8-1:	Etapas del procesamiento digital de imágenes.....	13
Figura 9-1:	Imagen original/ con ruido impulsivo.....	15
Figura 10-1:	Ruido “Gaussiano”.....	15
Figura 11-1:	Operación filtro mediana.	16
Figura 12-1:	Pantallas táctil LCD.....	24
Figura 13-1:	Batería Lipo.	26
Figura 14-1:	Modulo GPS.....	27
Figura 1-2:	Concepción general del sistema.....	33
Figura 2-2:	Tarjeta Raspberry Pi 4B.....	35
Figura 3-2:	Pi NoIR Camera V2.....	35
Figura 4-2:	Pantalla LCD TÁCTIL Kuman 3.5" 480X320 HDMI.....	37
Figura 5-2:	Módulo GPS NEO-6M.	37
Figura 6-2:	Diseño del compartimento principal 3D de la estructura.....	38
Figura 7-2:	Diseño de la tapa 3D del compartimento principal de la estructura.....	39
Figura 8-2:	Montaje de la estructura en 3D.	39
Figura 9-2:	Estructura armada en 3D.....	40
Figura 10-2:	Estructura montada y armada.....	40
Figura 11-2:	Esquema electrónico 3D.	41
Figura 12-2:	Etapas de adquisición de imagen.....	42

Figura 13-2:	Etapa de procesamiento.	43
Figura 14-2:	Etapa de interpretación de resultados.....	43
Figura 15-2:	Diagrama de flujo general de funcionamiento.....	46
Figura 16-2:	Diagrama de flujo de la Función OCR.....	47
Figura 17-2:	Diagrama de flujo función CONFIGURACION().....	48
Figura 18-2:	Función MOSTRAR RUTA().....	49
Figura 19-2:	Función NAVEGACIÓN().....	50
Figura 20-2:	Diagrama de flujo función INFORMACION().....	51
Figura 21-2:	Diseño de la interfaz de la pantalla principal.....	52
Figura 22-2:	Botón tomar imagen.....	52
Figura 23-2:	Interfaz de la pantalla principal.....	53
Figura 24-2:	Interfaz de la pantalla Base de datos de imágenes.....	53
Figura 25-2:	Interfaz de la pantalla captura de imágenes.....	53
Figura 26-2:	Interfaz de la pantalla Menú de rutas.....	54
Figura 27-2:	Interfaz de la pantalla Ruta.....	55
Figura 28-2:	Interfaz de la pantalla Información.....	55
Figura 1-3:	Medición de luxes para la tabla 2-3.....	57
Figura 2-3:	Captura a 45000 lx.....	58
Figura 3-3:	Captura a 120000 lx.....	59
Figura 4-3:	Captura a 120 lx.....	59
Figura 5-3:	Representación de la distancia entre el prototipo y la señalética.....	62
Figura 6-3:	Ruta para el destino del parque La Libertad con el prototipo.....	69
Figura 7-3:	Ruta con aplicación del parque La Libertad con Waze.....	69
Figura 8-3:	Información sobre el Parque Sucre.....	70
Figura 9-3:	Código para obtener el tiempo de respuesta.....	71

Figura 10-3:	Ejecución del código para obtener el tiempo de respuesta.	71
Figura 11-3:	Batería LiPO, 2 celdas 1000mAh.	78

ÍNDICE DE GRÁFICOS

Gráfico 1-3:	Total de imágenes analizadas.	61
Gráfico 2-3:	Imágenes analizadas a Dm igual a 3 metros.	64
Gráfico 3-3:	Imágenes analizadas a Dm igual a 2 metros.	65
Gráfico 4-3:	Imágenes analizadas a Dm igual a 1 metro.	66
Gráfico 5-3:	Imágenes analizadas a Dm menor a 1 metro.	67
Gráfico 6-3:	Tiempo promedio del procesamiento de detección del pictograma.	76

ÍNDICE DE ANEXOS

- Anexo A:** Datos técnicos de Raspberry PI 4B.
- Anexo B:** Datos técnicos Cámara pi NoIR Camera V2.
- Anexo C:** Esquema de conexión GPS Ublox NEO-6M.
- Anexo D:** Medidas de la estructura del prototipo.
- Anexo E:** Esquema de conexión del circuito general del prototipo.
- Anexo F:** Software del prototipo.

ÍNDICE DE ABREVIATURAS

CAD.	Diseño asistido por computadora.
FLANN.	Fast Library for Approximate Nearest Neighbors.
GPIO.	Entrada/Salida de Propósito General.
GPS.	Sistema de Posicionamiento Global.
OpenCV.	Open Source Computer (Computadora de Código Abierto).
PDI.	Procesamiento Digital de Imágenes.
SIFT.	Scale-invariant feature transform.
VA.	Visión artificial.

RESUMEN

En el presente trabajo de titulación se describe el desarrollo de un dispositivo portátil que permite detectar los diferentes pictogramas de la señalética turística del centro histórico de la ciudad de Riobamba empleando la visión artificial, y a su vez aplicando el algoritmo SIFT y FLANN el cual gracias a la detección de los puntos característicos de cada pictograma o imagen de la señalética puede detectar lugares turísticos, además posee un etapa de guiado mediante la generación de rutas de cada lugar turístico dependiendo del pictograma detectado, esta información es transmitida mediante un mapa en una pantalla instalada en el prototipo para una mejor interacción y la eficiencia de transmisión de datos al usuario final, conjuntamente se elaboró una aplicación amigable y fácil de usar para el usuario donde él puede tomar la imagen, elegir el lugar y guiarse mediante este dispositivo. Los lugares turísticos del centro histórico de la ciudad de Riobamba pueden ser de iglesias, parques, plazas, mercados etc. por ello además de la etapa de guiado posee una pantalla de información del lugar turístico con información básica e importante del lugar. Cabe recalcar que el sistema de visión posee dos partes para su funcionamiento la etapa de detección de imagen que abarca el algoritmo SIFT- FLANN antes mencionado y un algoritmo detección de caracteres el cual funciona para identificar las señaléticas escritas o que contengan palabras. El prototipo está destinado al campo turístico y experimental educativo, por ello la realización del prototipo se manejó con los métodos deductivo y experimental tanto en la etapa de recopilación de información para el desarrollo del dispositivo como la fase de selección, diseño y construcción del mismo. Por último se debe destacar el uso de hardware y software libre orientado al desarrollo de trabajos de investigación .

Palabras clave: <PROTOTIPO>, <PICTOGRAMAS>, <DISPOSITIVO PORTÁTIL>, <VISIÓN ARTIFICIAL>, <DETECCIÓN DE IMAGEN>, <DETECCIÓN DE CARACTERES>.



11-05-2021

1133-DBRA-UTP-2021

SUMMARY

The present graduation work describes the development of a portable device that allows to detect the different pictograms of the tourist signs of the historic center of the city of Riobamba using artificial vision, and in turn applying the SIFT and FLANN algorithm which, thanks to the detection of the characteristic points of each pictogram or image of the signs, can detect tourist places, it also has a guiding stage by means of generating routes of each tourist place depending on the pictogram detected, this information is transmitted through a map on a screen installed in the prototype for a better interaction and the efficiency of data transmission to the final user. Altogether, a user-friendly application was elaborated where tourists can take the image, choose the place and be guided by this device. The tourist places of the historical center of the city of Riobamba can be churches, parks, squares, markets etc. therefore, in addition to the guiding stage, it has an information screen of the tourist place with basic and important information of the place. It should be noted that the vision system has two parts for its operation, the image detection stage covered by the SIFT-FLANN algorithm previously mentioned, and a character detection algorithm which works to identify the written signs or those containing words. The prototype is intended for the tourist field as well as the experimental educational area, for that reason the development of the prototype applied the deductive and experimental methods both in the information gathering stage for the development of the device, and in the selection, design and construction stage. Finally, it must be remarked the use of hardware and free software oriented to the development of research works.

Keywords: <PROTOTYPE>, <PICTOGRAMS>, <PORTABLE DEVICE>,<ARTIFICIAL VISION>, <IMAGE DETECTION>, <CHARACTER DETECTION>.

INTRODUCCIÓN

La visión artificial o visión por computador es uno de los muchos campos de la inteligencia artificial, la cual intenta emular la capacidad de algunos seres vivos para ver una escena (imagen), entenderla y actuar en consecuencia. De otra manera puede entenderse como la programación de un computador para que “comprenda” características en una imagen, típicamente el reconocimiento de objetos, tipos de objetos, o modelos de “estructura espacial”. La mayor dificultad depende esencialmente del grado de control que se posee sobre las “condiciones” del tema a tratar. Una aplicación de visión artificial puede simplificarse tanto como se desee, restringiendo el tipo de escenas a procesar; por tanto, uno de los objetivos de la visión artificial es obtener sistemas cada vez más abiertos (García y Caranqui, 2015).

Existe un crecimiento en el número y tipo de aplicaciones industriales que demandan el uso de técnicas de visión artificial. El continuo desarrollo de nuevos algoritmos y aplicaciones hacen de esta disciplina una tecnología en constante evolución. Ha experimentado un rápido avance en las últimas décadas, así lo demuestran las numerosas investigaciones y publicaciones existentes en la comunidad científica. Esto podría deberse a la gran cantidad de contenido visual como imágenes y video que se genera en la actualidad, así como la capacidad de procesamiento y almacenamiento que tienen los dispositivos electrónicos y la disponibilidad de herramientas, librerías y lenguajes de programación para dicho fin (Alegre et al., 2016).

Existen diferentes áreas donde se pueden aplicar exitosamente las técnicas de visión artificial, como por ejemplo en la biología, medicina, seguridad, industria, robótica, geología, meteorología, cartografía, agricultura, turismo etc. (Bravo y Villca, 2017). El turismo posee una gran cabida para aplicarse gracias a la información que se puede obtener mediante señalética turística las cuales se las puede procesar mediante visión artificial, logrando obtener información relacionada a un sitio o atractivo turístico.

Sin importar el sector o la actividad, en todas ellas la visión artificial logra grandes beneficios y mejoras claves para el posicionamiento de una industria, como el aumento de la productividad y la reducción de costes. Las múltiples aplicaciones de la visión artificial se posicionan como una tecnología entendible, madura y asequible. La integración de dichos sistemas se fundamenta en una gran oferta de componentes opto-electrónicos e informáticos que permiten configurar un equipamiento con un claro retorno de la inversión, en tiempo y en costes (Romero et al., 2015).

El crecimiento turístico de un lugar en particular puede estar limitado por falta de recorridos o rutas hacia los diferentes circuitos turísticos que ofrezcan dicho lugar, teniendo como consecuencia un escaso desarrollo o explotación turística en dicha zona, debido a esto se puede tener un bajo beneficio de estos lugares en términos económicos al no ser explotados correctamente, esto se puede disminuir con la inserción de un aplicativo que por sus características de brindar rutas de circuitos turísticos de una manera sencilla, rápida y con una tecnología que se encuentra en constante evolución se logrará destacar y explotar al máximo un entorno por sobre los demás, elevará su alcance, y como consecuencia, mejorará el desarrollo económico de un lugar, ciudad, país etc. (Andrade, 2014).

Para lo cual se desarrolla un prototipo que brinda información de cada uno de los elementos que conforman un circuito turístico además de proporcionar las rutas hacia cada uno de estos destinos, al tener funciones específicas y ser altamente sencillo de utilizar ofrecerá una muy buena experiencia a los usuarios. Las características de este dispositivo están hechas para el Centro Histórico de Riobamba, no obstante esto no quiere decir que esta tecnología aplicada al turismo no se la pueda replicar en distintas partes del país o distintos sitios turísticos, el futuro inmediato en el turismo y la experiencia que obtienen las personas en el futuro llegará a ser muy diferente al que se conoce en el presente, pues estas aplicaciones que poseen visión artificial logran un gran desarrollo en diferentes áreas al irse incorporando al paso del tiempo teniendo una mayor cobertura o alcance en el desarrollo de la sociedad.

JUSTIFICACIÓN TEÓRICA

El turismo es uno de los sectores económicos con mayor dinamismo en nuestro país y en el mundo. Statista GmbH menciona que en América Latina el 2019, México lideró la lista de países con el mayor número de turistas internacionales, registrando aproximadamente 45 millones de personas (Statista, 2019). Según el ministerio de turismo del gobierno del Ecuador en el boletín mensual de diciembre del 2019 menciona que en dicho año al país llegaron 1534708 turistas sin contar con personas de nacionalidad venezolana. Mientras en Riobamba recibió 138.068 turistas nacionales y 31.413 turistas extranjeros durante el año 2019 según el boletín de demanda turística del GAD Municipal del Cantón Riobamba (Riobamba, 2020).

La preparación de entornos culturales para una actividad turística segura, requiere infraestructura y facilidades apropiadas (Andrade, 2014). En tal sentido, para contribuir al esfuerzo de los gestores turísticos se presenta un prototipo el cual brinda información de la señalética turística de un área en específica mediante visión artificial, el cual permite a las personas decidir si una área en específico es la indicada o preferida para ellos o continuar buscando otra zona turística más acorde a sus gustos o necesidades.

El tiempo de visita a los atractivos turísticos es extenso, muchos de los turistas no siguen el concepto de circuito turístico (Andrade, 2014), lo que provoca que el turista pase varias veces por el mismo lugar, y no encuentre que otro sitio visitar. Enfocándose en el prototipo propuesto, este brinda información de cada sitio turístico mediante las señaléticas de cada área en la que se encuentre, logrando de esta manera aprovechar su tiempo conociendo más zonas turísticas.

La visión artificial ha venido trabajando en muchas disciplinas como biomedicina, reconocimiento de rostros, reconocimiento de huellas dactilares, reconocimiento de imágenes, etc. (Rocha y Escorcía, 2010). La visión artificial engloba a cualquier proceso óptico mediante el cual un sistema inteligente es capaz de extraer información de un entorno para su interpretación mediante el uso de la computadora. El cerebro humano, en combinación con el sistema visual del que dispone nuestro cuerpo, permite un reconocimiento y una interpretación de su entorno mucho más flexible y adaptable a cambios que cualquier sistema de visión artificial. Sin embargo, los sistemas de visión artificial son capaces de procesar cantidades de datos mucho mayores y en menos tiempo si se trata de tareas repetitivas.

En las últimas décadas, la inteligencia artificial se ha introducido en los sistemas o productos de seguridad y sistemas de monitorización y ayuda a las personas. Unos de los sistemas más innovadores para la mejora de la vida cotidiana de las personas con alguna discapacidad son las

aplicaciones que permiten a los niños o personas con autismo a aprender cómo controlar su entorno o su vida diaria, aplicaciones para tener monitorizados los pacientes mayores con su médico o que les recuerde que es el tiempo de tomarse las pastillas, aplicaciones para reconocer los billetes de euro, dólares, u otras monedas del mundo, sistemas para ayudar a las personas ciegas a percibir el entorno mediante sonidos acústicos u otros, incluso detectar o reconocer caras etc. Con respecto a los sistemas de detección y reconocimiento de la señalética se han desarrollado muchas aplicaciones para los coches (Dunai et al., 2017).

JUSTIFICACIÓN APLICATIVA

Esta investigación se realiza debido a la existente necesidad de las personas de obtener información de un lugar turístico mediante el cual puedan economizar tiempo, puedan decidir si ésta es la zona turística que buscan, si es el entorno propicio para satisfacer sus necesidades de distraerse y disfrutar de su tiempo libre, logrando de esta manera optimizar su tiempo. Ya que los turistas comúnmente suelen perderse, si no van acompañados de un guía o una persona que conozca los lugares que visitan. Además de que la provincia de Chimborazo especialmente la Ciudad de Riobamba cuenta con varios atractivos turísticos, entre ellos existe una gran concentración en el Centro Histórico, mismo que no es explotado de gran manera, desperdiciándose su potencial sin que puedan aportar al desarrollo económico de la ciudad (Riobamba, 2020).

El proceso de detección y reconocimiento se basa en los algoritmos de procesamiento de imagen. El hardware del sistema de detección y reconocimiento de señalética se puede realizar con tarjetas de control que con sus características de tamaño reducido permiten que el sistema sea portable y ligero.

La ciudad de Riobamba posee un Centro Histórico el cual puede ser ampliamente explotado turísticamente, este se lograría al tener un dispositivo el cual permita recorrer y conocer la información de los diferentes sitios y circuitos turísticos que ofrece, los cuales se los ofrecerá mediante la programación del dispositivo, lo que con lleva a tener una mejor exploración de los de los sitios turísticos que posee el Centro Histórico de Riobamba, para realizar estos circuitos turísticos se presentan rutas las cuales se ofrecerán al usuario para tener una guía.

La adquisición de la información del entorno se lo realiza mediante una cámara utilizando diferentes técnicas de visión artificial. El algoritmo de visión artificial para este sistema es

entrenado tanto para el reconocimiento de imágenes como para la identificación de palabras que se encuentran en la señalética, esto debido a que en el trabajo de campo realizado para la identificación de la señalética, existen diversos lugares en una sola señalética (letrero), la cual consta de imágenes o palabras y para abarcar todos los diferentes lugares se necesita un algoritmo de visión artificial el cual pueda desenvolverse en ambas formas de adquirir la información necesaria. Una vez obtenida la información de a qué imagen o palabra corresponde cada sitio, el prototipo brinda información del sitio como historia o reseña del sitio turístico, todo este sistema en conjunto conformar el prototipo de información turística, logrando de esta manera que el usuario tenga una mejor experiencia turística y a su vez la nueva utilización de las diferentes tecnologías que están en un desarrollo continuo y que están creciendo exponencialmente tales como la visión artificial, la cual se encuentra en constante desarrollo que ofrece numerosas aplicaciones tanto en la industria como en la vida cotidiana.

OBJETIVO GENERAL

Implementar un prototipo de dispositivo electrónico que mediante el procesamiento de imágenes con visión artificial muestre la información de circuitos turísticos del centro histórico de la ciudad de Riobamba.

OBJETIVOS ESPECÍFICOS

- Definir las características de un sistema de información de circuitos turísticos que se pueda aplicar en el centro histórico de la ciudad de Riobamba con visión artificial.
- Establecer los requerimientos tanto de hardware como de software para el correcto funcionamiento del prototipo del dispositivo electrónico propuesto.
- Seleccionar la herramienta o técnica de visión artificial que servirá para el procesamiento de las imágenes en el prototipo del dispositivo electrónico para la información de circuitos turísticos del centro histórico de la ciudad de Riobamba.
- Validar el recorrido definido por el prototipo comparándolo con el recorrido establecido por aplicaciones informáticas disponibles en la web.

CAPÍTULO I

1 MARCO TEÓRICO

En el presente capítulo se realiza una introducción y una revisión bibliográfica de los campos más importantes y tomados en cuenta para llevar a cabo el trabajo de titulación. Los conceptos que se investigan son de visión artificial, letreros de información turística, lugares considerados turísticos, también se hace una revisión sobre procesamiento digital de imágenes, donde se describe cada una de sus etapas. Por último, se describe el tipo de *software* y *hardware* con el que se va a trabajar en esta investigación.

1.1 Circuitos turísticos

La actividad turística a nivel mundial ha demostrado ser uno de los motores fundamentales que impulsa el desarrollo económico del mundo. En la actualidad 1.2 billones de personas viajan fuera de su país de origen (Méndez, 2019).

El turismo es una actividad económica que beneficia a una ciudad, ya que la realizan personas que se desplazan a un lugar diferente del que son oriundos y por una motivación que se les da. Se dirigen al sector o a la actividad que desean cumplirla. Para conocer estos sectores se basan de circuitos turísticos los cuales se tratan de un conjunto de varios destinos que están enlazados entre sí de manera programada y articulada. Se pueden considerar recursos culturales, naturales y humanos, que son llamativos para los turistas que los visitan (Aigaje y López, 2013).

Según Aigaje I. y López definen a un circuito turístico como *“la suma de varios corredores que unen los servicios y productos de un destino turístico con otro(s) destino(s); para el beneficio de las zonas turísticas y para quienes disfrutan del turismo”* (Aigaje y López, 2013).

La Provincia de Chimborazo especialmente la ciudad de Riobamba cuenta con varios atractivos turísticos, entre ellos existe una gran concentración en el Centro Histórico, mismo que no es explotado de gran manera, desperdiciándose su potencial sin que puedan aportar al desarrollo económico de la ciudad de manera adecuada (Méndez, 2019). Para lo cual se realiza un prototipo de circuitos turísticos que tenga como características el reconocimiento de señalética mediante visión artificial, rutas hacia los diferentes destinos además que presente información de cada atractivo turístico al llegar a su destino, con lo cual esto con lleva a tener una mejor exploración del Centro

Histórico y es un esfuerzo por impulsar tanto el desarrollo económico como el turístico de la ciudad.

1.1.1 Lugares turísticos o culturales en la ciudad de Riobamba

El espacio de estudio abarca dos zonas. La de primer orden, donde se encuentra el espacio denominado el Centro Histórico y la de segundo orden que se le reconoce como zona de amortiguamiento o de respeto, en la Tabla 1-1 se observa los límites del Centro Histórico (Mejía, 2015).

Tabla 1-1: Límites del Centro Histórico de la ciudad de Riobamba.

ZONA DE PRIMER ORDEN		ZONA DE SEGUNDO ORDEN	
Norte	Calle Juan Lavalle	Norte	Calle Diego de Ibarra
Sur	Calle Pedro Alvarado	Sur	Calle Diego de Almagro
Este	Calle Argentinos	Este	Calle Junín
Oeste	Calle Gaspar de Villarroel	Oeste	Calle Colombia

Fuente: (Mejía, 2015, p. 14).

Realizado por: Berrones A, Crespata C; 2021.

En la Figura 1-1 se observa la zona de primer orden y segundo orden que conforman el Centro Histórico, donde la zona de primer orden corresponde al núcleo inicial de la fundación de la ciudad rodeado por edificaciones arquitectónicas e hitos históricos (Mejía, 2015).

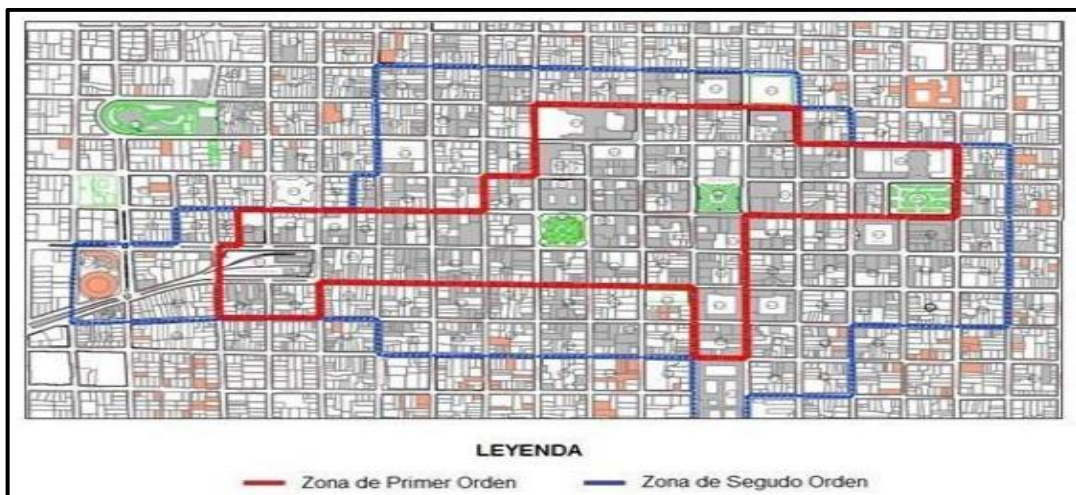


Figura 1-1: Zonificación del Centro Histórico de la ciudad de Riobamba.

Fuente: (Mejía, 2015).

Los atractivos turísticos son los conjuntos de lugares, costumbres y sucesos de un cierto lugar, los cuales poseen características que atraen el interés de un turista. Riobamba por su singular belleza natural y arquitectónica, es un atractivo para turistas nacionales y extranjeros (Loyola, 2018).

En la tabla 2-1 se indica la clasificación de los atractivos turísticos con los que cuenta el Centro Histórico de la ciudad de Riobamba, uno de los más hermosos y jóvenes del Ecuador (Riobamba, 2020).

Tabla 2-1: Atractivos turísticos del Centro Histórico de Riobamba.

Atractivos turísticos	Descripción	Nombres
Iglesias	La belleza de las Iglesias y Museos pertenecientes a Riobamba es incomparable. El encanto del Centro Histórico de diseño italiano junto a su arquitectura del siglo XIX y XX son espacios recomendados para que los visitantes puedan apreciar estos lugares.	Iglesia San Antonio de Padua, Iglesia la Concepción, Iglesia San Alfonso, Iglesia la Catedral, Iglesia la Basílica, Iglesia de San Francisco.
Museos	Conservan preciosos recuerdos, estos elementos están estrechamente relacionados con la historia, cultura, tradición y características de Riobamba.	Museo Arte Religioso de la Concepción, Museo de Ciencias Naturales del Colegio Pedro Vicente Maldonado, Casa Museo de Riobamba, Museo de piedra de Riobamba.
Parques	Poseen jardines y monumentos que conservan importantes hitos que marcan la historia del Ecuador.	Parque Maldonado, Parque Sucre, Parque La Libertad, Parque 21 de Abril.
Mercados y Plazas	Son una de las formas más importantes de relación e interacción entre las sociedades indígenas y mestizas, en un inicio las plazas llevan el nombre de los templos locales y permanecían al aire libre, los comerciantes vendían sus productos en sus propios quioscos de madera y permanecían cubiertos con telas.	Plaza Alfaro, Plaza Roja, Mercado Mariano Borja, Mercado San Alfonso, Mercado San Francisco.
Edificaciones	posee bellas joyas arquitectónicas del siglo XIX y XX.	Plaza de Toros Raúl Dávalos, Teatro León, Colegio Maldonado, Casa de la Independencia, Edificio del SRI, Edificio del Correo, Palacio Municipal, Edificio de la Gobernación, Estación del Ferrocarril.

Fuente: (Riobamba, 2020).

Realizado por: Berrones A, Crespata C; 2021.

De la clasificación anterior existe un recorrido muy conocido en Riobamba la cual es denominada la ruta de las iglesias, en la figura 2-1 se puede apreciar la ruta que conforma este circuito. De la misma forma se puede crear circuitos turísticos que conformen las plazas, mercados, museos, edificaciones y parques.

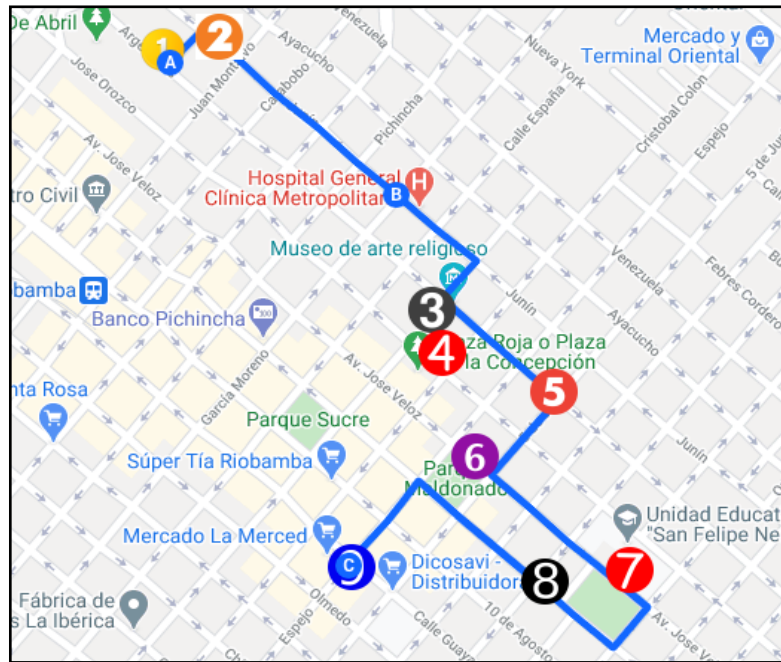


Figura 2-1: Ruta de las iglesias.
Fuente:(Riobamba, 2020).

1.1.2 Señalética turística

Tiene por objetivo identificar, reconocer y ayudar con el acceso a distintos servicios que necesitan los usuarios en un determinado lugar, la señalética debe crearse e introducirse de acuerdo a su campo. Este tipo de señalética como se indica en la figura 3-1 se basa en referirse a las señales, marcas o carteles necesarios para direccionar al usuario en el lapso que realiza un recorrido por un circuito turístico, pudiendo así brindar una breve descripción de información referente a los elementos atractivos que se encuentran en el recorrido (Andrade, 2014).



Figura 3-1: Señalética de información turística.
Fuente: (Zanutto et al., 2011).

Según el plan nacional del Buen Vivir 2017-2021 menciona que “*El espacio público y las áreas verdes serán los elementos estructurales de la vida comunitaria en la ciudad, consolidándose como lugares seguros de interrelación y de desarrollo de actividades culturales, de ocio y esparcimiento*”(SENPLADES, 2017).

1.1.2.1 Pictogramas

Son signos que representan esquemáticamente un símbolo, objeto real, figura o servicio. Se entiende como un signo claro que sintetiza un mensaje o información con el objetivo de informar y/o señalar. Se clasifican en los siguientes (Andrade, 2014):

- **Pictogramas de atractivos naturales.** Representan la riqueza biodiversa de un lugar, una región y un país.
- **Pictogramas de atractivos turísticos culturales.** Son símbolos representativos de nuestra cultura que identifican a este tipo de bienes.
- **Pictogramas de actividades turísticas.** Representan acciones de interés turístico y/o recreativas.
- **Pictogramas de apoyo a los servicios turísticos.** Son símbolos de apoyo a los atractivos turísticos que permiten orientar al visitante al momento de acceder al uso de los servicios turísticos.

Los pictogramas de atractivos turísticos culturales representa el conjunto de sitios y manifestaciones que se consideran de valor o aporte de una comunidad determinada y que permite al visitante conocer parte de los sucesos ocurridos en una región o país, reflejadas en obras de arquitectura, zonas históricas, sitios arqueológicos, iglesias, conventos, colecciones particulares, grupos étnicos, manifestaciones religiosas, artesanía, ferias y mercados, shamanismo, explotaciones mineras, centros científicos y técnicos, etc (Zanutto et al., 2011). En la figura 4-1 se aprecia pictogramas de atractivos culturales.



Figura 4-1: Pictogramas de atractivos culturales.

Fuente: (Zanutto et al., 2011).

1.2 Visión artificial

Es una herramienta que abarca procesos y características las cuales incorporan ojos a una máquina. Es una tarea complicada ya que interactúan varios elementos, accesorios a su procesador, estos pueden ser sensores, cámaras o tarjetas de señales (Saquinga y Andrade, 2019).

Si bien se sabe que la visión artificial en sí tiene muchas definiciones, por lo que se toma en cuenta a Gonzales Ana en 2016 donde manifiesta que *“La visión artificial o comprensión de imágenes describe la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, bien a partir de una o varias imágenes bidimensionales de ese mundo”* (Gonzales et al., 2016). El sistema de visión artificial se constituye de la siguiente manera como se observa en la figura 5-1.

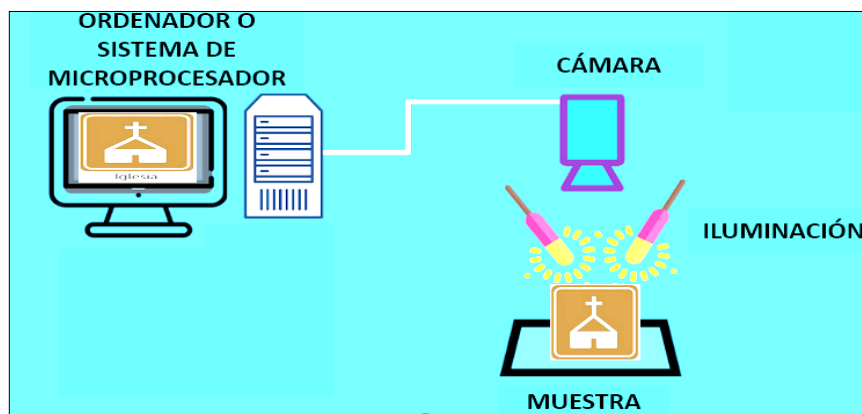


Figura 5-1: Elementos que constituyen un sistema de visión artificial.
Fuente: (Aula21, 2020).

De lo que constituye el sistema se puede reconocer que consta de 3 partes que son:

La cámara que captura la imagen que tiene como finalidad imitar la función del ojo humano, es decir capturar imágenes proyectadas. Mediante las ópticas se transfiere a un sistema electrónico para poder interpretarla, almacenarla y visualizarla. Las cámaras de visión artificial se asemejan a las cámaras digitales, con algunas diferencias elevadas, como en su tratamiento de la imagen o su control del disparo (Albert y Antiga, 2018).

La iluminación que permite reflejar la escena, es un factor muy importante porque las cámaras captan la luz que se refleja en los objetos cuando se realiza la adquisición de la imagen para posteriormente realizar su procesamiento. El objetivo de la iluminación es el de poder ayudar a la cámara que pueda hacer el reconocimiento adecuado, si se utiliza una iluminación correcta en la aplicación, esta podrá arrojar buenos resultados. Se sabe que existen distintos tipos como lo son:

iluminación artificial, por luz lateral, frontal, campo oscuro, por contraste e iluminación natural (Albert y Antiga, 2018).

La iluminación natural o luz ambiente, puede variar a diferentes horas o estados del día, pero para el desarrollo del prototipo es la que va a intervenir, debido a que la adquisición de las imágenes no se encuentran en un lugar fijo y por cuestiones de recursos no es factible realizar una iluminación a todas las señálicas del Centro Histórico.

Este tipo de iluminación es emitida por el sol como se aprecia en la figura 6-1, al ser nuestra principal fuente de iluminación y energía de forma natural, esta característica tiene la desventaja de que no se puede controlar, es impredecible y depende de muchos factores como es el clima, las diferentes estaciones del año, el tiempo donde pueden variar a diferentes horas del día, es por ello que para las pruebas de validación del prototipo se realizarán mediciones de la iluminación del entorno para conocer el comportamiento del prototipo a las diferentes variaciones que se presenten a lo largo de la captura (Ilumina, 2018).



Figura 6-1: Fuente de iluminación natural (Sol).
Fuente: (Ilumina, 2018).

Por último, forma parte un ordenador o sistema de microprocesador, donde se lleva a cabo el almacenamiento y procesamiento de las imágenes adquiridas. Este elemento realiza el procesamiento mediante un circuito lógico que responde a operaciones lógicas y aritméticas permitiéndoles funcionar a las computadoras, celulares, dispositivos etc (Saquina y Andrade, 2019).

1.3 Procesamiento Digital de Imágenes (PDI)

Se basa en procesos algorítmicos que convierten una imagen en otra de tal manera que resalta la información de interés y/o elimina o atenúa información de la imagen que no es relevante para la aplicación q está destinada. De esta manera las tareas del procesamiento digital de imágenes consiste en la eliminación de ruido, mejoramiento de contraste, eliminación de distorsiones por

efectos ópticos o movimiento, mapeos geométricos además de transformaciones de color etc (Guerrero, 2018).

1.3.1 La imagen digital

Es una representación bidimensional del mundo tridimensional a través de una escena. La imagen es el resultado obtenido de la adquisición de una señal la cual se proporciona por un sensor, el cual es el encargado de convertir la información del espectro electromagnético en codificaciones numéricas. Y su transformación en el formato de representación de imagen seleccionado, que forma información discreta en los valores que puede tomar y los parámetros que la definen. En general, una imagen digital se define como una matriz (o vector) de tamaño $N \times M$. Cada elemento de la matriz contiene un valor discreto, que cuantifica el nivel de información del elemento correspondiente y está representado por un número finito de bits (q) (Alegre et al., 2016). En la figura 7-1 se indica la manera de representar las coordenadas en una imagen digital.

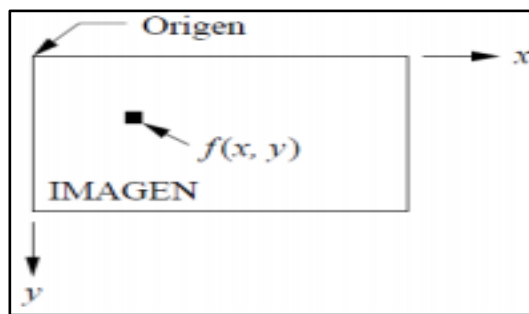


Figura 7-1: Convención de coordenadas.
Fuente: (Mundaca, 2016).

1.3.2 Etapas de procesamiento digital de imágenes

La figura 8-1 se muestra las etapas que poseen el procesamiento digital de imágenes.

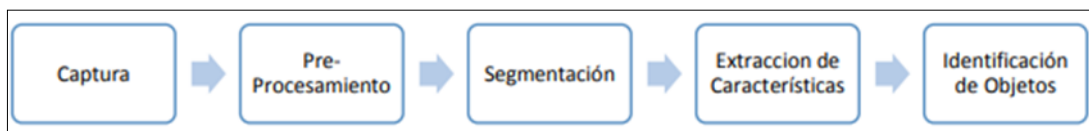


Figura 8-1: Etapas del procesamiento digital de imágenes.
Fuente: (Mundaca, 2016).

1.3.2.1 Captura

Es el proceso de adquisición de imágenes digitales a través de dispositivos como cámaras digitales, videocámaras, escáneres, telescopios, satélites, etc (García y Caranqui, 2015). Representa la materia prima para el procesamiento de imágenes para un reconocimiento exitoso o fallido. Si la

imagen no es adecuada, no importa cuán robusto sea el algoritmo, no se puede obtener el resultado esperado (Romero et al., 2015).

Para la adquisición de imágenes se utilizan dispositivos de captura que cuente con buena resolución donde este adquirirá la escena y después enví a una unidad donde pueda ser procesada (Alegre et al., 2016).

Para poder seleccionar la cámara y la óptica que compongan el sistema de visión que mejor se adecue a la aplicación que se desea resolver, es necesario tener en cuenta los siguientes parámetros:

1. Campo de visión (Field Of View, FOV): área del objeto o escena del que se desea capturar una imagen.
2. Resolución: tamaño de la característica más pequeña del objeto que se desea que se vea en la imagen.
3. Distancia de trabajo: separación que existe entre el objeto que se captura la imagen y la óptica.
4. Profundidad de campo: máxima profundidad del objeto necesaria para conseguir un enfoque adecuado.

1.3.2.2 Pre-Procesamiento

Se realiza técnicas tales como reducción de ruido, mejora de contraste, nitidez de imagen, mejora de ciertos detalles o de características de la imagen, restauración de imagen (García y Caranqui, 2015). El propósito es establecer las condiciones necesarias para que el funcionamiento del proceso sea óptimo y eficiente es decir, preparar la imagen para que cuando se aplique el algoritmo, pueda funcionar normalmente en cada etapa (Mundaca, 2016).

Es utilizado para eliminar ruido en donde se incluyen operaciones de mejora en la fotografía adquirida como es atenuar las imperfecciones debido a la función de transferencia del sistema de captación de imágenes, de regularizar la imagen, de mejorar el contraste o de optimizar la distribución de la intensidad o de realzar algunas características de la imagen, como bordes o áreas (Arevalo, 2019).

Cuando se captura la imagen esta es adquirido normalmente con ruido el cual se trata de información no deseada en una imagen que causa una contaminación, estos pueden ser:

- **Ruido impulsivo:** El valor de cada píxel no tiene relación con el valor ideal, sino con el valor del ruido, estos valores pueden ser muy altos o bajos (puntos blancos y/o negros) donde se distribuyen de manera aleatoria por la imagen. Este aparece producto de interferencias atmosféricas, o por acciones hechas por el humano (motores de coches, etc.)(Arévalo, 2019).

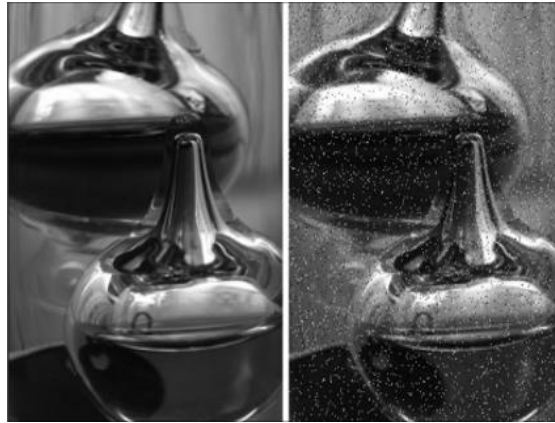


Figura 9-1: Imagen original/ con ruido impulsivo.
Fuente:(Arévalo, 2019).

- **Ruido “Gaussiano”:** Se puede identificarlo como ligeras variaciones en una imagen. El pixel toma el valor deseado pero se le agrega una cantidad de error, conocido como “variable aleatoria gaussiana”. Este error normalmente es generado por perturbaciones al momento de transmitir las señales eléctricas o errores de ganancias en la cámara (Villagómez, 2020).



Figura 10-1: Ruido “Gaussiano”.
Fuente:(Villagómez, 2020)

Como se revisa, el ruido se produce por diferentes circunstancias, y afectan al posterior tratamiento de la imagen, por este motivo se necesita realizar un filtrado de imagen, el cual consiste en la modificación de una imagen para mejorarla eliminando ruido o elementos no deseados (Maduell, 2016). En la tabla 3-1 se aprecia las características de los filtros más comunes para eliminar ruido.

Tabla 3-1:Tipos de filtros.

Filtro	Descripción	Características
Media	Consiste en revisar pixel por pixel y se calcula el valor de la media de los pixeles vecinos, el resultado obtenido será el nuevo valor del pixel analizado.	<ul style="list-style-type: none"> • Suavizar imágenes, es decir, reducir la cantidad de variaciones de intensidad entre píxeles vecinos. • Conseguir que las intensidades de los objetos pequeños se mezclen con el fondo con el fin de detectar los objetos de mayor tamaño. • Tiene como desventaja que es sensible a cambios locales, puede crear nuevas intensidades de grises que no aparecían en la imagen.
Mediana	Consiste en calcular la mediana utilizando el conjunto de pixeles vecinos al pixel que se desea analizar. Por lo tanto, si hay un pixel muy distinto al del entorno, este tomará el valor de la mediana con respecto a sus pixeles vecinos.	<ul style="list-style-type: none"> • Atenúa el ruido impulsional (Sal y pimienta). • Elimina efectos engañosos. • Preserva bordes de la imagen. • Es el más simple, intuitivo y fácil de implementar.
Gaussiano	Se utiliza cuando se presenta ruido gaussiano. Tiene la capacidad de suavizar la imagen, eliminando el ruido.	<ul style="list-style-type: none"> • Disminución de la nitidez. • Aumento de borrosidad. • Pérdida de detalles.

Fuente: (Maduell, 2016).

Realizado por: Berrones A, Crespata C; 2021.

Con respecto a la Tabla 3-1 por las características de implementación en cuanto a sencillez y la característica de preservar los bordes de la imagen, puesto que estos son importantes para la detección de una u otra señalética, el filtro mediana para la eliminación de ruido es el más apropiado. En la figura 11-1 se observa un ejemplo de cómo actúa este filtro al ordenar los valores en la vecindad de cada píxel de menor a mayor y seleccionar el valor en la posición intermedia (mediana).

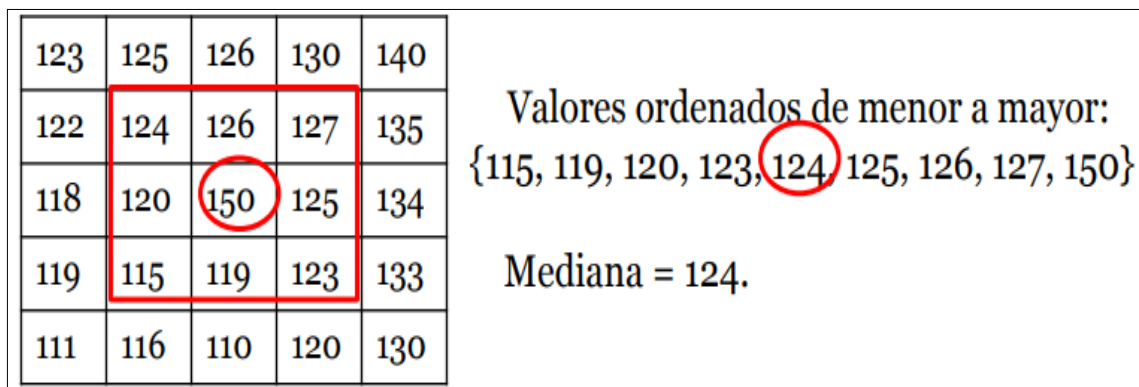


Figura 11-1: Operación filtro mediana.

Fuente:(Maduell, 2016).

1.3.2.3 Segmentación

Es el proceso en el cual se divide una imagen en objetos o regiones los cuales sean del interés para el estudio. (García y Caranqui, 2015). Divide la imagen en segmentos de píxeles que son semejantes en relación con ciertos criterios (Mundaca, 2016). La segmentación subdivide una imagen en sus partes u objetos constituyentes para separar la parte de interés del resto de la imagen, por lo que el nivel de subdivisión que se lleve a cabo depende del problema destinado a resolver (Palomino y Concha, 2014).

Los algoritmos de segmentación en su mayoría se basan en dos propiedades básicas: la similitud y la discontinuidad. La características de segmentación por discontinuidad es la de realizar divisiones de la imagen por cambios abruptos en intensidad, como lo es en el caso de la detección de bordes. En relación a segmentación basada en la similitud, lo realiza mediante particiones de una imagen en regiones que son similares con respecto a un conjunto de criterios predefinidos (Alegre et al., 2016).

El proceso de segmentación se encarga de evaluar cada píxel de la imagen y decidir si contiene o no las características de interés. Como resultado, este método genera una imagen binaria, donde los píxeles que pertenecen al objeto se representan con un '1' (objeto en primer plano), mientras que los que no pertenecen al mismo se representan con un '0' (fondo). La decisión de pertenencia de un píxel a uno u otro segmento se basa en el análisis de alguna característica de la imagen, como por ejemplo los niveles de intensidad o la textura (Alegre et al., 2016).

La efectividad de un método de segmentación varía en relación a la aplicación a utilizar, debido a esto y a la gran cantidad de imágenes es difícil evaluar la aplicación de un método en general. Fundamentalmente, lo que se busca es que diferentes objetos tengan valores claramente diferentes de las características discriminantes, y el éxito de la operación se comprueba experimentalmente en cada caso (Alegre et al., 2016). En la Tabla 4-1 se enumera los tipos de segmentación y características para cada uno de ellos.

Tabla 4-1: Tipos de segmentación.

Clasificación	Tipos	Características
Segmentación basada en características de píxel	<ul style="list-style-type: none"> • Segmentación por niveles de gris • Segmentación de imágenes en color • Segmentación por texturas 	<ul style="list-style-type: none"> • Se usa comúnmente cuando se requiere separar objetos con similares características de color o intensidad de un fondo heterogéneo. • Resulta útil para la detección de objetos que tienen colores y formas muy definidas, como las señales de tráfico. • Sensible al ruido. • El tiempo de computación y memoria usados es alto. • Facilita el análisis de imágenes.
Segmentación basada en transiciones	<ul style="list-style-type: none"> • Detección de bordes 	<ul style="list-style-type: none"> • Proporciona información útil acerca de los límites del objeto. • Simplifica el análisis de imágenes, realizando una reducción drástica de la cantidad de datos a ser procesados. • Son muy efectivos para acentuar el contraste y detectar puntos aislados o pequeños detalles. • Los bordes de una escena suelen ser invariantes a los cambios de luz. • Costo computacional bajo. • Es fuerte frente al ruido, asegura una robusta detección.
Segmentación basada en modelos	<ul style="list-style-type: none"> • Transformada de Hough 	<ul style="list-style-type: none"> • Fácil implementación en paralelo. • Útil para la detección de formas complejas. • Reconoce patrones ligeramente deformados, ocultos. • Complejidad en el análisis de las imágenes. • Robusta frente al ruido. • Consume muchos recursos. • No ofrece respuesta absoluta, sino un índice de probabilidad de que cada una de las formas posibles sea la buscada.

Fuente: (Maduell, 2016), (Palomino y Concha, 2014), (Gonzales et al., 2016).

Realizado por: Berrones A, Crespata C; 2021.

De acuerdo a la Tabla 4-1 y la aplicación que se va a desarrollar la segmentación basada en transiciones, sería la mejor opción para ser utilizada ya que sus características de bajo costo computacional, su robustez en presencia de ruido y la simplicidad en el análisis de imágenes son necesarios para obtener un tiempo corto de respuesta del prototipo.

1.3.2.4 Extracción de características

En esta apartado se realiza el estudio de los procesos, donde se indican los algoritmos a implementar, para la extracción de puntos característicos de las imágenes, el cual se basa en obtener características con alguna información cuantitativa de interés de imágenes para diferenciar un tipo de objeto de otro (Palomino y Concha, 2014).

Es una de las partes indispensables en muchas aplicaciones de visión artificial y ha atraído cada vez más atención en los últimos años. A lo largo de todo este tiempo, se han presentado diversos tipos de detectores y descriptores de características, según los parámetros elegidos una vez creada la elección de las características (Narro, 2019). A continuación en la tabla 5-1 se hace un estudio de los tres algoritmos descriptores más representativos:

Tabla 5-1: Algoritmos para la extracción de características.

Tipos	Ventajas	Desventajas
SIFT (SCALE-INVARIANT FEATURE TRANSFORM)	<ul style="list-style-type: none"> • Mayor número de características detectadas (extracción de puntos). • Puede calcular varios descriptores (con distinta orientación) para una misma posición de la imagen. • Detecta puntos más robustos. • Trabaja en escala de grises. • Puede detectar objetos con oclusión parcial. 	<ul style="list-style-type: none"> • Al detectar una mayor cantidad de puntos, le toma más tiempo que SURF realizar el trabajo.
SURF (SPEED UP ROBUST FEATURE)	<ul style="list-style-type: none"> • Detecta puntos con mayor velocidad. • Trabaja en escala de grises. • Fiable ante transformaciones de la imagen. • Es una mejora de SIFT. 	<ul style="list-style-type: none"> • Calcula un vector descriptor para una determinada posición (x, y). • Detecta un mayor número de falsos positivos. • Detecta menor cantidad de puntos que SIFT.
Redes Neuronales Artificiales (RNA)	<ul style="list-style-type: none"> • Es un método moderno, separa regiones no lineales de decisiones complejas. • Resuelve problemas de clasificación de gran complejidad. 	<ul style="list-style-type: none"> • Elevada cantidad de datos para el entrenamiento. • Tiempo de aprendizaje elevado. • Alto consumo de recursos para el entrenamiento.

Fuente: (Aracil, 2012), (García Barquero et al., 2012), (Guerrero, 2014), (Juárez, 2013), (Romero y Cazorla, 2009).

Realizado por: Berrones A, Crespata C; 2021.

El algoritmo sobre Redes Neuronales Artificiales (RNA) se descarta en primera instancia por presentar un tiempo de aprendizaje elevado y un alto consumo de recursos al realizar el entrenamiento sobre los otros algoritmos descriptores. Es por esto que por poseer características similares se elige entre SIFT y SURF. En la tabla 5-1 se aprecia que el algoritmo SIFT puede detectar un mayor número de puntos característicos mientras SURF detecta en menor cantidad, también SIFT puede calcular varios descriptores (con distinta orientación) para una misma posición de la imagen, mientras SURF calcula solamente un descriptor. En relación a un trabajo de investigación en la Universidad Autónoma de España, se ha realizado un estudio sobre estos

descriptores SIFT y SURF aplicado a imágenes, de ahí se obtiene la tabla 6-1 indicando el rendimiento de cada descriptor.

Tabla 6-1: Comparación de rendimiento entre descriptores SIFT y SURF.

Tipo de escena	SIFT	SURF
Cambio de ángulo	0.9268	0.9
Cambio de iluminación	0.9506	0.9258
Escena de movimiento	0.847	0.8397
Variación de zoom	0.9652	0.9682
Escena global	0.9112	0.9084

Fuente: (Boullosa, 2011).

Realizado por: Berrones A, Crespata C; 2021.

En la tabla 6-1 se aprecian diferentes tipos de escenas donde el rendimiento de SIFT es superior a SURF, excepto en la escena con variación de zoom. Por todo lo mencionado anteriormente y por la superioridad de rendimiento en los tipos de escena como el cambio de ángulo, el cambio de iluminación, escena en movimiento y escena global se propone SIFT para la extracción de características o keypoints de los pictogramas.

Por trabajos de investigación sobre el costo computacional que requiere el algoritmo SIFT en el Instituto Nacional de Astrofísica, Óptica y Electrónica de Puebla realizados sobre una PC con procesador P4 con 2GB de (Chang, 2010), se toma como base escoger una tarjeta controladora que incorpore mínimo 2 GB de RAM.

1.3.2.5 Identificación de la señalética

El reconocimiento es la última etapa del proceso digital de imágenes, es el proceso de asignar etiquetas a los objetos en función de la información proporcionada por sus descriptores. La interpretación implica asignar significado a un grupo de objetos identificados (Palomino y Concha, 2014). Los sistemas de reconocimiento de imágenes deben de ser robustos y capaces de trabajar bajo variaciones de los objetos a reconocer. Variaciones como: punto de vista, iluminación, escalado. Para clasificar una imagen se comparan todas y cada una de sus características locales con todas y cada una de las características locales extraídas del conjunto de aprendizaje, las imágenes se clasifican en la clase que mayor número de coincidencias tenga (Fernandez, 2019).

Para la clasificación de señalética se basara en algoritmos de correspondencia de características, los cuales se encargan de trabajar en la coincidencia de puntos de interés, los descriptores de imágenes se comparan entre sí para identificar características similares. Para dos imágenes, puede obtener un conjunto de pares $(X_i, Y_i) \leftrightarrow (X'_i, Y'_i)$, donde (X_i, Y_i) están las características de la

primera imagen y $(X'i, Y'i)$, su correspondencia en la otra imagen (Narro, 2019). En la tabla 7-1 se realiza un estudio de los algoritmos de correspondencia más utilizados actualmente.

Tabla 7-1: Algoritmos para la comparación de características.

Algoritmos	Descripción	Características
Fuerza bruta	BFMatcher (Brute-Force Matcher por sus siglas en inglés) es un algoritmo simple, se encarga de tomar cada descriptor de la primera imagen y lo compara con las características de la segunda imagen mediante calculo por distancia y eligiendo el punto más cercano.	<ul style="list-style-type: none"> • El tiempo de ejecución crece exponencialmente si el conjunto de datos aumenta. • Se puede implementar por Open CV.
FLANN	Son las siglas de Fast Library for Approximate Nearest Neighbors. Contiene un conjunto de algoritmos que implementan la búsqueda del vecino más cercano para grandes conjuntos de datos y características de alta dimensión, y estos algoritmos están optimizados para una lectura uniforme fija.	<ul style="list-style-type: none"> • Mejor rendimiento y más rápido frente a grandes conjuntos de datos. • Se puede usar fácilmente en la mayoría de los entornos donde se usa el lenguaje C, como MATLAB y Python. • Implementado por OpenCV. • Puede reducir las coincidencias de funciones falsas.

Fuente: (Narro, 2019), (Pacheco, 2011), (OpenCV, 2021), (Vijayan y Kp, 2019).

Realizado por: Berrones A, Crespata C; 2021.

De acuerdo a las características que se presentan en la tabla 7-1. se escoge el algoritmo FLANN por presentar un mejor rendimiento y ser más rápido frente a grandes conjuntos de datos. Por consiguiente se utilizara una combinación de los algoritmo SIFT para detección de características y FLANN para las coincidencias entre un par de imágenes. Cuando se fusiona SIFT y FLANN hace coincidir los descriptores de características de un conjunto con las características en el conjunto de destino. Con respecto a SIFT el descriptor suele compararse y emparejarse utilizando la distancia euclidiana, mientras que FLANN hace coincidir una característica con otra cuando la distancia a esa característica es menor que un cierto valor de umbral de la distancia a la siguiente característica más cercana. Esto puede reducir las coincidencias de funciones falsas (OpenCV, 2021), (Vijayan y Kp, 2019).

1.4 Hardware

Para la captura y el procesamiento de imágenes que se obtienen para el desarrollo de un proyecto de visión artificial, se necesita una tarjeta o placa controladora que realice este trabajo, para que el sistema que se desarrolla, actué de acuerdo a sus especificaciones, se necesita una cámara que realice la adquisición de imágenes, un dispositivo que permita posicionar al usuario y una pantalla donde se visualice la información para el usuario. Es por esto que la selección de hardware es de gran importancia.

1.4.1 Tarjetas controladoras

Son computadoras de bajo costo, de tamaño reducido, que se comparan con la dimensión de una tarjeta de crédito. Permiten la expansión e incorporación de capacidades donde se procesa y se controla señales emitidas por periféricos de entrada y salida (Saquina y Andrade, 2019). De acuerdo a la selección que se realizó del algoritmo se concluyó trabajar con SIFT por las diferentes características que presenta, en estudios relacionados al enfoque, desarrollo y uso de este algoritmo en el Instituto Nacional de Astrofísica, Óptica y Electrónica de Puebla se utiliza como base para el desarrollo una PC con procesador P4 con 2GB de RAM, en consecuencia para la selección de la tarjeta controladora se busca las familias que ofrezcan esta característica, en la Tabla 8-1 se presenta los diferentes tipos de tarjetas de desarrollo que cumplen con esta condición (Chang, 2010).

Tabla 8-1: Características de tarjetas controladoras

Características	Familia FPGA	Familia Raspberry Pi
Fabricantes	Xilinx, Altera.	Raspberry Pi Foundation
Modelos	Virtex, Kintex, Spartan, Artix.	Raspberry Pi 4B, Raspberry Pi 3B+, Raspberry Pi Zero, Raspberry Pi Zero Wh, Raspberry Pi Zero W.
RAM	512MB - 16GB	512MB - 8 GB.
Puertos	100 – 1400 E/S.	26 - 40 GPIO.
Lenguaje de programación	VHDL y Verilog.	Python, C, C++, Ruby.
Alimentación	Consumen mayor cantidad de energía (5v-12v).	Tienen un menor consumo de energía 7mA - 2.5A y una fuente alimentación de 5v.
Facilidad de adquisición	Opciones muy limitadas en el área local, adquisición internacional supone gastos de envío, aumento en el precio y con espera considerable.	Existe en el mercado local.
Sistema operativo	Microsoft Windows, Linux.	Linux, Raspbian Jessie.
Disponibilidad de Software	Intel Quartus Design Software desde \$2995.	Software libre.
Costo (\$)	132.84 – 3434.16.	35- 150.

Fuente: (Daroch y Antíñire, 2018).

Realizado por: Berrones A, Crespata C; 2021

Para la seleccionar de entre las diferentes familias de tarjetas presentadas, se debe realizar basándose principalmente en ciertos criterios de hardware que sirvan para cumplir con la implementación del prototipo, dispositivo o sistema de visión artificial; Además, el costo y la disponibilidad en el mercado, que sea compatible el hardware y software de visión artificial, interfaz de comunicación con periféricos externos, que se compatible con sistemas operativos de tipo código abierto. De acuerdo a la Tabla 8-1 Se escoge la familia Raspberry para el desarrollo del prototipo, puesto que en características como costo, disponibilidad de software, facilidad de adquisición y lenguajes de programación ofrece mejores beneficios en relación a las FPGA.

1.4.2 Cámaras

Es el componente primordial en la captura de imágenes durante la visión artificial. Se compone de una lente y un sensor de imagen. La lente es responsable de enfocar y capturar la escena, y luego transmitir la imagen. Un buen sistema óptico de lentes puede determinar la calidad y resolución de la imagen. En la tabla 9-1 se aprecian las características de 2 diferentes familias de cámaras (Manobanda, 2019).

Tabla 9-1: Comparación entre familias de cámaras.

Características	Familia Raspberry	Familia Sony
Dimensiones	5 × 24 × 9 mm	62.4 x 61.8 x 33.3 mm
Peso (g)	3 -9	70-90
Resolución (megapíxeles)	5-8	12- 18.2
Modos de video	1080p30, 720p60 and 640 × 480p60/90	Full HD
Integración Linux	V4L2 driver available	No dispone
Sensor de Resolución	3280 x 2464 pixeles 2592 x 1944 pixeles	zoom 10x Enfoque estabilizado
Tamaño óptico	3.175-6.35 mm	25-250 mm
Campo de visión horizontal o enfoque	62.2 grados	4.45 - 44.5 mm de enfoque
Campo de visión vertical o enfoque	48.8 grados 54 X 41 grados	
Costo	20-30\$	110-150\$
Ventajas	<ul style="list-style-type: none"> • Precio más económico. • Compatibilidad con Raspberry. 	<ul style="list-style-type: none"> • Tiene zoom. • Tiene mejor resolución.
Desventajas	No posee zoom.	Más pesada y de mayor dimensiones.

Fuente: (Raspberry Pi, 2019), (Tiendatec, 2019).

Realizado por: Berrones A, Crespata C; 2021.

Para escoger el modelo de la cámara se basa en la compatibilidad con la tarjeta Raspberry Pi, que permita obtener datos para ser analizados con el *software* y librerías de visión artificial. En una investigación similar con visión artificial realizada en la Universidad Técnica de Ambato, se trabaja con una cámara que tiene una resolución de 8 megapíxeles, donde se menciona que brinda una mejor calidad de imagen, para llegar a esta conclusión realizaron pruebas de funcionamiento con diferentes modelos de cámaras y diferentes resoluciones, siendo la de 8 megapíxeles la que menos errores cometió y a su vez sus tiempos de respuesta fueron más rápidos (Pico, 2019). Con estos antecedentes se debe elegir entre distintas cámaras que mínimo tengan una resolución de imagen de 8 megapíxeles, sin olvidar que el costo y las dimensiones también influye en la selección de este componente.

1.4.3 Pantalla táctil

Es un elemento que fue desarrollado con el objetivo de tener un medio de visualización, que permite la interacción del usuario a través de uno o varios toques sobre la pantalla, en referencia a estas pantalla se observa un ejemplo de ellas en la figura 12-1 (Manobanda, 2019).

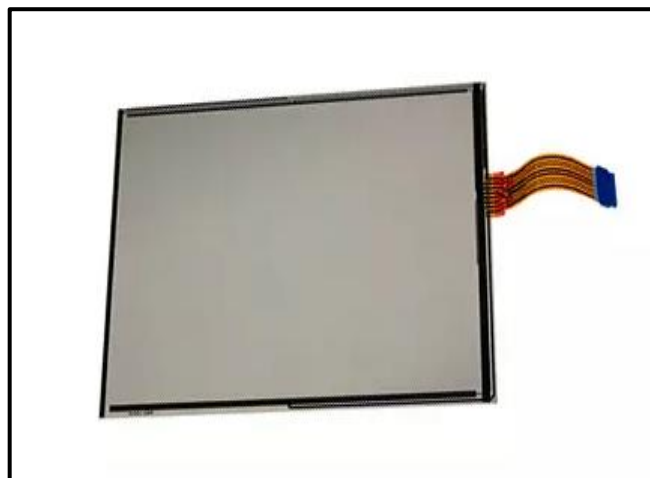


Figura 12-1: Pantallas táctil LCD.

Fuente: (Manobanda, 2019).

En la tabla 10-1 se pueden apreciar 3 tipos de modelos de pantallas táctiles, que de las presentadas según sus características se procede a escoger una para la implementación del prototipo.

Tabla 10-1: Características de pantallas táctiles Raspberry.

Características	PANTALLA LCD HDMI TÁCTIL RESISTIVA	PANTALLA LCD HDMI TÁCTIL CAPACITIVA
Tamaño de pantalla	3.5" -7" LCD	4.3"-7" LCD
Resolución	480x320 - 1920x1080 60fps	800x480-1920x1080fps
Alimentación	GPIO Micro USB adicional	5V
Tecnología	Táctil resistiva	Táctil capacitivo
Compatibilidad	Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi 4	Cualquier dispositivo con salida HDMI
Sistema Operativo	Raspberry y derivados	

Fuente: (Raspberry Pi, 2019), (Tiendatec, 2019).

Realizado por: Berrones A, Crespata C; 2021.

Los puntos a considerar para la selección de una u otra pantalla varían entorno a las características en las que va a ser utilizada, pero para este prototipo en específico se busca que el dispositivo sea portátil, del menor tamaño posible sin que se pierda una visualización clara de la información que se presenta en la pantalla, además se debe utilizar pantallas que sean nativas de Raspberry ya que poseen una conexión directa entre la pantalla y la tarjeta Raspberry, es por esto que se escoge trabajar con la pantalla de menor tamaño y al ser de tecnología resistiva es compatible con los modelos de tarjetas a partir de la Raspberry Pi 2, ya que una capacitiva necesita un adaptador externo para la conexión con la tarjeta. Además de que la alimentación de una resistiva va directamente a los terminales GPIO de la tarjeta.

1.4.4 Batería

Para el desarrollo del prototipo, es necesario una batería ya que será la encargada de suministrar la energía a todo el sistema de visión artificial al ser un prototipo portátil. Una batería es un componente electrónico que convierte la energía química en energía eléctrica. de Los tipos de baterías comunes que se encuentran en el mercado son: baterías Pilas alcalinas, pilas de níquel-cadmio (Ni-Cd), pilas de hidruro de níquel-metal y pilas de hidruro de metal (Ni-MH), Baterías de iones de litio, baterías de polímero de litio (Li-Po). Para considerar un tipo de batería de otra se tienen diferentes parámetros al elegir la batería más adecuada para la aplicación; por ejemplo: voltaje nominal, capacidad de la batería, velocidad de descarga y peso (Cayambe y Castañeda, 2016).

En la tabla 11-1 se resumen algunas baterías de las que se encuentran en el mercado con sus características.

Tabla 11-1: Características de distintos tipos de baterías.

Tipo de batería	Tensión por celda	Tiempo de recarga (horas)	Auto descarga por mes (%)	Número de ciclos	Capacidad por tamaño(Wh/kg)	Precio
Plomo-acido	2 V	8-16	< 5	Medio	30-50	Bajo
Ni-Cd(níquel-cadmio)	1.2 V	1	20	Elevado	50-80	Medio
Ni-Mh(níquel-metal hydride)	1.2 V	2-4	20	Medio	60-120	Medio
Li ion(ión litio)	3.6 V	2-4	6	Medio-bajo	110-160	Alto
Li-po (polímero de litio)	3.7 V	2-4	5	Medio	130-200	Medio

Fuente: (Tiendatec, 2019), (Ramón y Pineda, 2015).
 Realizado por: Berrones A, Crespata C; 2021.

Para determinar la batería se tienen en cuenta parámetros como su auto descarga que de entre las presentadas, la batería Li-po es la que mejor mantiene su carga nominal, así como su voltaje de celda y su capacidad de energía por tamaño. En la figura 13-1 se aprecia un ejemplo de baterías de este tipo.



Figura 13-1: Batería Lipo.
 Fuente: (Cayambe y Castañeda, 2016)

1.4.5 GPS

En la actualidad el uso de un GPS (Global Position System) ha pasado a ser parte de los seres humanos en las actividades cotidianas, ya sean en industrias, en compañías telefónicas, en transporte para ubicación, y un sinnúmero más de aplicaciones, pues la utilización de un GPS permite ubicarse y tomar rutas con gran precisión. Un GPS necesita estar conectado a una red no menor de 24 satélites, que son distribuidos a 6 planos orbitales, cada uno con 4 satélites. Estos satélites toman orbitas que se distribuyen de modo que sean visibles desde cualquier punto de la Tierra en cualquier momento, puesto que cada satélite cuenta con un reloj atómico que funciona con 1 nano segundo de precisión para poder detallar la ubicación con gran exactitud (Villegas, 2020).



Figura 14-1: Modulo GPS.

Fuente: (Villegas, 2020).

El aporte principal del GPS para el prototipo es la de mostrar rutas hacia uno u otro destino, todos los elementos que conforman el hardware deben ser de conexión directa con la tarjeta Raspberry ya que debido a esto se logra tener un equipo compacto y de fácil transporte, en especial en la selección de este dispositivo, para este tema se basa entorno al precio del elemento. En la tabla 12-1 se aprecian las características más importantes de diferentes familias de módulos GPS.

Tabla 12-1: Características de módulos GPS.

Características	Familia U-blox	Familia GARMIN
Alimentación	3.5-5 V	8-40 V
Frecuencia	L1	L1
Satélites	4	12
Margen de error promedio	10 m	3-5 m
Consumo de corriente	100 mA	65 mA
Tiempo de inicio promedio	35 s	38 s
Antena	Separada	Incorporada
Costo \$	Bajo	Alto

Fuente: (Avelectronics, 2020)(Techmake, 2020), (Campbell, 2021).

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 12-1 se observa las principales características de GPS fabricados por U-blox y los de la marca GARMIN. Para la construcción del prototipo se escoge al fabricante U-blox, los cuales se los puede encontrar fácilmente en el mercado nacional, como son los de la familia NEO, además de tener un bajo costo, también poseen un tiempo de inicio más rápido y su alimentación varía de 3 a 5 voltios.

1.5 Software

Los programas de visión artificial están conformados por diferentes partes como: entorno visual de pruebas, procesado y estudio; lenguajes de programación y librerías de desarrollo, usualmente los programas de visión artificial poseen un entorno en el cual se puede capturar la imagen, procesarlas con diferentes algoritmos y determinar las funciones a utilizar además que poseen un lenguaje de programación en el cual se crean las rutinas y subrutinas con las que debe cumplir dichos programas. Cuanto más completa sea la librería de funciones, más fácilmente se podrá solucionar los problemas que se vayan presentando en cuanto al desarrollo del programa, es por esta razón que se deben seleccionar el Software necesario para trabajar en visión artificial así como utilizar librerías más funcionales entorno a este campo, las cuales faciliten el desarrollo de los programas (Gonzales et al., 2016).

1.5.1 Sistema operativo

Todo sistema de visión artificial se encuentra desarrollado sobre una plataforma o sistema operativo, que tenga soporte para el uso de librerías de visión artificial, en la actualidad todas las plataformas que se encuentran en el mercado tienen estas características, en la tabla 13-1 se presenta una comparativa de los sistemas operativos más conocidos (Sivisapa, 2014).

Tabla 13-1: Ventajas y desventajas de los sistemas operativos.

Sistema operativo	Ventajas	Desventajas
Windows	<ul style="list-style-type: none"> • Interfaces amigables al usuario. • Múltiples versiones. • Posibilidad de integrar recursos multimedia (textos, imagen y sonido). 	<ul style="list-style-type: none"> • Falta de estabilidad en muchas versiones. • No gratuito y poco flexible. • Requiere de equipos potentes y con alta memoria.
Linux	<ul style="list-style-type: none"> • Sistemas estables y seguros. • Software libre. • Flexible. • Gran seguridad. • No cuenta con una sola empresa que lo respalde. 	<ul style="list-style-type: none"> • No es fácil de usar por usuarios corrientes. • Documentación y terminología muy técnica. • Incompatibilidad con programas de Windows.
Mac	<ul style="list-style-type: none"> • Basado en Linux. • Mejor Interfaz gráfica del mercado. • Utilización sencilla. • Gran seguridad. 	<ul style="list-style-type: none"> • Hardware Exclusivo. • Dificultad al encontrar software. • Costoso.
Raspbian Jessie	<ul style="list-style-type: none"> • Versión de Debian adaptada y optimizada para Raspberry Pi. • Distribución libre y gratuita. • Raspbian es estable y es el que mejor aprovecha el rendimiento de la Raspberry Pi • La seguridad es una de las prioridades para su comunidad, los virus no serán una preocupación. 	<ul style="list-style-type: none"> • Se necesita un previo conocimiento en Linux para poder usar con comodidad el sistema operativo. • Los dispositivos externos como pendrives, etc. tienen que ser montados por el usuario.

Fuente: (Sivisapa, 2014).

Realizado por: Berrones A, Crespata C; 2021.

Cada sistema operativo cuenta con diferentes características en la cual se puede trabajar un sistema de visión artificial, pero para la selección de uno de estos se enfoca principalmente en que sea un software libre y tenga un buen rendimiento con la tarjeta escogida anteriormente la Raspberry Pi, por lo cual la elección es la del sistema operativo Raspbian Jessie.

1.5.2 Lenguaje de programación

Se utilizan para el desarrollo de programas o conjuntos de instrucciones que sean ejecutadas por los ordenadores, existen muchos de ellos con licencias libres y comerciales a pesar de ello para la elección de uno u otro lenguaje de programación se debe realizar un análisis de las características y necesidades que requiera el prototipo, a más de eso debe brindar soporte para las diferentes librerías especializadas para visión artificial. En la tabla 14-1 se observa los lenguajes de programación que soportan la integración de librerías de visión artificial así como sus ventajas y desventajas (Sivisapa, 2014).

Tabla 14-1: Lenguajes de programación

Lenguaje de programación	Ventajas	Desventajas
Java	<ul style="list-style-type: none"> • Multiplataforma. • Portable. • Lenguaje Simple. • JRE para cualquier plataforma. • Orientado a objetos. 	<ul style="list-style-type: none"> • Velocidad. • Funcionalidad escasa o bajo soporte para visión artificial. • No toda la plataforma de java es libre.
Matlab	<ul style="list-style-type: none"> • Amplio soporte. • Alta precisión. • Soporte de OpenCV. 	<ul style="list-style-type: none"> • No es de código abierto. • Problemas eventuales de velocidad.
C++	<ul style="list-style-type: none"> • Rapidez. • Orientado a Objetos. • Compilado. • Código abierto. • Lenguaje de bajo nivel. 	<ul style="list-style-type: none"> • No portable. • Mayor control =Mayor dificultad. • Manejo de librerías es más complicado.
Python	<ul style="list-style-type: none"> • Multiplataforma. • Comunidad. • Multiparadigma. • Código Abierto. • Lenguaje de alto nivel. • Fácilmente portable a otras plataformas entre las que se incluye Raspberry Pi. • Permiten realizar tareas complejas en pocas líneas de código. • Compatible con Raspbian. • Mayor soporte de librerías de visión artificial (OpenCV). • Bajo consumo de recursos. 	<ul style="list-style-type: none"> • La mayoría de servidores no poseen soporte a Python. • La mayoría de los servidores no tienen soporte a Python, y si lo soportan, la configuración es difícil.

Fuente: (Sivisapa, 2014), (Amaya et al., 2016).

Realizado por: Berrones A, Crespata C; 2021.

Cada lenguaje de programación cuenta con diferentes características en la cual se puede trabajar visión artificial, pero para la selección de uno de estos se enfoca principalmente en que sea de código abierto, sea compatible con el sistema operativo Raspbian Jessie y posea gran soporte de librerías de visión artificial por lo cual la elección es Python.

Python es un lenguaje de programación de código abierto, alto nivel en el cual se puede modificar el o los programas en cualquier momento, además ofrece la ventaja de tener en su estructuración de lenguaje realizar tareas complejas en pocas líneas de código es decir un código fácil y simple, donde cualquier usuario puede entender y analizar el código que se encuentre implementado en las diferentes aplicaciones que se puedan dar. Existe también la facilidad de que al ser un software libre se puede encontrar una gran cantidad de información donde le permiten al usuario aprender a utilizar este lenguaje de manera fácil y rápida, lo más importante de este lenguaje es que al ser de alto nivel se consideraría que puede tener un exceso de consumo de los recursos de un computador, sin embargo, es todo lo contrario, es un lenguaje donde su consumo de recursos es bajo lo que conlleva a considerarlo el lenguaje libre más óptimo para el trabajo en el desarrollo de aplicaciones de visión artificial (Amaya et al., 2016).

CAPÍTULO II

2 PROPUESTA Y DISEÑO DE PROTOTIPO

En el presente capítulo se detalla cada una de las etapas que conforman parte del sistema de visión artificial que se enfoca en el desarrollo metodológico de la implementación de un prototipo de información de circuitos turísticos del Centro Histórico de la ciudad de Riobamba, utilizando visión artificial y los elementos necesarios para realizar el trabajo. Se detallará las etapas, el diseño y armado del prototipo necesario para el reconocimiento de la señalética turística. Después se inspeccionará el montaje electrónico para controlar la comunicación del sistema de visión artificial. Así como también se especifica sus características tanto como de hardware y de software.

2.1 Requerimientos del sistema

Para cumplir con las diferentes características que proporcionará el dispositivo, y teniendo consideraciones de las necesidades del sistema general tanto en *hardware* y *software*, a continuación, se enlista cada una de ellas.

- Ubicación precisa de la cámara en la parte posterior del prototipo para lograr capturas de la escena.
- Cámara con resolución mínima de 8 megapíxeles para la adquisición de imágenes.
- Tarjeta Raspberry Pi 4B con memoria RAM de 2GB.
- Desarrollo del prototipo en el entorno planteado (Centro Histórico de Riobamba) en ambientes con luz natural entre 85000 y 120000 luxes, ante un buen estado de la señalética.
- Realizar el reconocimiento de la señalética turística con el algoritmo SIFT y FLANN.
- Almacenar la información (localización geográfica, historia, imágenes sobre cada lugar turístico) dentro de la memoria interna del dispositivo obtenida previamente por la señalética turística y esta a su vez se encuentra almacenada en una base de datos.
- Mostrar la información almacenada de la base de datos del prototipo, rutas, y sugerencias de los circuitos turísticos en una pantalla en donde se detalle cada uno de estos datos.
- Ser un sistema de bajo costo.
- Ser de fácil manejo para los usuarios.

2.2 Concepción general del sistema

En un futuro se espera que el prototipo sea personal, por el momento se encuentra destinado hacia agencias que promocionen el turismo del Centro Histórico de Riobamba, el cual va a permitirle al usuario guiarle a los diferentes atractivos turísticos, mediante una captura de imagen que el prototipo reconoce y que traza la ruta para llegar al destino de interés.

La concepción general del sistema se aprecia en la figura 1-2. En donde se observa cada etapa y componente del dispositivo, este diseño está constituido por un proceso que se dedica a la adquisición de cada una de las imágenes, estas imágenes son almacenadas en la tarjeta SD del controlador Raspberry donde posteriormente en la segunda etapa se realiza el procesamiento digital de las imágenes para ser sometidas al algoritmo de reconocimiento SIFT y FLANN para la señalética, una vez finalizado el tratamiento de la imagen y al obtener la información necesaria de la cámara, esta se coteja con una pequeña base de datos que posee el sistema para decidir si la imagen que aparece en escena posee características de una señalética turística, para que de acuerdo a eso se presente la información y recorrido turístico por las rutas.

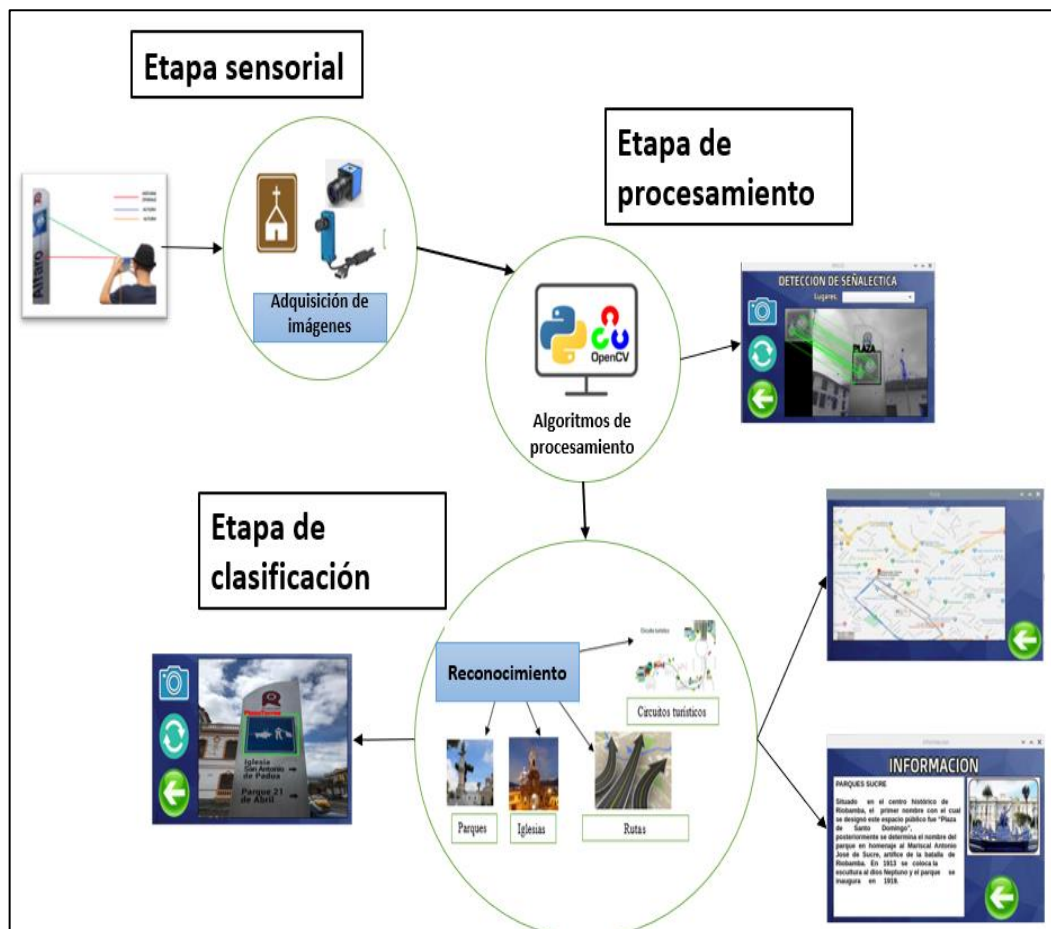


Figura 1-2: Concepción general del sistema.
Realizado por: Berrones A, Crespata C; 2021.

2.3 Selección de los componentes necesarios para el hardware del prototipo

En el siguiente apartado se realiza la selección y el detalle de los componentes electrónicos necesarios para la implementación del prototipo.

2.3.1 Hardware del prototipo

En esta sección se realiza la selección de los componentes físicos que requiere el prototipo para poder ser armado y montado.

2.3.1.1 Tarjeta Raspberry Pi 4B

Desarrollada por la empresa Raspberry, conformada por 40 terminales GPIO con una memoria RAM de 2 GB con dimensiones de 8.5x5.35cm. En la tabla 1-2 se puede observar sus principales características, y en la figura 2-2 se puede apreciar el elemento (Raspberry Pi, 2019).

Tabla 1-2: Características de la tarjeta controladora Raspberry Pi 4B

Características	Descripción
Procesador	Quad core 64-bit ARM-Cortex A72 r
Memoria RAM	2GB
Conectividad Inalámbrica	802.11 b/g/n/ac Bluetooth 5.0 with BLE
Frecuencia	1.5GHz
Conectividad de Red	Gigabit Ethernet
Puertos USB 2.0	2
Puertos USB 3.0	2
Salidas de Video	Supports dual HDMI display output up to 4Kp60
Almacenamiento	Tarjeta SD
Periféricos Bajo Nivel	40 x GPIO
Corriente	2.5A
Alimentación	5V
Dimensiones	8.5x5.35cm

Fuente: (Raspberry Pi, 2019).

Realizado por: Berrones A, Crespata C; 2021.

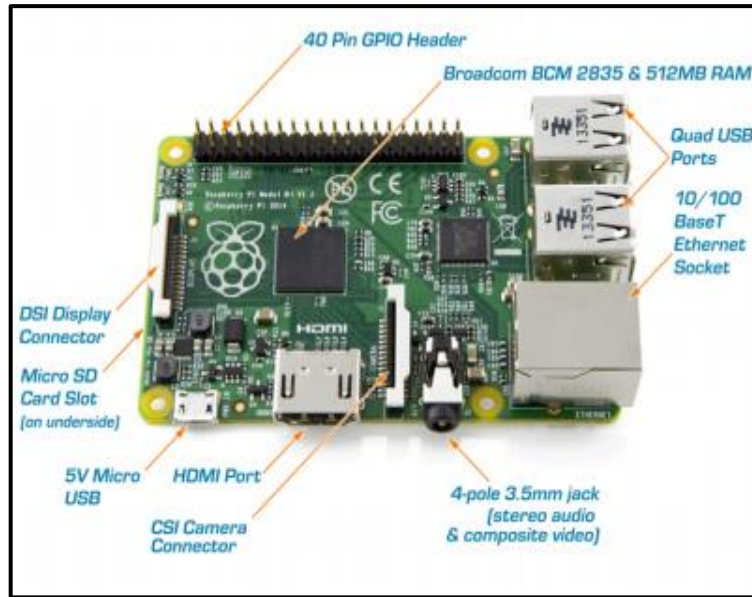


Figura 2-2: Tarjeta Raspberry Pi 4B.
Fuente:(Cayambe y Castañeda, 2016).

2.3.1.2 Cámara Pi NoIR V2

Desarrollada por la empresa Raspberry, con una resolución de 8 megapíxeles, con dimensiones de 5x24x9mm y un peso de 3 gramos, se presenta en la tabla 2-2 sus principales características, mientras que en la figura 3-2 se puede apreciar el elemento (Raspberry Pi, 2019).



Figura 3-2: Pi NoIR Camera V2.
Realizado por: Berrones A, Crespata C; 2021.

Tabla 2-2: Característica de la Camara Pi NoIR V2

Características	Descripción
Tamaño	5x24x9 mm
Peso	3 g
Resolución	8 megapíxeles
Modos de video	1080p30, 720p60 and 640 × 480p60/90
Integración Linux	V4L2 driver available
Sensor	Sony IMX219
Sensor de Resolución	3280 × 2464 pixeles
Tamaño óptico	1/4"
Campo de visión horizontal	62.2 grados.
Campo de visión vertical	48.8 grados.

Fuente: (Raspberry Pi, 2019).

Realizado por: Berrones A, Crespata C; 2021.

2.3.1.3 Pantalla LCD táctil Kuman 3,5" 480X320 HDMI

Desarrollada por la empresa Raspberry, con un tamaño de pantalla de 3.5 pulgadas, con dimensiones de 13.4x10.10cm y tecnología resistiva, se presenta en la tabla 3-2 sus principales características, mientras que en la figura 4-2 se puede apreciar el elemento (Raspberry Pi, 2019).

Tabla 3-2: Características de la pantalla LCD táctil Kuman 3.5" 480X320 HDMI.

Características	Descripción
Tamaño de pantalla	3.5" LCD.
Resolución	480x320 ~ 1920x1080 60fps.
Alimentación	GPIO. Micro USB adicional.
Tecnología	Táctil resistiva.
Compatibilidad	Raspberry Pi 2, Raspberry Pi 3, Raspberry Pi 4.
Sistema Operativo	Raspberry y derivados.

Fuente: (Tiendatec, 2019).

Realizado por: Berrones A, Crespata C; 2021.



Figura 4-2: Pantalla LCD TÁCTIL Kuman 3.5" 480X320 HDMI.

Realizado por: Berrones A, Crespata C; 2021.

2.3.1.4 Módulo GPS NEO-6M

Desarrollado por la empresa china Shenzhen Wonder-Chip Electronics Company Limited, tiene un tamaño de 23x30mm con memoria EEPROM. En la tabla 4.2 se aprecian sus principales características, mientras que en la figura 5-2 se puede observar el elemento (Techmake, 2020).

Tabla 4-2: Características del Módulo GPS NEO-6M.

Características	Descripción
Voltaje de entrada	3V -5V.
Interface	Serial UART 5V.
Antena	Cerámica
Memoria	EEPROM
Indicador de señal	LED
Baud rate	9600 bps.
Precio	\$16.90

Fuente: (Techmake, 2020).

Realizado por: Berrones A, Crespata C; 2021.



Figura 5-2: Módulo GPS NEO-6M.

Realizado por: Berrones A, Crespata C; 2021.

2.3.2 Estructura del dispositivo

El diseño de la estructura para el dispositivo se lo realizó mediante el software SolidWork versión 2016. El motivo por el que se decidió trabajar con el mencionado software es porque permite diseñar y modelar en 3D, así también se pudo crear la estructura con material PLA para su impresión.

Luego de contar con todos los elementos del dispositivo para armarlo, se continúa con el diseño de la estructura o carcasa para proteger y manipular de una mejor manera el prototipo. Para realizar el diseño se procede a tomar las medidas de cada uno de los componentes, sin despreciar o dejar pasar por el alto el más mínimo detalle para que la estructura pueda coincidir perfectamente con el dispositivo.

La estructura está conformada por 2 piezas. La primera es el cuerpo o compartimento donde se va a asentar el dispositivo. Mediante la figura 6-2 se presenta el diseño modelado en 3D de esta pieza.

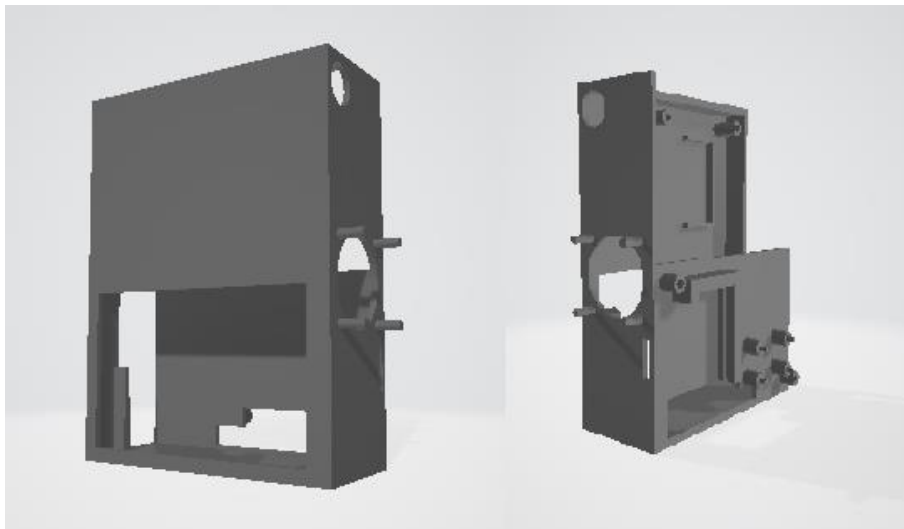


Figura 6-2: Diseño del compartimento principal 3D de la estructura.
Realizado por: Berrones A, Crespata C; 2021.

La segunda pieza es la tapa del compartimento principal que va a permitir cubrir y proteger el dispositivo. Mediante la figura 7-2 se presenta el diseño modelado en 3D de esta pieza.

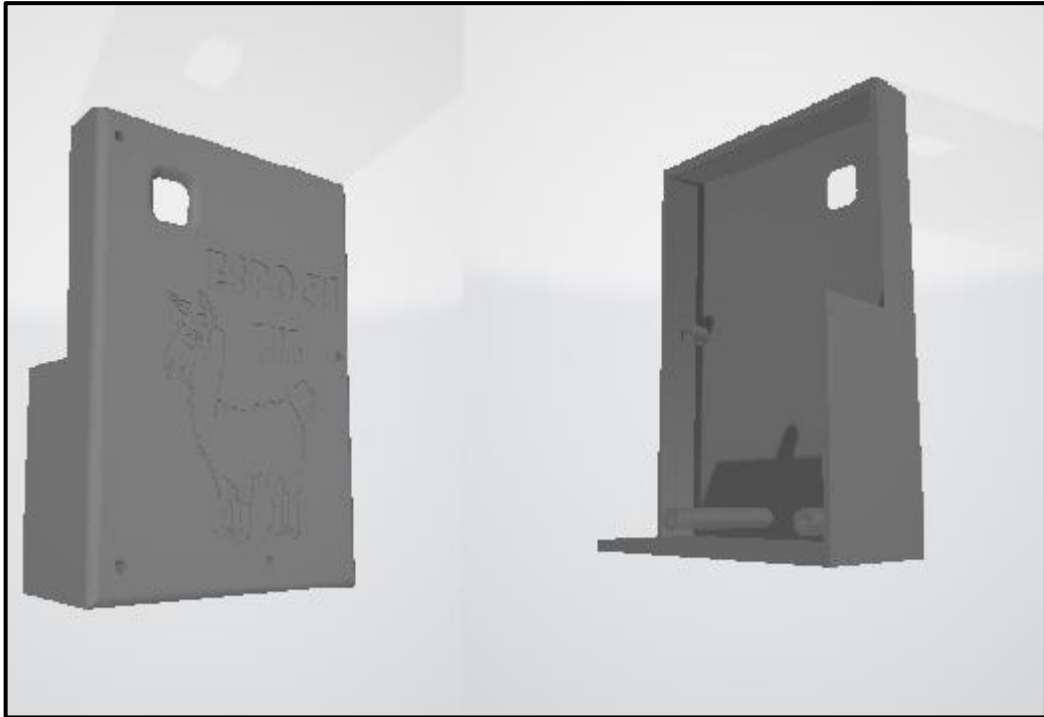


Figura 7-2: Diseño de la tapa 3D del compartimento principal de la estructura.
Realizado por: Berrones A, Crespata C; 2021.

Finalmente, en la figura 8-2 y en la figura 9-2 se observa las 2 piezas diseñadas, montadas y ensambladas como resultado final del dispositivo, mientras que en la figura 10-2 se observa el prototipo totalmente armado.

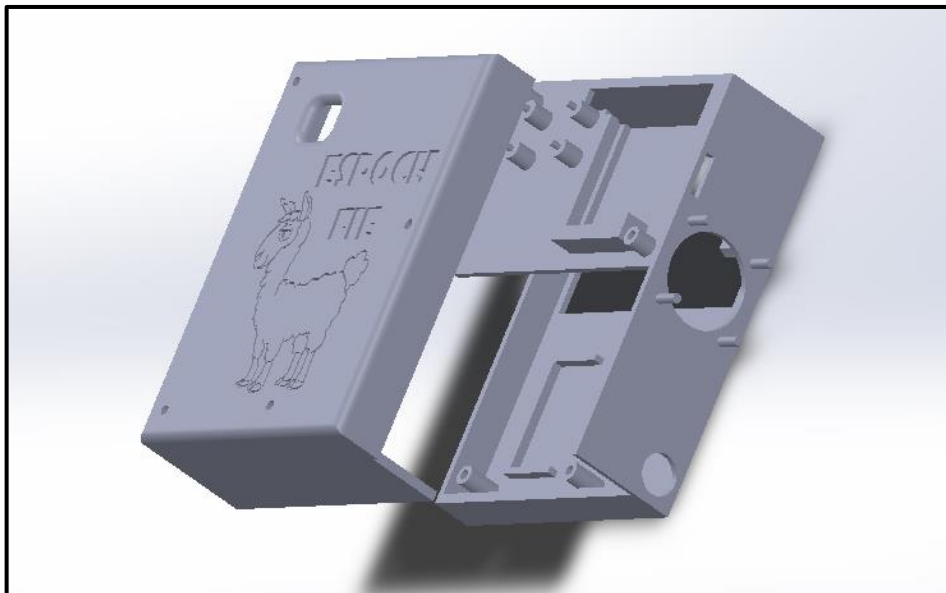


Figura 8-2: Montaje de la estructura en 3D.
Realizado por: Berrones A, Crespata C; 2021.



Figura 9-2: Estructura armada en 3D.
Realizado por: Berrones A, Crespata C; 2021.



Figura 10-2: Estructura montada y armada.
Realizado por: Berrones A, Crespata C; 2021.

En el ensamblaje final de la estructura, las dimensiones totales del dispositivo son 90.4 mm de ancho, 121.4 mm de largo y 41.11 mm de altura. Las dimensiones en detalle se pueden observar en el Anexo C.

2.3.3 Esquema general de conexiones del prototipo

En la tabla 5-2 se aprecia el número del terminal de cada elemento que va conectado el número del terminal de la tarjeta Raspberry, mientras en el Anexo D se puede observar el esquema general de conexión del circuito realizado en Fritzing. Se debe tomar en cuenta que la tarjeta tiene un puerto exclusivo para la conexión de terminales con la cámara donde van desde el terminal 1 hasta el terminal 15. Finalmente en la figura 11-2 se puede ver el esquema electrónico en 3D.

Tabla 5-2: Conexión de terminales entre la tarjeta Raspberry y elementos.

Cámara	Tarjeta Raspberry Puerto Cámara	Pantalla	Tarjeta Raspberry GPIO	GPS	Tarjeta Raspberry GPIO	Batería	Tarjeta Raspberry GPIO
1	1	1	1	VC	2	Batería +	2
2	2	2	2	RX	8	Batería -	25
3	3	4	4	TX	10		
4	4	6	25	GND	6		
5	9	9	20				
6	10	11	11				
7	11	14	6				
8	12	17	17				
9	5	18	18				
10	6	19	19				
11	7	20	14				
12	15	21	21				
13	8	22	22				
14	13	23	23				
15	14	24	24				
		25	9				
		26	26				

Realizado por: Berrones A, Crespata C; 2021.

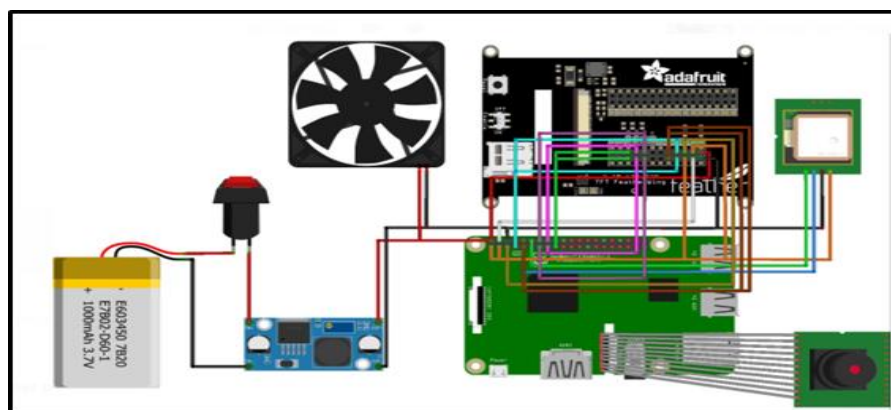


Figura 11-2: Esquema electrónico 3D.

Realizado por: Berrones A, Crespata C; 2021.

2.4 Diseño del software

En este apartado se describen cada una de las etapas que el prototipo va a desarrollar para lograr su funcionamiento, el cual es brindarle información acerca de los destinos turísticos y sus rutas en el Centro Histórico de la ciudad de Riobamba.

2.4.1 Etapa de adquisición

Esta etapa se basa en la adquisición de la imagen de su entorno para lograr el tratamiento de la misma y obtener características que permitan identificar el tipo de señalética turística a la que pertenece la imagen en la figura 12-2 se puede apreciar el diagrama de bloques de esta etapa, inicia con la captura de la imagen por medio de la cámara, la cual se encuentra conectada al puerto de la Raspberry, para de esta manera enviar los datos obtenidos al microprocesador.

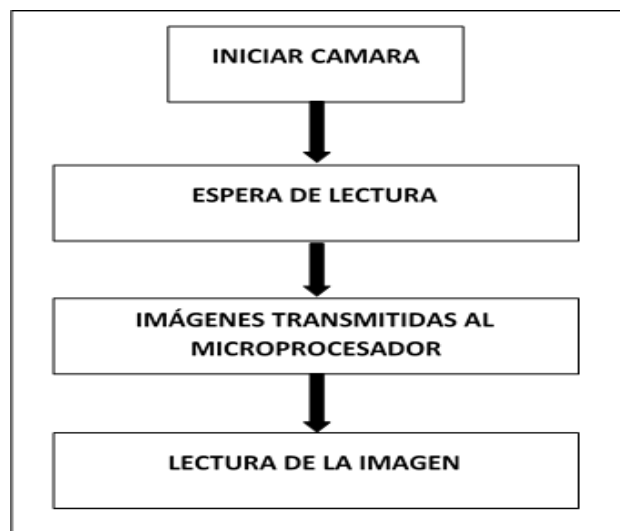


Figura 12-2: Etapa de adquisición de imagen.
Realizado por: Berrones A, Crespata C; 2021.

2.4.2 Etapa de procesamiento

Las imágenes obtenidas para el procesamiento en el momento de la adquisición pueden generar ciertos fallos para la aplicación final destinada, algunos de estos inconvenientes son pérdida de definición de la imagen debido a un mal enfoque de la cámara, es por esta razón que esta etapa se utilizan diversos mecanismos para contrarrestar estos efectos, buscando reparar los deterioros de la imagen obtenida que fueron generados por el hardware, en la figura 13-2 se muestra el diagrama de bloques de esta etapa.

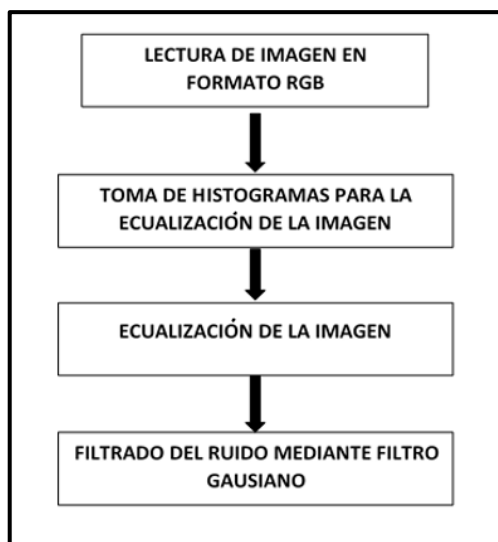


Figura 13-2: Etapa de procesamiento.
 Realizado por: Berrones A, Crespata C; 2021.

2.4.3 Etapa de interpretación de resultados

Esta etapa se designa para la interpretación y reconocimiento de la señalética, basada principalmente en etiquetas que se le da al objeto, en este caso a la imagen adquirida pero ya procesada, para de esta manera tener un significado del conjunto de objetos y presentar el conjunto de respuestas requeridas por el usuario. En la figura 14-2 se aprecia el diagrama de bloques de esta etapa.

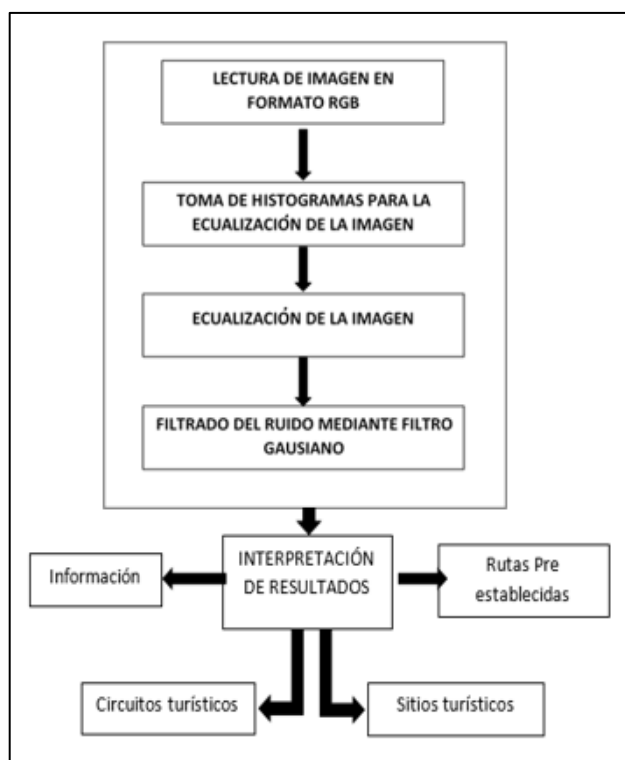


Figura 14-2: Etapa de interpretación de resultados.
 Realizado por: Berrones A, Crespata C; 2021.

2.5 Software

En este apartado se detalla el algoritmo con el que se ejecuta el prototipo, partiendo de un diagrama general, se va desglosando y explicando cada una de sus funciones.

2.5.1 *Software del prototipo*

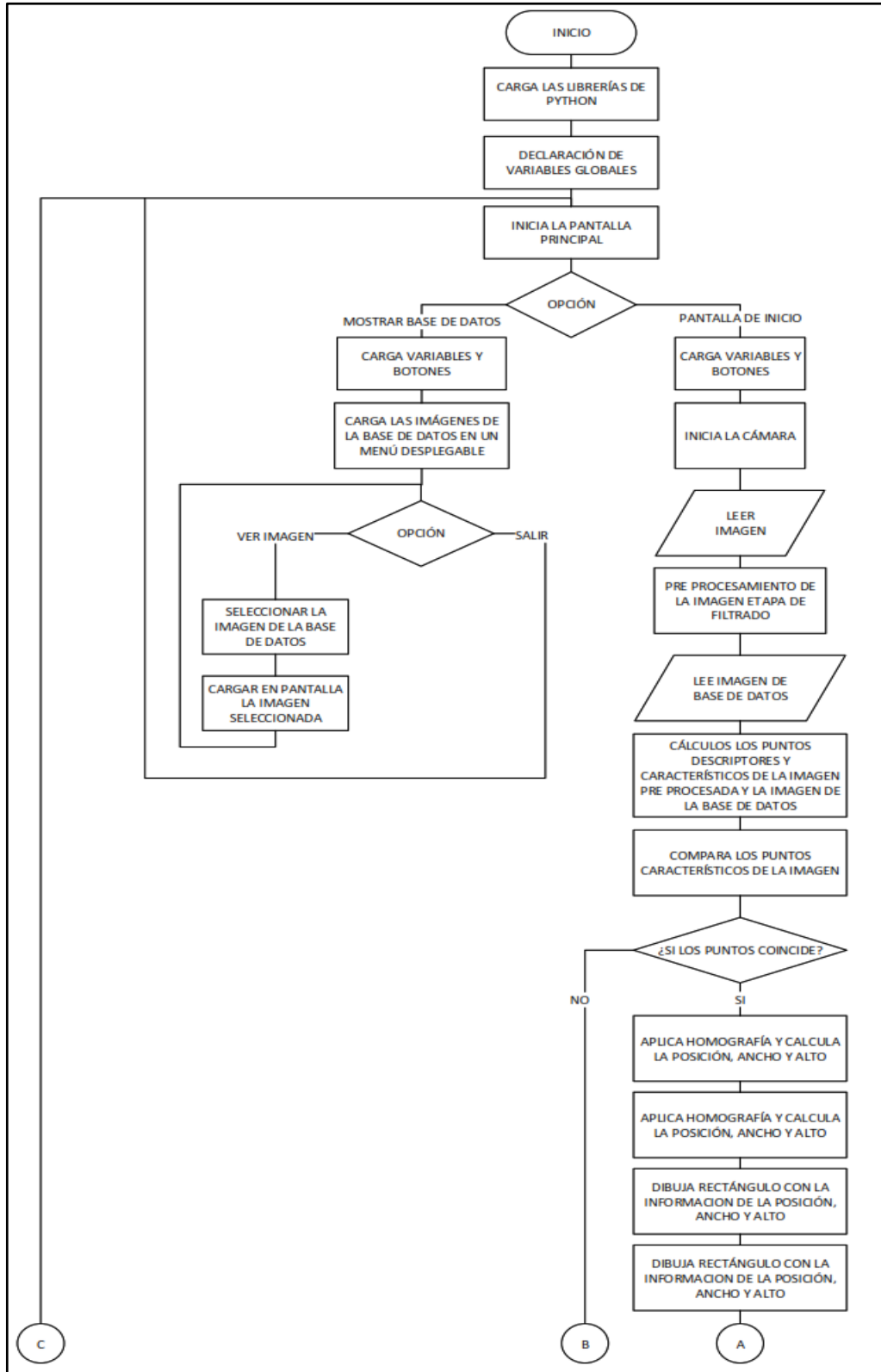
Se encuentra conformado por cinco funciones, las cuales contribuyen a cumplir con el desarrollo de cada característica presente en el dispositivo, por lo que se ha hecho uso del lenguaje de programación Python versión 3.9, un software libre, óptimo para el trabajo de visión artificial. En la figura 15-2 se aprecia el diagrama general de funcionamiento del prototipo el cual consta de dos etapas la de seteo y la de lazo infinito, que engloban las funciones que posteriormente se realiza un detalle de cada una.

Primeramente para llegar a la ejecución de cada una de estas funciones se debe cargar las librerías y declarar las variables globales, para después darle al usuario la opción de iniciar con el reconocimiento de la señalética u observar las imágenes que se encuentran almacenadas en la memoria del prototipo, si la selección es visualizar la biblioteca de imágenes en un inicio se cargan variables y botones de la interfaz gráfica, consecutivamente despliega las imágenes de la base de datos para finalizar con la presentación en pantalla y la opción de salir y regresar a la pantalla principal, todo esto es la etapa de seteo.

Caso contrario selecciona reconocimiento de imágenes en el cual se tiene el inicio de la cámara y captura la imagen de la señalética, para aplicar un filtro mediano, posteriormente el algoritmo realiza el cálculo de los puntos descriptores, característicos de la imagen adquirida y de la imagen que se encuentra en la memoria del prototipo. Luego se comparan los puntos entre las 2 imágenes, en el caso de que coincidan se aplica la homografía, se coloca el nombre del objeto y se muestra el mensaje “Señalética encontrada”, en el caso de no coincidir se presenta el mensaje de “No se encontró señalética” y se procede a ejecutar la Función OCR, este proceso lo realiza una sola vez por cada imagen que sea tomada por el prototipo, las pruebas de tiempo que tarda en realizar en el procesamiento se las presenta en el capítulo 3.

Una vez detectado la señalética o nombre de un atractivo se procede a ejecutar a la función CONFIGURACION(), encargada del inicio y configuración del GPS, finalizado esta función se tiene dos opciones a elegir, la primera la función MOSTRAR_RUTA() encargada de la visualización del trazado de la ruta, la segunda NAVEGACION() la cual muestra la ubicación actual del prototipo entorno al punto de llegada para finalmente ejecutar INFORMACION() la

cual muestra datos del sitio turístico, todo esto conforma la etapa de lazo infinito la cual una vez finalizado regresa a la etapa de seteo para empezar nuevamente con la ejecución inicial del programa .



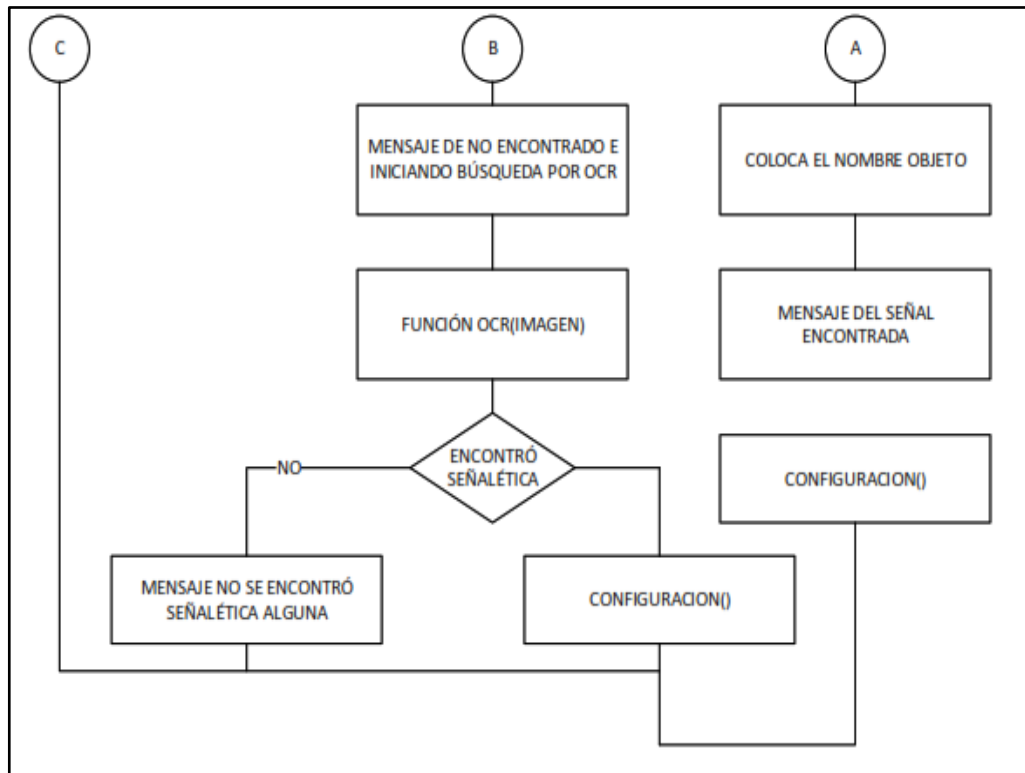


Figura 15-2: Diagrama de flujo general de funcionamiento.
 Realizado por: Berrones A, Crespata C; 2021.

Función OCR(). El objetivo de esta función es el reconocimiento de palabras o nombres de los atractivos turísticos existentes en los diferentes puntos de información del Centro Histórico, para su ejecución no requiere parámetros previos en la figura 16-2 se aprecia su diagrama de flujo, el cual se inicia con la lectura de la imagen capturada, la cual posteriormente pasa a la etapa de pre procesamiento en donde se aplica un filtro mediano encargado de eliminar distorsiones o ruidos producidos en la captura.

Después procede a la aplicación del OCR configurando el idioma, transformando así la imagen a texto para posteriormente compararlo con las imágenes almacenadas en la memoria del prototipo, donde si coincide el texto se llama a la función CONFIGURACION(). Por otro lado en el caso de no coincidir se presenta el mensaje “No se encontró coincidencias”.

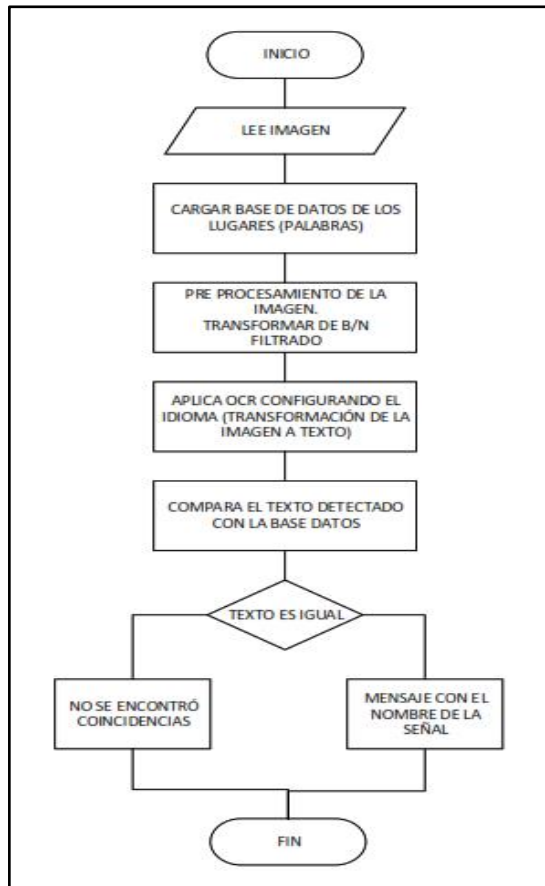


Figura 16-2: Diagrama de flujo de la Función OCR.
 Realizado por: Berrones A, Crespata C; 2021.

Función CONFIGURACION(). Tiene por objetivo la configuración del GPS para obtener los datos de ubicación, los cuales se utilizaran para el trazado de rutas hacia los diferentes destinos, no posee parámetros de entrada y como valor de retorno arroja los datos de longitud y latitud del dispositivo además del nombre del destino de llegada , para posteriormente utilizarlos en las funciones MOSTRAR_RUTA() o iniciar NAVEGACIÓN().

En la figura 17-2 se aprecia su diagrama de flujo, en el cual se inicia con la lectura de la ubicación del GPS y posterior se carga de la base de datos el posicionamiento de los sitios turísticos, inmediatamente se configura la biblioteca gmaps para la ubicación de los puntos tanto de partida como de llegada para el trazado de rutas, a continuación carga la lista de lugares de acuerdo a la señalética detectada.

Finalmente lee el sitio a donde se desee dirigir, y se tiene la opción de pasar a las siguientes funciones. Para cargar la información de los lugares de la base de datos existe un documento en Excel creado exclusivamente para la aplicación donde se almacena la posición geográfica, la posición en calles, la información turística del lugar, las imágenes del lugar que se presentan en la aplicación. En la tabla 6-2 se puede observar un ejemplo de los campos con los datos del Parque

Sucre almacenados en el documento de Excel. Todos estos datos son cargados de acuerdo a la señalética que se seleccionó anteriormente.

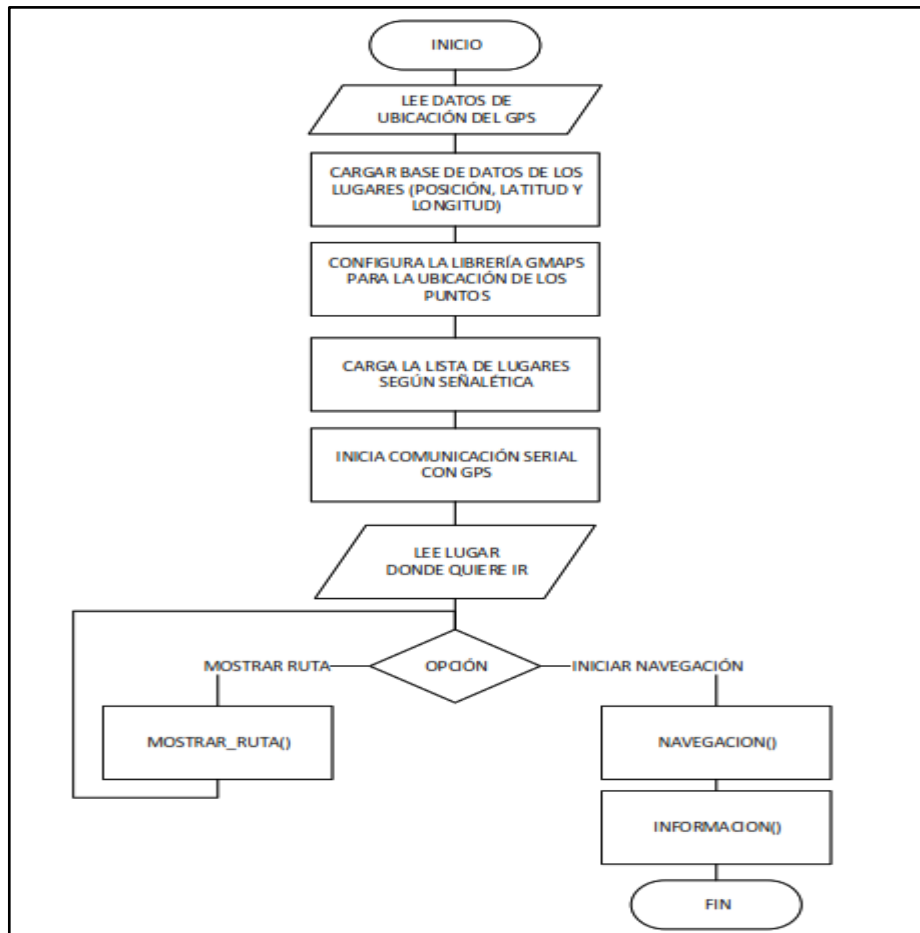


Figura 17-2: Diagrama de flujo función CONFIGURACION().
Realizado por: Berrones A, Crespata C; 2021.

Tabla 6-2: Ejemplo de la información que contiene la tabla en Excel.

Lugar	Latitud	Longitud	Dirección	Información	Nombre de la Imagen
PARQUE SUCRE	-1.671847	-78.650427	10 de Agosto, España, Primera Constituyente y Juan Larrea	Situado en el Centro Histórico de Riobamba, el primer nombre con el cual se designó este espacio público fue “Plaza de Santo Domingo”, posteriormente se determina el nombre del parque en homenaje al Mariscal Antonio José de Sucre, artífice de la batalla de Riobamba. En 1913 se coloca la escultura al dios Neptuno y el parque se inaugura en 1919.	sucre.jpg

Realizado por: Berrones A, Crespata C; 2021.

Función MOSTRAR _RUTA(). Tiene por objetivo la visualización de la ruta hacia el destino turístico una vez finalizado la detección de señalética y obtenido los datos del GPS, como parámetros de entrada tiene la posición del prototipo, no posee valor de retorno, en la figura 18-2 se observa su diagrama de flujo, en el cual se inicia con la lectura del parámetro de entrada para posteriormente cargar de la base de datos el posicionamiento del sitio turístico seleccionado, después carga la ruta establecida y finalmente muestra en pantalla la ruta en el mapa hacia su destino y posee la opción de regresar a la función principal CONFIGURACION().

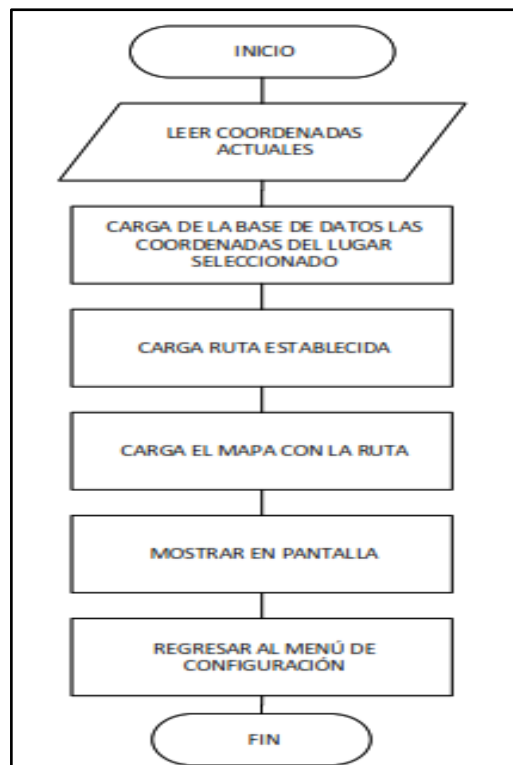


Figura 18-2: Función MOSTRAR RUTA().
Realizado por: Berrones A, Crespata C; 2021.

Función NAVEGACIÓN(). Permite visualizar la navegación desde el punto actual del prototipo hacia el punto de llegada, esta función tiene como parámetros de entrada la posición del prototipo, no posee valor de retorno, en la figura 19-2 se observa su diagrama de flujo, en el cual se inicia con la lectura del parámetro de entrada después carga de la base de datos las coordenadas del sitio turístico seleccionado y continúa con la carga del mapa con la ruta a continuación el prototipo realiza una lectura del posicionamiento actual del prototipo, para calcular la distancia entre el punto actual y el de llegada.

Finalmente realiza una comparación con una distancia de 10 metros al punto de llegada si esta es igual o menor el prototipo mediante su aplicación le presenta el mensaje “Llegó al sitio” y

procederá a ejecutar la función INFORMACION(), caso contrario continuará realizando la comparación.

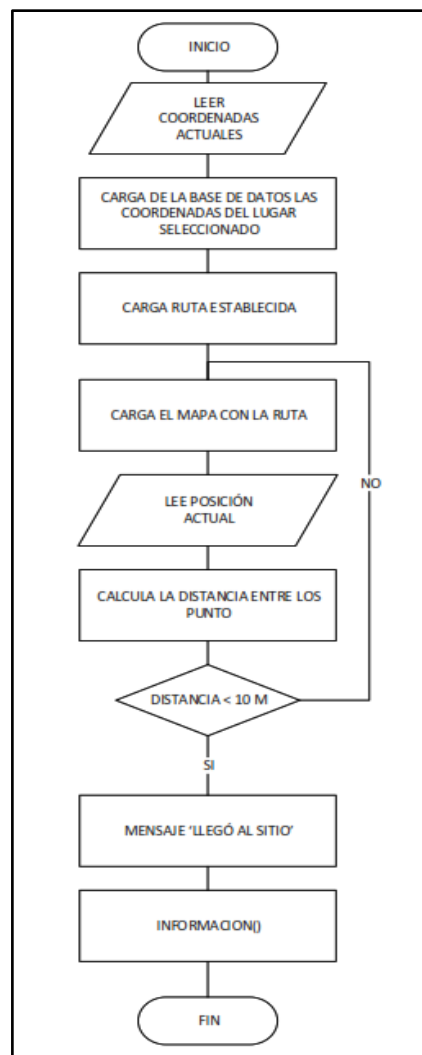


Figura 19-2: Función NAVIGACIÓN().
Realizado por: Berrones A, Crespata C; 2021

Función INFORMACIÓN(). Una vez el usuario llega a su destino el objetivo de la función es mostrar la información del sitio turístico y una foto del mismo, posee como parámetro de entrada el nombre del destino de llegada proporcionada por la función CONFIGURACION(), a continuación en la figura 20-2 se muestra el diagrama de flujo de esta función del dispositivo, en la cual se inicia con la lectura del sitio turístico de destino o parámetro de entrada, posterior se carga la información e imagen almacenada en la base de datos, después se transforma la información a texto para que se pueda visualizar, finalmente se procede mostrar la información en pantalla.

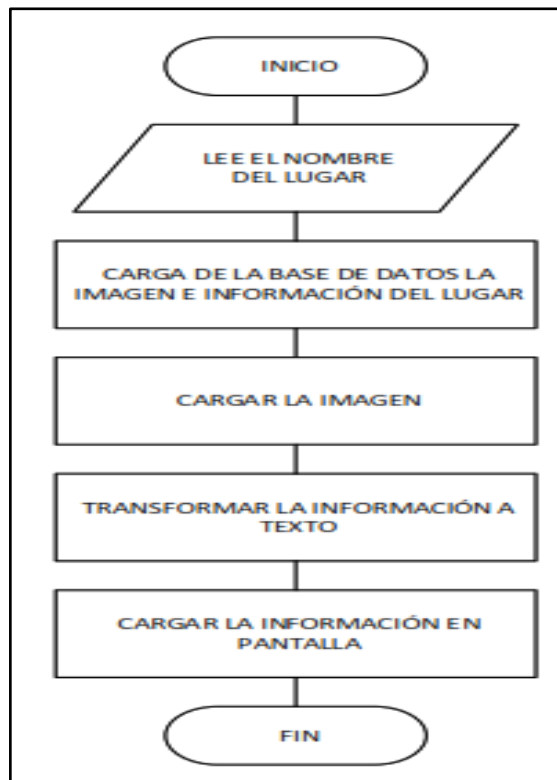


Figura 20-2: Diagrama de flujo función INFORMACION().
Realizado por: Berrones A, Crespata C; 2021.

2.6 Diseño de la interfaz del prototipo

Para desarrollar la aplicación en el prototipo, se crea una interfaz que interactiva, intuitiva y de fácil manejo para el usuario, por lo que se trabaja con el *software* Qt Designer versión 4.14.1 para llevar a cabo su diseño (Qt, 2021).

2.6.1 Diseño de interfaz y botones para las pantallas

Para que el usuario pueda navegar en la aplicación se considera realizar 6 pantallas, se debe considerar el dimensionamiento del tamaño de la pantalla, como la pantalla que se utiliza tiene un tamaño de 480 x 320 pixeles, se la redimensiona a estos valores. Para desarrollar la interfaz, primero se realiza los fondos de pantalla y las imágenes para los botones, para ello se utilizó Photoshop versión 22.4.1. En la figura 21-2 se observa el diseño de la interfaz de la pantalla principal del prototipo (Adobe, 2021).



Figura 21-2: Diseño de la interfaz de la pantalla principal.
Realizado por: Berrones A, Crespata C; 2021.

Gracias a Qt Designer se puede colocar botones, cuadros de texto y pantallas para enlazarlos desde Python mediante la librería PYQT5. Una vez finalizado el diseño se procede a guardar y crear un archivo .ui el cual es ejecutado mediante código en Python. En la figura 22-2 se aprecia un ejemplo de un botón de los 8 diseñados, en este caso se trata del botón que permite realizar la captura de la imagen.

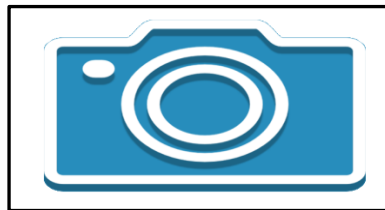


Figura 22-2: Botón tomar imagen.
Realizado por: Berrones A, Crespata C; 2021.

En la figura 23-2 se presenta la pantalla principal, luego de que el usuario ejecuta la aplicación, donde puede apreciar que tiene como encabezado el nombre de la institución, la facultad, la escuela y el tema del trabajo de titulación. En la pantalla principal se encuentran incluidos dos botones (1 y 2). Donde al ejecutarse el botón 1, le presenta al usuario las imágenes de las señaléticas que se encuentran en el Centro Histórico de la ciudad de Riobamba, que a su vez, estas se encuentran almacenadas en la memoria del prototipo, en la figura 24-2 se aprecia la interfaz de esta pantalla. El botón 2 al ser presionado, se despliega la segunda pantalla donde el usuario puede realizar la captura de imagen.



Figura 23-2: Interfaz de la pantalla principal.
Realizado por: Berrones A, Crespata C; 2021.



Figura 24-2: Interfaz de la pantalla Base de datos de imágenes.
Realizado por: Berrones A, Crespata C; 2021.

En la figura 25-2 se presenta la pantalla donde se activa la cámara para realizar las capturas de las imágenes para aplicar el reconocimiento del pictograma. Esta pantalla incluye 3 botones (3, 4, 5), donde al ejecutar el botón 3 realiza la captura de la escena, el botón 4 permite activar nuevamente la cámara en caso de que el usuario haya realizado erróneamente una captura. El botón 5 le permite al usuario regresar a la pantalla principal.



Figura 25-2: Interfaz de la pantalla captura de imágenes.
Realizado por: Berrones A, Crespata C; 2021.

Cuando el prototipo reconoce la señalética se presenta la interfaz de la pantalla sobre el menú de rutas, la cual se observa en la figura 26-2. Esta pantalla incluye 3 botones (6, 7, 8), el botón 6 permite conocer la ruta a tomar para llegar al destino, el botón 7 le presenta al usuario su posición actual en el mapa mientras se va dirigiendo a su lugar de interés, mientras el botón 8 realiza la acción de regresar a la pantalla de captura de imagen. Además cuenta con dos cuadros de texto (A y B) donde en A se presenta los datos (latitud, longitud, dirección) extraídos de la tabla de información de Excel que trata sobre el lugar que reconoce el prototipo y en B se presenta la información de la posición donde está ubicado el usuario, estos datos son brindados por el GPS.

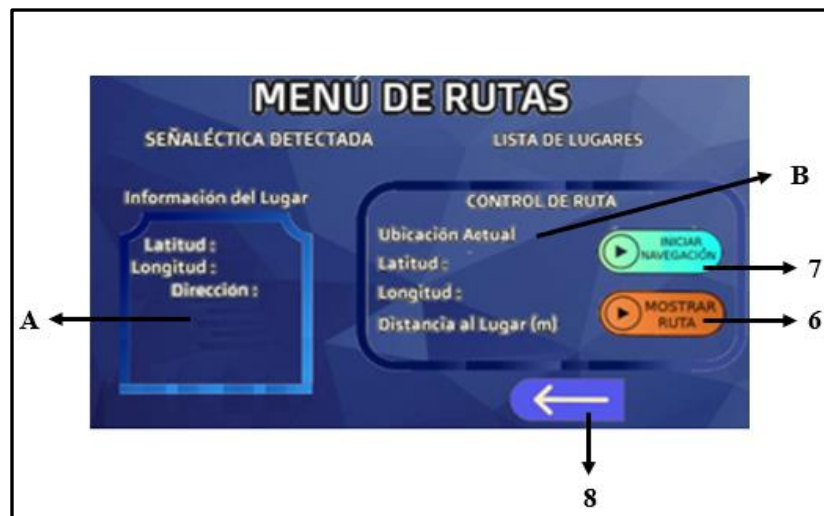


Figura 26-2: Interfaz de la pantalla Menú de rutas.
Realizado por: Berrones A, Crespata C; 2021.

Tanto como para presentar la ruta o como para estar en la navegación se presenta la misma interfaz de pantalla, como por ejemplo, en la figura 27-2 se visualiza la pantalla sobre la ruta, donde consta de dos botones (9, 10), el botón 9 permite regresar a la pantalla de menú de rutas y el botón 10 inicia la navegación. Al finalizar el recorrido, el usuario puede apreciar en la pantalla información sobre, historia, hitos importantes y una imagen del lugar de interés, en la figura 28-2 se presenta un ejemplo de esta interfaz.



Figura 27-2: Interfaz de la pantalla Ruta.
 Realizado por: Berrones A, Crespata C; 2021.



Figura 28-2: Interfaz de la pantalla Información.
 Realizado por: Berrones A, Crespata C; 2021.

CAPÍTULO III

3 VALIDACIÓN DEL PROTOTIPO

En este capítulo se describe las pruebas realizadas al prototipo, las cuales se encuentran basadas en el cumplimiento de los objetivos planteados.

3.1 Descripción de las pruebas realizadas

Las pruebas que se evalúan constan de tres etapas, la primera la adquisición de imágenes en la cual se refiere a la iluminación y distancia, la segunda validación del sistema de información turística donde se comprueba la información presentada al usuario y las rutas hacia los diferentes destinos, finalmente se pondrá a prueba el tiempo de respuesta del prototipo frente al reconocimiento de la señalética cual es el tiempo máximo y mínimo que se tarda en cada atractivo turístico.

3.2 Prueba de la etapa de adquisición de imágenes

En esta etapa se busca medir la capacidad de respuesta del prototipo frente a variaciones de iluminación y distancia de captura, para de esta forma evaluar el reconocimiento de las señaléticas y la eficiencia que posee en cada una de las pruebas.

3.2.1 *Pruebas a diferentes niveles de iluminación*

El objetivo de esta prueba es medir la capacidad del prototipo para detectar la señalética a las variaciones de iluminación del entorno provocados por el transcurso del día y ambientes climáticos, para lo cual se realiza una captura de 60 imágenes para cada uno de los diferentes niveles de iluminación encontrados, con esto se pone a prueba el dispositivo frente a estas variables, como equipos se necesita el prototipo y la aplicación del luxómetro.

Para clasificar los niveles de iluminación que se pueden encontrar en el transcurso del día se toma como referencia la tabla 1-3 publicada por la Oficina Técnica para la Protección de la Calidad del Cielo (OTPC) del instituto astrofísico de Canarias.

Tabla 1-3: Niveles de iluminación aproximados.

FUENTE	ILUMINACIÓN (lux)
Día Soleado	103000
Día parcialmente soleado	50000
Día nublado	1000-40000

Fuente:(Ponce, 2017).

Realizado por: Berrones A, Crespata C; 2021.

Para las pruebas en este apartado se las realiza en diferentes días en los cuales se tuvo diferentes niveles de iluminación, con la aplicación de luxómetro se ha realizado mediciones la cual se puede observar en la figura 1-3 en los diferentes horarios del día para poder tener una referencia y poder clasificarlos según si el día está despejado, nubloso u obscuro con lo cual se ha logrado clasificar de acuerdo a la tabla 2-3.

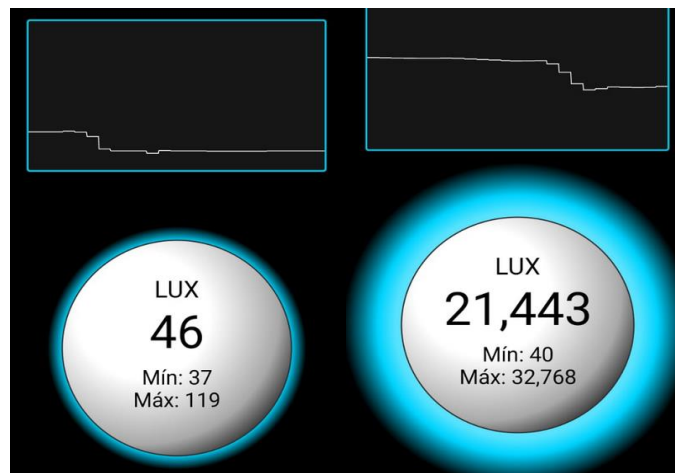


Figura 1-3: Medición de luxes para la tabla 2-3.

Realizado por: Berrones A, Crespata C; 2021.

Tabla 2-3: Niveles de iluminación a diferentes horas del día.

NO	PARTE DEL DÍA	HORA	LUXES MEDIDOS
1	SOLEADO	10:00	85000
2	SOLEADO	12:00	120000
3	NUBLOSO	14: 00	45000
4	NUBLOSO	16:00	27000
5	OSCURO	18:00	220
6	OSCURO	20:00	46

Realizado por: Berrones A, Crespata C; 2021

Es necesario mencionar que no siempre en un horario determinado como a las 10 am. siempre esté despejado, puede darse ocasiones en los que este nublado, pero esto no tiene por qué interferir con las pruebas, pues por ese motivo se tiene la medición de los luxes con los cuales se determina la medida para las diferentes estados que se pueden dar en un día, puesto que puede haber días que se encuentren totalmente nublados así como otros en los cuales se tiene un sol radiante.

A continuación se describe el procedimiento utilizado para la captura de imágenes.

1. Dirigirse a la señalética a evaluar
2. Realizar la medición de luxes del entorno para comparar y clasificar de acuerdo a la tabla 4-3.
3. Realizar la captura de la imagen
4. Evaluación del reconocimiento de la imagen

En la figura 2-3 se aprecia un ejemplo de la captura realizada de la señalética del parque de Guayaquil, en un rango de iluminación perteneciente a un entorno nublado.



Figura 2-3: Captura a 45000 lx.
Realizado por: Berrones A, Crespata C; 2021.

En la figura 3-3 se observa evidencia de la captura realizada de la señalética de la Iglesia de San Antonio a un nivel de iluminación perteneciente a un ambiente soleado.



Figura 3-3: Captura a 120000 lx
Realizado por: Berrones A, Crespata C; 2021.

En la figura 4-3 se aprecia la señalética de la iglesia de San Antonio con un ambiente nublado, estas capturas son realizadas por la cámara y almacenadas en la base de datos que posee el dispositivo, con lo cual procede al reconocimiento de la señalética.



Figura 4-3: Captura a 120 lx.
Realizado por: Berrones A, Crespata C; 2021.

Posteriormente se procede a realizar una tabla con 60 muestras analizadas para cada nivel de iluminación además de un cuadro resumen de cada prueba y un gráfico de barras que representen estos datos obtenidos, cada detección correcta está representada por el número 1 y una detección incorrecta o no detección por el número 0.

En la tabla 3-3 se aprecia los resultados que se obtiene por cada prueba y en el diferente tipo de iluminación en el que se realiza, para finalmente obtener el nivel de eficiencia del prototipo, para lo cual se utiliza la ecuación 1-3.

$$Eficiencia = \frac{Identificaciones\ correctas * 100}{Número\ de\ imágenes\ capturadas} \quad \text{Ecuación 1-3}$$

Tabla 3-3: Pruebas al algoritmo con diferentes tipos de iluminación.

	(Mañana) (nivel de luxes entre 85000- 120000)	(Nublado) (nivel de luxes entre 45000 - 2750)	(Noche) (nivel de luxes entre 220-46)		(Mañana) (nivel de luxes entre 85000- 120000)	(Nublado) (nivel de luxes entre 45000 - 2750)	(Noche) (nivel de luxes entre 220-46)
No. Prueba	SIFT Y FLANN	SIFT Y FLANN	SIFT Y FLANN	No. Prueba	SIFT Y FLANN	SIFT Y FLANN	SIFT Y FLANN
1	1	1	0	32	1	1	1
2	1	1	1	33	1	1	0
3	1	1	1	34	1	1	0
4	1	1	1	35	1	1	0
5	1	0	0	36	0	1	0
6	1	1	1	37	1	0	0
7	1	0	0	38	1	1	0
8	1	1	0	39	1	1	0
9	1	1	0	40	1	1	0
10	1	1	0	41	1	1	0
11	0	1	0	42	1	1	0
12	1	1	1	43	1	1	0
13	1	1	0	44	1	0	0
14	1	1	0	45	1	1	0
15	1	0	0	46	1	1	0
16	1	1	0	47	1	1	0
17	1	1	0	48	1	1	0
18	1	1	1	49	1	1	1
19	1	1	0	50	1	1	0
20	1	0	0	51	1	1	1
21	1	1	0	52	1	1	0
22	1	1	1	53	1	0	0
23	1	1	1	54	0	1	0
24	1	1	0	55	1	1	1
25	1	1	0	56	1	1	1
26	1	0	0	57	1	1	0
27	1	1	1	58	1	1	0
28	1	1	0	59	1	1	0
29	1	1	1	60	1	1	1
30	1	1	0				
31	1	1	0	Eficiencia	95%	86.67%	26.67%

Realizado por: Berrones A, Crespata C; 2021.

Tabla 4-3 : Total de pruebas de identificación a diferentes tipos de iluminación.

Estado del día	Mañana luxes entre (85000-120000)	Nublado Luxes entre (2750-45000)	Noche Luxes entre (46-220)	Total
Identificaciones correctas	57	52	16	125
Identificaciones incorrectas	3	8	44	55
Total	60	60	60	180

Realizado por: Berrones A, Crespata C; 2021.

De los datos recogidos en la tabla 4-3 se obtiene el gráfico 1-3:

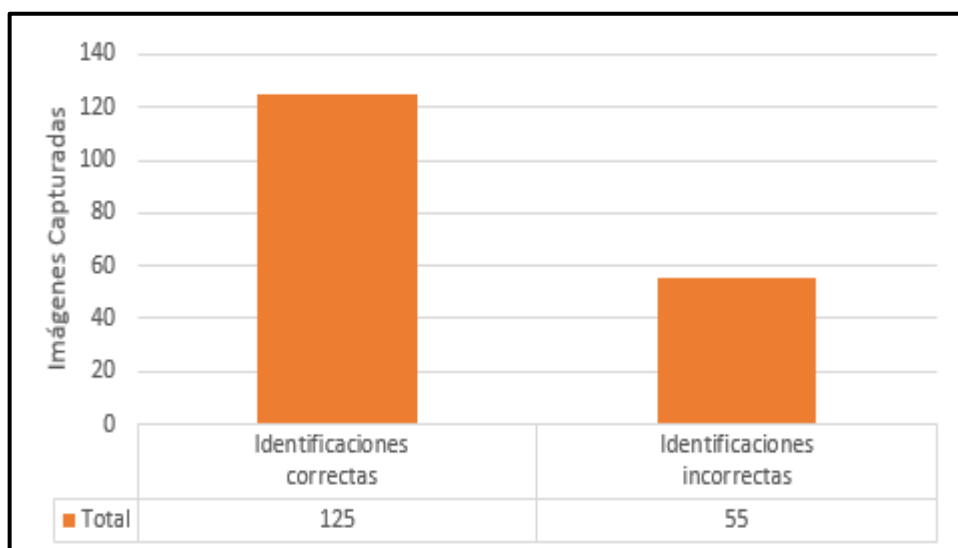


Gráfico 1-3: Total de imágenes analizadas.

Realizado por: Berrones A, Crespata C; 2021.

En general a esta prueba con el número total de identificaciones correctas e incorrectas en los diferentes niveles de iluminación, se hace uso de la ecuación 1-3 con la cual se obtiene una eficiencia del 69% del prototipo, esto debido a las pruebas realizadas en la noche que por el poco nivel de iluminación que existe hace que su eficiencia no sea mayor.

Por lo cual se concluye entorno a los resultados obtenidos por las pruebas, que el prototipo puede trabajar en el transcurso del día sin inconvenientes con una eficiencia del 86.67% sin embargo para tener una eficiencia del 95% necesita una iluminación en un rango de 85000 a 120000 luxes.

3.2.2 Pruebas a diferentes distancias del prototipo a la señalética

En este apartado se realizan las pruebas en relación a la distancia en la que el prototipo es capaz de trabajar, con el objetivo de conocer el rango entre la señalética y la lente de la cámara en la cual se tenga un alto porcentaje de eficiencia de captura. Para lo cual se procede una evaluación de 20 muestras en rangos de distancia de 3m, 2m, 1m y menores a 1m. Como equipos necesarios se requiere el prototipo y una cinta métrica.

En primer lugar para conocer la distancia de la señalética al lente de la cámara, se parte de la figura 5-3, donde se observa las diferentes variables a utilizar. De acuerdo a esta figura se procede a realizar los cálculos correspondientes para los diferentes rangos de distancia.

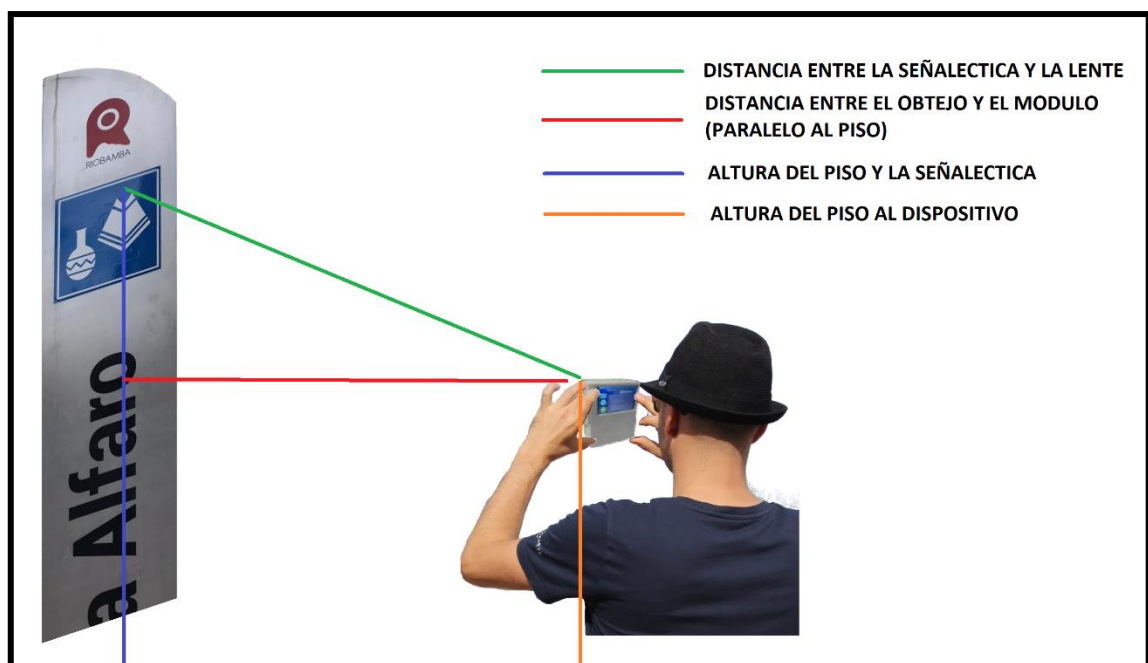


Figura 5-3: Representación de la distancia entre el prototipo y la señalética.

Realizado por: Berrones A, Crespata C; 2021.

Para conocer la distancia de la señalética y la cámara se aplica las siguientes formulas:

$$DISTANCIA = \sqrt{Dm^2 + AL^2} \text{ Ecuación 2-3}$$

Siendo:

Distancia: Longitud entre la señalética y la lente.

Dm: Distancia entre el objeto y el módulo paralelo al piso.

AL: Altura entre la señalética y el dispositivo.

Para calcular la altura AL que es la diferencia entre la altura del piso a la señal y la altura entre el piso y el dispositivo se aplica la siguiente ecuación:

$$AL = AS - AD \text{ Ecuación 3-3}$$

Donde :

AS: Altura del piso a la señalética.

AD: Altura del piso al dispositivo.

El resultado de las pruebas aplicado a las distancias que el usuario puede estar frente a la señalética se detalla en la tabla 5-3.

Tabla 5-3: Diferentes distancias entre la señalética y el usuario.

No	AS	AD	AL	Dm	Distancia
1	2	1.5	0.5	1.2	1.30
2	2	1.5	0.5	1.4	1.49
3	2	1.5	0.5	1.6	1.68
4	2	1.5	0.5	1.8	1.87
5	2	1.5	0.5	2	2.06
6	2	1.5	0.5	2.2	2.26
7	2	1.5	0.5	2.4	2.45
8	2	1.5	0.5	2.6	2.65
9	2	1.5	0.5	2.8	2.84
10	2	1.5	0.5	3	3.04

Realizado por: Berrones A, Crespata C; 2021.

A continuación, se procede a realizar los cálculos de distancia para el rango establecido de 3m, 2m, 1m y menores a 1m. Además para determinar la eficiencia en esta prueba sobre el dispositivo se hace uso de la ecuación 1-3.

Como parte del procedimiento de esta prueba que se lleva a cabo se tiene los siguientes pasos.

1. Dirigirse a la señalética a evaluar.
2. Realizar las mediciones y cálculos de las diferentes variables involucradas.
3. Realizar captura de la imagen.
4. Comprobar el rango de detección con mayor eficiencia.

Tabla 6-3: Prueba de distancia de detección de algoritmo de visión con $D_m= 3m$.

No. Prueba	AS	AD	AL	D_m	Distancia	SIFT Y FLANN
1	2	1.4	0.6	3	3.06	0
2	2	1.5	0.5	3	3.04	0
3	2	1.5	0.5	3	3.04	0
4	2	1.5	0.5	3	3.04	0
5	2	1.3	0.7	3	3.08	0
6	2	1.4	0.6	3	3.06	1
7	2	1.5	0.5	3	3.04	0
8	2	1.4	0.6	3	3.06	0
9	2	1.4	0.6	3	3.06	0
10	2	1.5	0.5	3	3.04	0
11	2	1.5	0.5	3	3.04	0
12	2	1.5	0.5	3	3.04	0
13	2	1.5	0.5	3	3.04	0
14	2	1.4	0.6	3	3.06	0
15	2	1.5	0.5	3	3.04	0
16	2	1.5	0.5	3	3.04	0
17	2	1.4	0.6	3	3.06	0
18	2	1.4	0.6	3	3.06	1
19	2	1.5	0.5	3	3.04	0
20	2	1.5	0.5	3	3.04	0
EFICIENCIA						10%

Realizado por: Berrones A, Crespata C; 2021.

Realizado los cálculos para una distancia de 3m y con una muestra de 20 pruebas se obtiene una eficiencia del 10%, además se representa las identificaciones obtenidas en el gráfico 2-3.

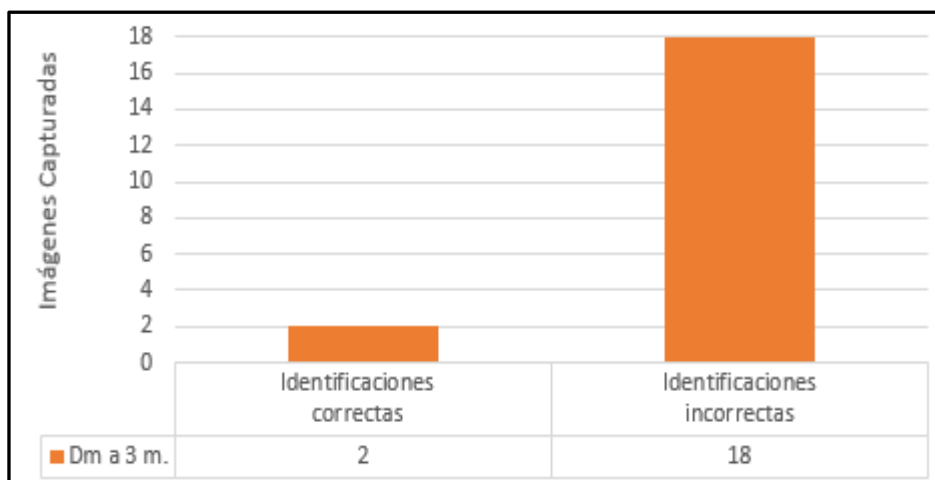


Gráfico 2-3: Imágenes analizadas a D_m igual a 3 metros.

Realizado por: Berrones A, Crespata C; 2021.

A continuación se realiza el cálculo para una distancia de 2m.

Tabla 7-3: Prueba de distancia de detección de algoritmo de visión con $D_m=2m$.

No. Prueba	AS	AD	AL	D_m	Distancia	SIFT Y FLANN
1	2	1.4	0.6	2	2.09	0
2	2	1.5	0.5	2	2.06	0
3	2	1.5	0.5	2	2.06	1
4	2	1.5	0.5	2	2.06	0
5	2	1.3	0.7	2	2.12	0
6	2	1.4	0.6	2	2.09	0
7	2	1.5	0.5	2	2.06	1
8	2	1.4	0.6	2	2.09	1
9	2	1.4	0.6	2	2.09	0
10	2	1.5	0.5	2	2.06	0
11	2	1.5	0.5	2	2.06	0
12	2	1.5	0.5	2	2.06	0
13	2	1.5	0.5	2	2.06	0
14	2	1.4	0.6	2	2.09	1
15	2	1.5	0.5	2	2.06	0
16	2	1.5	0.5	2	2.06	0
17	2	1.4	0.6	2	2.09	0
18	2	1.4	0.6	2	2.09	1
19	2	1.5	0.5	2	2.06	0
20	2	1.5	0.5	2	2.06	0
EFICIENCIA						25%

Realizado por: Berrones A, Crespata C; 2021.

Realizado los cálculos para una distancia de 2m se procede evaluar la detección para posteriormente calcular la eficiencia, haciendo uso de la ecuación 1-3 se tiene un valor del 25%, además se representa las identificaciones obtenidas en el gráfico 3-3.

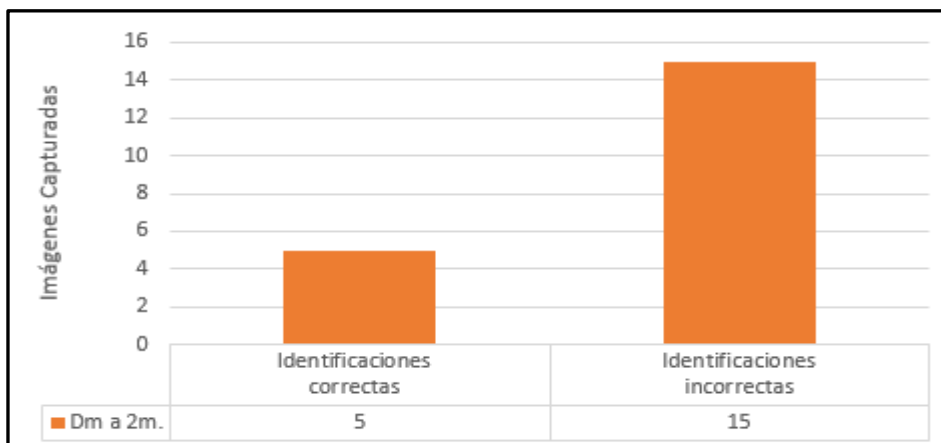


Gráfico 3-3: Imágenes analizadas a D_m igual a 2 metros.

Realizado por: Berrones A, Crespata C; 2021.

Tabla 8-3: Prueba de distancia de detección de algoritmo de visión con $D_m = 1m$.

No. Prueba	AS	AD	AL	D_m	Distancia	SIFT Y FLANN
1	2	1.4	0.6	1	1.17	1
2	2	1.5	0.5	1	1.12	1
3	2	1.5	0.5	1	1.12	1
4	2	1.5	0.5	1	1.12	1
5	2	1.3	0.7	1	1.22	1
6	2	1.4	0.6	1	1.17	0
7	2	1.5	0.5	1	1.12	1
8	2	1.4	0.6	1	1.17	1
9	2	1.4	0.6	1	1.17	1
10	2	1.5	0.5	1	1.12	1
11	2	1.5	0.5	1	1.12	1
12	2	1.5	0.5	1	1.12	0
13	2	1.5	0.5	1	1.12	0
14	2	1.4	0.6	1	1.17	1
15	2	1.5	0.5	1	1.12	0
16	2	1.5	0.5	1	1.12	1
17	2	1.4	0.6	1	1.17	1
18	2	1.4	0.6	1	1.17	1
19	2	1.5	0.5	1	1.12	1
20	2	1.5	0.5	1	1.12	1
EFICIENCIA						80%

Realizado por: Berrones A, Crespata C; 2021.

En la tabla 8-3 se observa los datos obtenidos para una distancia de 1m, de igual forma se obtiene una eficiencia del 80%, mientras que en el gráfico 4-3 se representa el número de detecciones correspondiente a esta distancia.

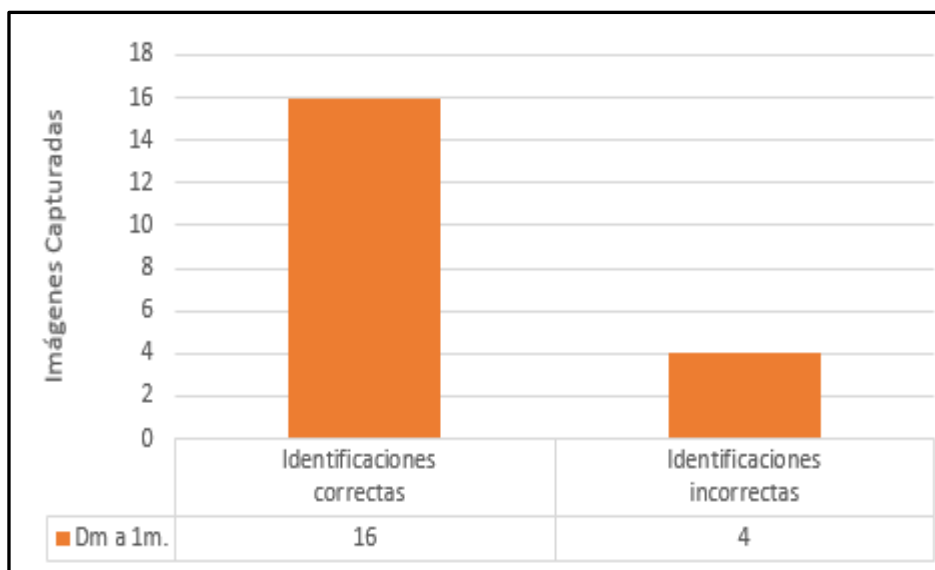


Gráfico 4-3: Imágenes analizadas a D_m igual a 1 metro.

Realizado por: Berrones A, Crespata C; 2021.

Finalmente se realiza el cálculo para distancias menores a 1m.

Tabla 9-3: Prueba de distancia de detección de algoritmo de visión con $D_m < 1m$.

No. Prueba	AS	AD	AL	Dm	Distancia	SIFT Y FLANN
1	2	1.4	0.6	0.8	1.00	1
2	2	1.5	0.5	0.8	0.94	1
3	2	1.5	0.5	0.8	0.94	1
4	2	1.5	0.5	0.8	0.94	1
5	2	1.3	0.7	0.8	1.06	1
6	2	1.4	0.6	0.8	1.00	1
7	2	1.5	0.5	0.8	0.94	1
8	2	1.4	0.6	0.8	1.00	1
9	2	1.4	0.6	0.8	1.00	1
10	2	1.5	0.5	0.8	0.94	1
11	2	1.5	0.5	0.5	0.71	0
12	2	1.5	0.5	0.5	0.71	1
13	2	1.5	0.5	0.5	0.71	1
14	2	1.4	0.6	0.5	0.78	1
15	2	1.5	0.5	0.5	0.71	1
16	2	1.5	0.5	0.5	0.71	0
17	2	1.4	0.6	0.5	0.78	1
18	2	1.4	0.6	0.5	0.78	1
19	2	1.5	0.5	0.5	0.71	1
20	2	1.5	0.5	0.5	0.71	1
EFICIENCIA						90%

Realizado por: Berrones A, Crespata C; 2021.

En la tabla 9-3 se observa los datos obtenidos para una distancia menor a 1m, en este caso se observa una eficiencia del 90%, mientras que en el gráfico 5-3 se representa el número de detecciones correspondiente a esta distancia.

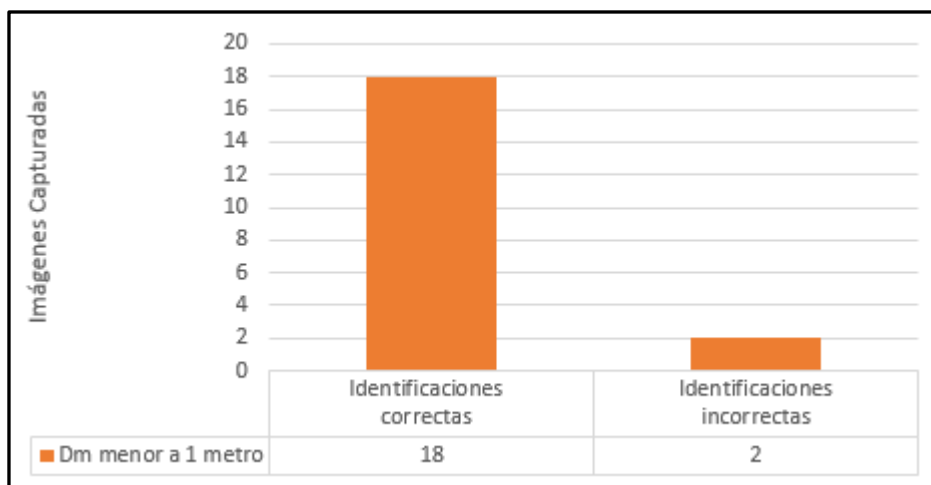


Gráfico 5-3: Imágenes analizadas a D_m menor a 1 metro.

Realizado por: Berrones A, Crespata C; 2021.

En general a esta prueba con el número total de 80 imágenes tomadas y con las diferentes identificaciones correctas e incorrectas en los distintos rangos de distancias establecidos, se hace uso de la ecuación 1-3 con la cual se obtiene una eficiencia del 51.25 % del prototipo a esta prueba.

Por lo cual se concluye entorno a los resultados obtenidos por las pruebas, que el prototipo necesita el rango de adquisición de imagen sea de 1m o menor distancia, para tener una eficiencia del 90%.

3.3 Pruebas del trazado de rutas y de la información del lugar de destino.

Este apartado se divide en 2 secciones, donde la primera evalúa el trazado de la ruta del prototipo con el trazado de la ruta con una aplicación similar. La segunda sección realiza las pruebas sobre la fiabilidad de la información obtenida que el prototipo arroja en su pantalla sobre el lugar de destino al que llegó el usuario.

3.3.1 Trazado de rutas

Como objetivo de esta prueba se tiene determinar si el prototipo puede realizar el trazado correcto de rutas hacia los diferentes atractivos comparándolo con una aplicación de rutas en este caso Waze en su versión 2.0.

El procedimiento a seguir para realizar la prueba es el siguiente:

1. Ambiente controlado (Iluminación y distancia adecuada).
2. Visitar diferentes atractivos turísticos y verificar mediante observación que el destino de la ruta sea el mismo que al comparar con la aplicación Waze.
3. Observar que la ruta trazada me lleve al sitio deseado.

Las rutas para cada sitio y circuito turístico del Centro Histórico de Riobamba, se encuentran determinadas para cumplir con un recorrido definido por la primera señalética que se encuentre, ya sea si esta es parte de un circuito turístico o en si es un destino a donde se desea llegar, en la figura 6-3 se aprecia una muestra de la prueba realizada con destino hacia el parque La Libertad en el prototipo.



Figura 6-3: Ruta para el destino del parque La Libertad con el prototipo.
Realizado por: Berrones A, Crespata C; 2021.

En la figura 7-3 se presenta el trazado de rutas con la aplicación Waze.

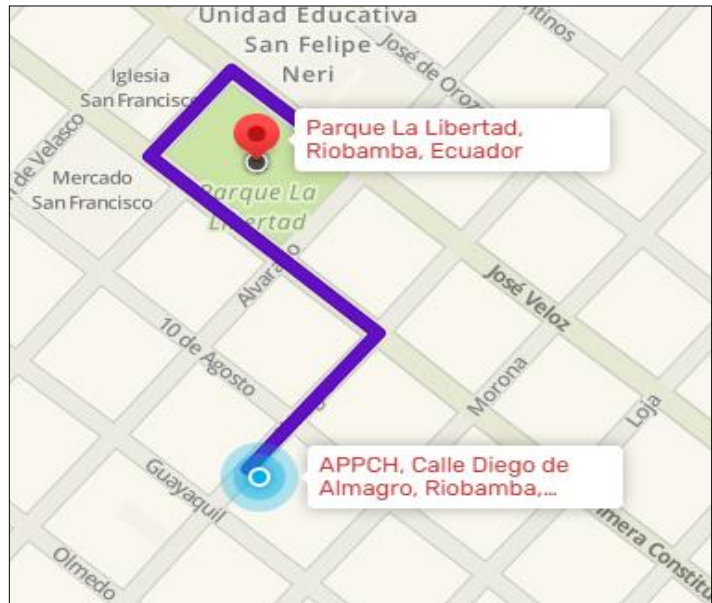


Figura 7-3: Ruta con aplicación del parque La Libertad con Waze.
Realizado por: Berrones A, Crespata C; 2021.

Mediante observación se ha comprobado que de las pruebas realizadas el 100% son correctas con esto se concluye que el prototipo es fiable en el trazado de rutas y es competitivo frente aplicaciones similares.

3.3.2 Evaluación de la información obtenida

El objetivo de esta prueba es medir la capacidad que tiene el prototipo para brindar la información y comprobar en la pantalla si esta es congruente con el destino visitado.

El procedimiento a seguir para realizar la prueba es el siguiente:

1. Ambiente controlado (Iluminación y distancia adecuada).
2. Visitar los 6 diferentes atractivos turísticos ejecutando la aplicación y trazando la ruta desde la señalética y con el traslado del prototipo hasta el lugar de destino.
3. Observar que la información sea coherente con el lugar de destino.

En esta etapa, una vez superado las pruebas de adquisición de imágenes, se contrasta la información que presenta el dispositivo con la información de cada sitio turístico que sea la perteneciente a cada señalética. En la tabla 10-3 se encuentran las señaléticas cotejadas para esta prueba.

Tabla 10-3: Pruebas de la información obtenida por el dispositivo.

Señalética	Información
Plaza de toros	Correcta
Estación del tren	Correcta
Iglesia de San Antonio	Correcta
Parque 21 de abril	Correcta
Parque Guayaquil	Correcta
Parque Sucre	Correcta
Correos del Ecuador	Correcta

Realizado por: Berrones A, Crespata C; 2021.

De acuerdo a la tabla 10-3 del total de las 7 señaléticas presentadas todas muestran la información correcta en pantalla, teniendo un 100% de efectividad en la prueba de la información turística presentada. En la figura 8-3 se aprecia un ejemplo de la prueba realizada y la información turística del Parque Sucre que se presenta en pantalla.

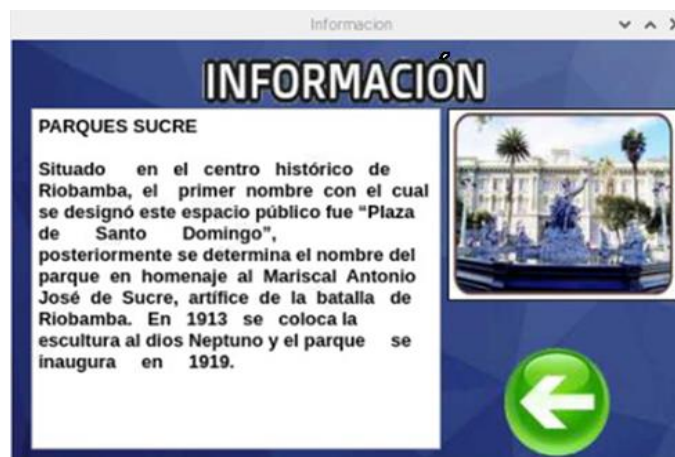


Figura 8-3: Información sobre el Parque Sucre.

Realizado por: Berrones A, Crespata C; 2021.

Mediante observación se comprobó que del total de las pruebas realizadas el 100% son correctas dando como conclusión que el prototipo es altamente confiable entorno a la información presentada al usuario.

3.4 Tiempo de respuesta

Como objetivo de los tiempos de respuesta es la de evaluar el tiempo de reconocimiento del prototipo frente a las distintas señaléticas. Para las pruebas de tiempo de procesamiento se ocupa la librería time, la que mediante la función time.time() puede calcular el tiempo de ejecución del sistema, es decir se coloca una flag (bandera) al inicio y otro al final la diferencia de tiempo de los 2 se tendrá el tiempo de procesamiento. Cabe recalcar que la unidad de tiempo que proporciona la librería está dada en segundos.

```
import time
t = time.time()
# algoritmo
tiempo = time.time() - t
print(tiempo)
```

Figura 9-3: Código para obtener el tiempo de respuesta.
Realizado por: Berrones A, Crespata C; 2021.

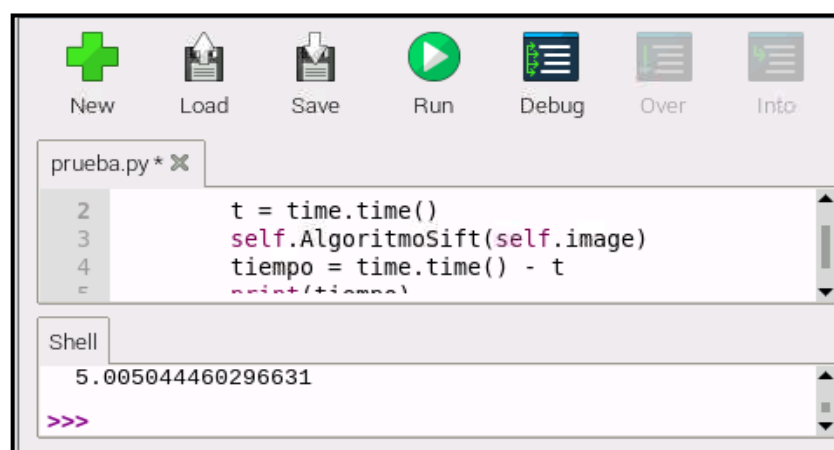


Figura 10-3: Ejecución del código para obtener el tiempo de respuesta.
Realizado por: Berrones A, Crespata C; 2021.

El tiempo de respuesta se lo ha determinado mediante el promedio del tiempo que le toma realizar el procesamiento al prototipo bajo condiciones normales con intervalos de 10 a 20 segundos, realizando 60 pruebas de campo sobre los diferentes pictogramas en la señalética.

El procedimiento para realizar esta prueba es el siguiente:

1. Tomar 10 pruebas por cada clasificación de la señalética turística.
2. Durante dos horas.
3. Analizar el tiempo máximo y mínimo de detección que posee el dispositivo.

3.4.1 Pruebas de campo con las señaléticas del entorno

Las pruebas de campo se presentan tomando el tiempo invertido que le toma al dispositivo procesar la imagen adquirida de los diferentes pictogramas de la señalética que se encuentra en el Centro Histórico de Riobamba para poder estimar un promedio sobre la rapidez del prototipo.

Tabla 11-3: Tiempo para la identificación de iglesias.

Cantidad	Señalética	Tiempo Invertido (Segundos)
1	IGLESIA	5.414
2		5.766
3		5.904
4		5.968
5		5.453
6		5.513
7		5.197
8		5.789
9		5.61
10		5.449
Tiempo promedio en iglesias		5.6063

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 11-3 se observa el promedio del tiempo que le toma al prototipo realizar el procesamiento para el reconocimiento del pictograma de iglesias con 10 pruebas recogidas que da como resultado un tiempo de 5.6063 segundos.

Tabla 12-3: Tiempo para la identificación de museos.

Cantidad	Señalética	Tiempo Invertido (Segundos)
1	MUSEO	5.818
2		5.771
3		5.25
4		5.664
5		5.573
6		5.528
7		5.654
8		5.218
9		5.681
10		5.652
Tiempo promedio en museo		5.5809

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 12-3 se observa el promedio del tiempo que le toma al prototipo realizar el procesamiento para el reconocimiento del pictograma de museos con 10 pruebas recogidas que da como resultado un tiempo de 5.5809 segundos.

Tabla 13-3: Tiempo para la identificación de parques.

Cantidad	Señalética	Tiempo Invertido (Segundos)
1	PARQUE	5.774
2		5.465
3		5.951
4		5.795
5		5.066
6		5.282
7		5.082
8		5.951
9		5.45
10		5.502
Tiempo promedio en parques		5.5318

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 13-3 se observa el promedio del tiempo que le toma al prototipo realizar el procesamiento para el reconocimiento del pictograma de parques con 10 pruebas recogidas que da como resultado un tiempo de 5.5318 segundos.

Tabla 14-3: Tiempo para la identificación de mercados.

Cantidad	Señalética	Tiempo Invertido (Segundos)
1	MERCADO	5.11
2		5.866
3		5.866
4		5.059
5		5.732
6		5.055
7		5.378
8		5.769
9		5.34
10		5.353
Tiempo promedio en mercados		5.4528

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 14-3 se observa el promedio del tiempo que le toma al prototipo realizar el procesamiento para el reconocimiento del pictograma de mercados con 10 pruebas recogidas que da como resultado un tiempo de 5.4528 segundos.

Tabla 15-3: Tiempo para la identificación de Patrimonios Arquitectónicos.

Cantidad	Señalética	Tiempo Invertido (Segundos)
1	PATRIMONIOS ARQUITECTÓNICOS	5.701
2		5.118
3		5.149
4		5.774
5		5.998
6		5.539
7		5.286
8		5.119
9		5.978
10		5.186
Tiempo promedio en Patrimonios Arquitectónicos		5.4848

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 15-3 se observa el promedio del tiempo que le toma al prototipo realizar el procesamiento para el reconocimiento del pictograma de Patrimonios Arquitectónicos con 10 pruebas recogidas que da como resultado un tiempo de 5.4848 segundos.

Tabla 16-3: Tiempo para la identificación de la Plaza de Toros.

Cantidad	Señalética	Tiempo Invertido (Segundos)
1	PLAZAS	5.967
2		5.035
3		5.415
4		5.188
5		5.956
6		5.65
7		5.491
8		5.583
9		5.725
10		5.414
Tiempo promedio en Plazas		5.5424

Realizado por: Berrones A, Crespata C; 2021.

Por medio de la tabla 16-3 se observa el promedio del tiempo que le toma al prototipo realizar el procesamiento para el reconocimiento del pictograma de la Plaza de Toros con 10 pruebas recogidas que da como resultado un tiempo de 5.6063 segundos.

Finalmente, en el gráfico 6-3 se presenta el promedio calculado de todos los promedios de datos recogidos de cada uno de los pictogramas que le toma procesar al prototipo desde la adquisición de la imagen, hasta cuando presenta el resultado, que en este caso es la ruta, el cual arroja un promedio total de 5.5332 segundos.

El tiempo de reconocimiento está en un rango mínimo de 5.45s y máximo de 5.6s esto permite concluir que el prototipo es veloz en el reconocimiento de señalética debido a que posee un tiempo de respuesta máximo de 5.6s, en relación a las 70 pruebas realizadas.

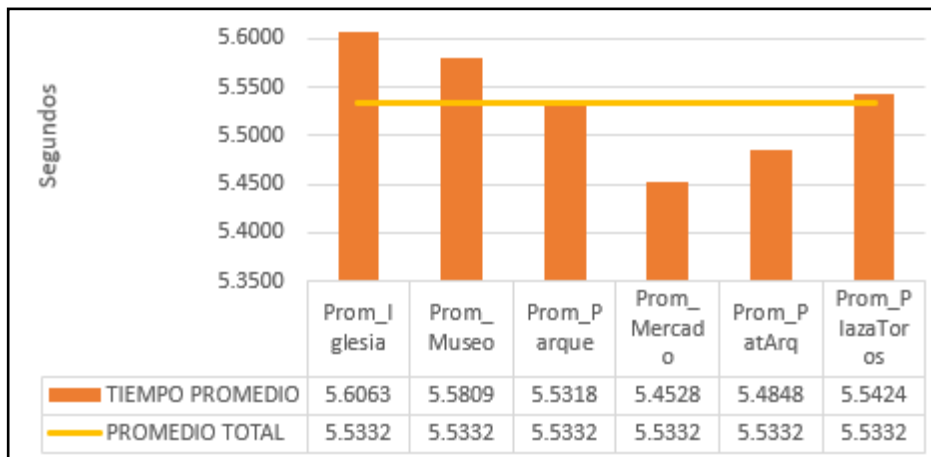


Gráfico 6-3: Tiempo promedio del procesamiento de detección del pictograma.
Realizado por: Berrones A, Crespata C; 2021.

3.5 Análisis de autonomía del dispositivo y batería

Debido a la demanda de energía que necesita el prototipo para trabajar, se requiere una batería de alimentación que le permita funcionar al prototipo correctamente. Para la selección de la batería se toma ciertos parámetros como son la autonomía y el tamaño de esta para ellos se realizó el siguiente análisis de consumo de corriente del prototipo.

Tabla 17-3: Consumo de Stand-by o reposo.

No	Corriente consumida
1	0.6
2	0.61
3	0.59
4	0.6
5	0.62
Promedio	0.604

Realizado por: Berrones A, Crespata C; 2021.

Tabla 18-3: Consumo de corriente en funcionamiento.

No	Elemento del modulo	Consumo en corriente (A)
1	Raspberry	0.62
2	Cámara	0.256
3	Pantalla	0.130
4	GPS Neo 6m	0.067
TOTAL		1.073 [A]

Realizado por: Berrones A, Crespata C; 2021.

Para conocer la autonomía que tendría el prototipo con diferentes baterías se aplica lo siguiente:
 Se calcula la potencia instantánea consumida por el equipo mediante la ecuación 4-3.

$$P_p = I_p * V \quad \text{Ecuación 4-3}$$

$$P_p = (1.073 A) * (5 v)$$

$$P_p = 5.365 W$$

Donde:

P_p : Potencia del prototipo.

I_p : Corriente consumida por el prototipo.

V : Voltaje consumido por el prototipo.

Para el consumo de la batería se utiliza la $TA = \text{Energía de la Batería} / \text{Potencia del prototipo}$ y se realizan los cálculos para 3 tipos de batería como se puede observar en la tabla 19-3.

$$P_B = I_B * V \quad \text{Ecuación 5-3}$$

La ecuación 5-3 permite determinar el tiempo de autonomía del dispositivo, cuyos valores calculados se presentan en la table 19-3.

$$TA = \frac{\text{Energía de la Batería}}{\text{Potencia del prototipo}} \quad \text{Ecuación 6-3}$$

Siendo:

P_B : Potencia de la batería.

I_B : Corriente de la batería.

TA : Tiempo de autonomía de las baterías de la tabla 19-3.

Tabla 19-3: Consumo, potencia y tiempo de autonomía.

No	Tipo de Batería	Energía (Wh)	Tiempo de autonomía (h)
1	LiPO 1000 mA, 7.4V	3.7	0.7
2	LiPO 1500 mA, 7.4V	5.55	1.3
3	LiPO 2200 mA, 7.4V	8.14	1.51

Realizado por: Berrones A, Crespata C; 2021.

Con estos tiempos ya establecidos lo más adecuado es escoger la batería de 2200 mA para que el prototipo trabaje por más tiempo, pero el tamaño también es un parámetro muy importante para escoger el tipo de batería, por lo cual mediante la tabla 20-3 se muestra los tamaños de las baterías.

Tabla 20-3: Dimensiones de las baterías.

No	Tipo de Batería	Largo (mm)	Ancho (mm)	Alto (mm)
1	LiPO 1000 mA, 7.4V	72	34	12
2	LiPO 1500 mA, 7.4V	73	34	30
3	LiPO 2200 mA, 7.4V	110	34	13

Realizado por: Berrones A, Crespata C; 2021

En la tabla 21-3 se presentan las características de la batería que se va a implementar en el dispositivo. Al ser una batería Li-PO (polímero de iones de litio) de descarga alta, es una muy buena elección para alimentar el prototipo al ser un dispositivo portátil, y al requerir una gran demanda de energía, es una batería pequeña con mucha fuerza. La velocidad de descarga que posee la batería es suficientemente alta como para abastecer a los diferentes componentes que forman parte del prototipo (Electronilab, 2021).

Tabla 21-3: Características de la batería LiPO 2S 1000 mAh.

Detalle	Descripción
Capacidad	1000mAh
Voltaje	2S1P / 2 / celulares 7.4V
Descarga	25C Constante / 35C Burst
Peso	67 g (incluyendo cable y enchufe)
Dimensiones	72x34x15mm
Conector de balance	JST-XH
Conector de descarga	XT60

Fuente: (Electronila, 2021).

Realizado por: Berrones A, Crespata C; 2021.

Con estos datos se puede verificar que el tamaño de la batería de 2200 es demasiado grande por ello y más por criterios de construcción se escoge a la batería de 1000 mA para la integración en el dispositivo la cual se presenta en la figura 11-3.



Figura 11-3: Batería LiPO, 2 celdas 1000mAh.

Realizado por: Berrones A, Crespata C; 2021.

Se debe acotar que una de las principales ventajas de usar baterías LiPO, es que implementan una tasa de descarga que en las baterías LiPO medidas en C por ejemplo 20C, 40C etc. el cual

multiplica la corriente por este valor, esto quiere decir que si la batería seleccionada es de 1000 mA y 30C la corriente en realidad de la batería puede alimentar una carga con 30C por ello el cálculo se debe de replantear de la siguiente manera.

$$E_c = I_B * V_B * t \quad \text{Ecuación 7-3}$$

$$E_c = (500mA * 30) * 7.4 * 1h$$

$$E_c = 111Wh$$

Donde :

E_c : Energía que consume el prototipo.

t : Tiempo en horas.

V_B : Voltaje que suministra la batería.

I_C : Corriente de la batería.

Al aplicar la Ecuación 6-3 para determinar el tiempo de autonomía del prototipo, donde la energía de la batería es de 111 Wh y la potencia del prototipo es de 5.635 W, arroja un tiempo de 20 horas.

En base a los cálculos realizados el prototipo alcanza las 20 horas de funcionamiento teórico, porque los valores de potencia y consumo de la batería son teóricos, en la puesta en marcha del prototipo se comprobó que puede trabajar constantemente más de 6 horas debido a que no se puede llevar al límite de descarga a la batería, ya que su construcción es de tipo Li-po y la descarga total puede restar la vida útil de la batería. En relación al tiempo de recorrido del circuito turístico más distante se tiene que es de 60 min, este valor se representa como el 16,67% del total de duración por lo cual se concluye que tiene una autonomía aceptable.

CAPÍTULO IV

4 EVALUACIÓN ECONÓMICA

Para el desarrollo del sistema de información turística se destinaron diferentes materiales y equipos necesarios los cuales se encuentran detallados en los siguientes apartados.

4.1 Análisis de costos de materiales

A continuación, se presenta la lista de materiales y herramientas utilizadas en el desarrollo del dispositivo.

Tabla 1-4: Costos de materiales.

Cantidad	Detalle	Valor unitario	Total
1	Cable Flex para cámara Raspberry	\$9.89	\$9.89
1	Multímetro	\$10.00	\$10.00
1	Pantalla LCD TÁCTIL Kuman 3.5" 480X320 HDMI	\$21.99	\$21.99
1	Pi NoIR Camera V2 8 Megapíxeles	\$27.50	\$27.50
1	Batería	\$15.00	\$15.00
1	Impresión 3d	\$40.00	\$40.00
1	GPS NEO-6M	\$16.90	\$16.90
Total			\$141.28

Realizado por: Berrones A, Crespata C; 2021.

4.2 Análisis de costos de equipos

Los equipos utilizados para el desarrollo y ejecución del dispositivo se presentan a continuación tanto la cantidad utilizada como los costos de cada uno.

Tabla 2-4: Costos de equipos.

Cantidad	Detalle	Valor unitario	Total
1	Raspberry Pi 4	\$155.00	\$155.00
1	Teclado	\$15.00	\$15.00
1	Monitor	\$21.99	\$21.99
Total			\$191.99

Realizado por: Berrones A, Crespata C; 2021.

4.3 Costo total del dispositivo

El costo total generado por el desarrollo se calcula mediante la suma de cada costo que se generó en cada etapa del dispositivo el cual da como resultado de un costo final de \$ 333.27 el cual se detalla a continuación.

Tabla 3-4: Costo total del dispositivo.

Descripción	costos
Costos de materiales	\$141.28
Costos de equipos	\$191.99
Total	\$333.27

Realizado por: Berrones A, Crespata C; 2021.

En las tablas descritas anteriormente se aprecia los costos originados en el desarrollo del dispositivo.

4.4 Tabla comparativa de *softwares* de aplicaciones y prototipo

En la tabla 4-4 se realiza una comparación del prototipo con aplicaciones disponibles en la web, las cuales tiene una característica en común que es la de mostrar rutas hacia un destino, se ha tomado estas aplicaciones como motivo de validar la funcionalidad del prototipo y si este aporta diferentes funciones y características en razón de sus adversarios.

Tabla 4-4 : Tabla comparativa de *software* de aplicaciones con prototipo.

	PROTOTIPO	GOOGLE MAPS	WAZE
RUTAS	Predefinidas por la biblioteca gmaps. Alcance local. Información turística.	Rutas proporcionadas por las vías principales. Rutas proporcionadas para diferentes tipos de transporte alcance global.	Busca la ruta más rápida alcance global mayor información en rutas proporcionadas.
MAPAS	G_NORMAL_MAP	Mapa Callejero – G_NORMAL_MAP. Mapa Satélite – G_SATELLITE_MAP. Mapa Híbrido – G_HYBRID_MAP.	HOJA DE RUTA (normal, default 2D map).
MANEJO	Simple, intuitiva	Simple.	Fácil manejo.
IMAGEN Y DISEÑO(DISEÑO DE INTERFAZ)	Llamativo. Sin publicidad molesta.	Buen estilo visual. Posee Publicidad.	Calificado como infantil. Posee Publicidad.
INFORMACIÓN	Información detallada acerca de los sitios turísticos.	Posee la mayor base de datos disponible actualmente pero información no relacionada al turismo en sí.	Información básica.
NAVEGACIÓN OFFLINE	no	si	no

Realizado por: Berrones A, Crespata C; 2021.

De la tabla 4-4 se concluye que el prototipo desarrollado posee un limitante en cuanto a sus rutas presentadas, ya que este se encuentra desarrollado para un entorno en específico, pero en cuanto a imagen y diseño es superior además de su fácil manejo que es simple e intuitivo, a todo esto se añade el uso de una tecnología llamativa, que está en constante expansión en la industria como en la vida cotidiana, la visión artificial es la tecnología que cada día sigue abarcando más campos de aplicación, en esta ocasión ha sido un pilar fundamental en el desarrollo del prototipo logrando con esto un sistema de visión artificial innovador en cuanto al desarrollo del turismo local de un sitio en particular, por tal motivo el prototipo es superior en comparación a las aplicaciones presentadas en la tabla 4-4 en lo que concierne al turismo y visión artificial.

CONCLUSIONES

Se diseñó e implementó el prototipo de información turística, un dispositivo portátil capaz de reconocer la señalética del entorno en el Centro Histórico de la ciudad de Riobamba.

La eficiencia del dispositivo depende de la distancia, de la captura de la imagen y la iluminación por ello para obtener unos buenos resultados, la distancia de captura debe ser menor o igual a un metro y el nivel de iluminación debe estar entre el rango de 45 000 a 120000 luxes para obtener una eficiencia superior al 80%.

El prototipo es altamente veloz en el reconocimiento de señalética debido a que posee un tiempo de respuesta máximo de 5.6s, en relación a las 70 pruebas realizadas.

El prototipo posee un sistema de trazado de rutas altamente confiable, puesto que de todas las pruebas realizadas tuvo un 100% de efectividad además que se la comparo con software comercial en trazado de rutas.

El prototipo presenta una autonomía practica de 6 horas en funcionamiento, mientras que el tiempo de recorrido del circuito turístico más distante es 60 min, este valor se representa como el 16.67% del total de duración por lo cual se concluye que tiene una autonomía aceptable.

RECOMENDACIONES

Se recomienda ampliar el tiempo de pruebas para determinar posibles mejoras en el diseño del prototipo.

Se recomienda el uso del prototipo para incrementar el desarrollo turístico, expandir el área de trabajo del prototipo, no limitarlo a un lugar en específico, además incorporar el reconocimiento de nuevos sitios turísticos de la ciudad como hoteles, restaurantes etc.

Se recomienda aplicar inteligencia artificial para el trazado de rutas, para obtener la ruta más óptima, sin tráfico y con el menor tiempo posible hacia el destino.

Se recomienda el estudio de nuevos algoritmos que permitan la mejora del reconocimiento del prototipo, su tiempo de respuesta e incluyan nuevas funciones al mismo que se estudie la posibilidad de realizar un dispositivo más compacto en relación al actual o a su vez el desarrollo sea capaz de ser portable en una aplicación.

BIBLIOGRAFÍA

AGUIRRE, Omar de Jesús; & ARIAS PALENCIA, Luis Eduardo. Aplicación Android para silla de ruedas, que permita controlarla por comandos de voz, pantalla táctil y programación de ruta. [en línea] (tesis). (Pregrado) . Universidad Cooperativa de Colombia, Nariño, Colombia. 2016. pp. 42-43. [Consulta: 17 diciembre 2020]. Disponible en: http://repository.ucc.edu.co/bitstream/20.500.12494/14480/1/2016_aplicación_android_silla.pdf

AIGAJE ASTUDILLO, Ingrid Paola; & LÓPEZ VELASTEGUI, Lisette Katherine. Diseño de un circuito turístico para contribuir al desarrollo del turismo gerontológico en la ciudad de Milagro y Naranjito de la provincia del Guayas. [en línea] (tesis). (Pregrado) .Universidad Estatal De Milagro, Guayas, Ecuador. 2013. pp. 26. [Consulta: 03 diciembre 2020]. Disponible en: <http://repositorio.unemi.edu.ec/handle/123456789/758>.

ALBERT, Nogue; & ANTIGA, Jordi. Aplicación práctica de la visión artificial en el control de procesos industriales Conocimientos básicos de visión artificial., 2018. pp. 1-28. [Consulta: 20 mayo 2020]. Disponible en: <http://www.adimenlehiakorra.eus/documents/29934/43025/Aplicación+práctica+de+la+visión+artificial+en+el+control+de+procesos+industriales.pdf/92a1a455-ffe2-4290-be95-f9ba33c02241>

ALEGRE, Enrique; et al. *Conceptos y Métodos en Visión Por Computador.* España: CEA, 2016. pp. 1-372. [Consulta: 29 mayo 2020]. Disponible en: <https://intranet.ceautomatica.es/sites/default/files/upload/8/files/ConceptosyMetodosenVxC.pdf>.

AMAYA ZAPATA, Sebastian; et al. Desarrollo e Implementación de un Sistema de Visión Artificial Basado en Lenguajes de Uso Libre para un Sistema Seleccionador de Productos de un Centro Integrado de Manufactura (CIM). *Lámpsakos* [en línea], 2016. (15), ISSN 2145-4086. DOI 10.21501/21454086.1702. pp. 20 - 23 [Consulta: 15 julio 2020]. Disponible en: https://www.researchgate.net/publication/306023557_Desarrollo_e_Implementacion_de_un_Sistema_de_Vision_Artificial_Basado_en_Lenguajes_de_Uso_Libre_para_un_Sistema_Seleccionador_de_Productos_de_un_Centro_Integrado_de_Manufactura_CIM.

ANDRADE, Raquel. Estudio De Señalética Turística Para Impulsar El Desarrollo De La Actividad En La Zona De Intag, Cantón Cotacachi, Provincia De Imbabura [en línea] (tesis). (Pregrado) . Universidad internacional SEK, Cotacachi, Ecuador. 2014. pp. 50 - 53 [Consulta: 03 diciembre 2020]. Disponible en: <https://repositorio.uisek.edu.ec/handle/123456789/2683>.

ARACIL, Rafael. Desarrollo de un sistema cognitivo de visión para la navegación robótica. 2012. pp. 1-59. [Consulta: 08 junio 2020]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/17010/memoria.pdf?sequence=1>

ARÉVALO, Vicente; et al. La librería de visión artificial opencv aplicación a la docencia e investigación 1. [en línea] , 2016. pp. 1-6. [Consulta: 12 noviembre 2020]. Disponible en: https://www.researchgate.net/publication/236668252_La_Libreria_de_Vision_Artificial_OpenC_V_Aplicacion_a_la_Docencia_e_Investigacion_in_spanish.

AVELECTRONICS, 2020. Módulo GPS NEO-6M | AV Electronics. [en línea] . [Consulta: 22 febrero 2021]. Disponible en: <https://avelectronics.cc/producto/modulo-gps-neo-6m/>.

BOULLOSA, Oscar. Estudio Comparativo de Descriptores Visuales Para la Deteccion de Escenas Cuasi-Duplicadas. [en línea](tesis).(Pregrado). Escuela Politecnica Superior Universidad Autonoma de Madrid, Madrid, España. 2011. pp.75. [Consulta: 01 agosto 2021]. Disponible en: https://repositorio.uam.es/bitstream/handle/10486/9977/51258_20110318OscarBoullosa.pdf;sequence=1.

CAJAS IDROVO, Marco Vinicio; & VIRI AVIAL, Pablo Andres. Diseño e implementacion de un sistema de seguridad vehicular mediante reconocimiento facial a traves de visión artificial. [en línea] (tesis). (Pregrado) . Universidad Politécnica Salesiana Sede Cuenca, Cuenca , Ecuador. 2017. pp. 41-43. [Consulta: 15 julio 2020]. Disponible en: <http://dspace.ups.edu.ec/bitstream/123456789/13566/1/UPS-CT006920.pdf>.

CAMPBELL SCIENTIFIC. (2021). *GPS Receiver with integrated Antenna.* [en línea] . [Consulta: 08 agosto 2021]. Disponible en: <https://www.campbellsci.es/gps16x-hvs>.

CAYAMBE, Diana; & CASTAÑEDA, Juan. Inteligencia artificial aplicada a robot asistencial para la interacción con niños en edad temprana. [en línea] (tesis). (Pregrado) . Escuela Superior Politécnica De Chimborazo, Riobamba , Ecuador. 2016. pp. 11. [Consulta: 02 junio 2020]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/6036>.

DUEÑAS, Platero. 2014. Tema 1.- Introducción a la Visión Artificial.Universidad de Córdoba: Escuela Politécnica Superior [en línea] . pp. 15-18. [Consulta: 05 mayo 2020]. Disponible en: <http://www.uco.es/users/ma1fegan/2012-2013/vision/Temas/ruido.pdf>.

DUNAI, Larisa; et al. Sistema de comunicación y reconocimiento de señalética en entornos cerrados y abiertos para personas ciegas., 2017. pp. 3-7. [Consulta: 19 noviembre 2020]. Disponible en: https://www.researchgate.net/publication/325124807_Sistema_de_comunicacion_y_reconocimi

ento_de_senaletica_en_entornos_cerrados_y_abiertos_para_personas_ciegas

ELECTRONILAB. (2021). *Baterías Lipo*. [en línea] . [Consulta: 15 febrero 2021]. Disponible en: <https://electronilab.co/tienda/bateria-lipo-de-1000mah-7-4v/>.

ELEKTORMAG. (2018). *The Ultimate Guide to Raspbian and other Raspberry Pi Software*. [en línea] . [Consulta: 15 febrero 2021]. Disponible en: <https://www.elektormagazine.com/news/the-ultimate-guide-to-raspbian-and-other-raspberry-pi-software>.

GARCÍA, Ines; et al. Comparación de algoritmos detectores de puntos singulares para reconocimiento de objetos en vídeo quirúrgico., 2014. pp. 1-4 [Consulta: 19 octubre 2020]. Disponible en: <http://oa.upm.es/20480/>.

GARCÍA, Ivan; & CARANQUI, Victor. La visión artificial y los campos de aplicación. Tierra Infinita [en línea] , 2015. pp. 93-103. [Consulta: 26 mayo 2020]. Disponible en: <http://revistasdigitales.upec.edu.ec/index.php/tierrainfinita/article/view/76>.

GONZALES, Ana; et al. TECNICAS Y ALGORITMOS BASICOS DE VISION ARTIFICIAL. En: U. de la R.S. DE PUBLICACIONES (ed.). 2016. pp. 90-91. [Consulta: 14 abril 2020]. Disponible en: <https://publicaciones.unirioja.es/catalogo/monografias/mdi24.shtml>.

GRANIZO, Sergio. Sistema web de ubicación geográfica y trazo de rutas de transporte para llegar a los colegios de quito [en línea] (tesis). (Pregrado). Universidad Central Del Ecuador, Quito, Ecuador. 2016. pp. 37. [Consulta: 31 mayo 2020]. Disponible en: <http://www.dspace.uce.edu.ec/bitstream/25000/6055/1/T-UCE-0011-255.pdf>.

GUERRERO, Pablo Nicolás. Análisis biomecánico del cuerpo humano mediante el procesamiento digital de imágenes [en línea] (tesis). (Pregrado) . Universidad Tecnológica Nacional, Buenos Aires , Argentina. 2018. pp. 1 - 59. [Consulta: 31 mayo 2020]. Disponible en: <https://ria.utn.edu.ar/handle/20.500.12272/2872>.

JUÁREZ, César. Reconocimiento E Identificación De Logotipos En Imágenes Con Transformada Sift. [en línea] (tesis). (Pregrado). Universidad Carlos III de Madrid, Getafe, España. 2013. pp. 19. [Consulta: 28 julio 2021]. Disponible en: <http://docplayer.es/3758361-Proyecto-fin-de-carrera.html>.

LOYOLA, Katty. 2018. Folleto Digital Para Mejorar La Información Turística Del Cantón

Riobamba. Interciencia [en línea] . pp. 25-28. [Consulta: 21 enero 2021]. Disponible en: <http://dspace.uniandes.edu.ec/bitstream/123456789/9154/1/PIURHYT011-2018.pdf>.

MADUELL, Eloi. Visión artificial. España: UOC, 2016. pp. 55-59. [Consulta: 26 julio 2021]. Disponible en: https://docplayer.es/19036658-Vision-artificial-eloi-maduell-i-garcia-pid_00184756.html.

MANOBANDA, Luis. Implementación De Un Prototipo De Máquina Clasificadora Y Contadora De Monedas Basada En Visión Artificial [en línea] (tesis). (Pregrado). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2019. pp. 22 - 25 [Consulta: 02 junio 2021]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/13514>.

MEJÍA, Danilo. Diseño De Un Modelo De Catastro Patrimonial Georeferenciado Para El Centro Histórico Del Cantón Riobamba, Provincia De Chimborazo [en línea] (tesis). (Pregrado). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2015. pp. 33 - 37. [Consulta: 28 enero 2021]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/4235>.

MUNDACA, George. Detección de caracteres de placas de automóviles mediante técnicas de visión artificial [en línea] (tesis). (Pregrado). Universidad de Piura, Piura, Peru. 2016. pp. 43 - 45. [Consulta: 31 mayo 2020]. Disponible en: <https://pirhua.udep.edu.pe/handle/11042/2578>.

NARRO, Alberto. Comparativa de algoritmos de detección de características para visión artificial [en línea] (tesis). (Pregrado). Universidad Politécnica de Madrid, Madrid, España. 2019. pp. 51 - 54. [Consulta: 07 marzo 2020]. Disponible en: <http://oa.upm.es/56421/>.

OPENCV, (2021). Feature Matching methods comparison in OpenCV Master Data Science. [en línea] . [Consulta: 23 julio 2021]. Disponible en: <http://datahacker.rs/feature-matching-methods-comparison-in-opencv/>.

PACHECO, Javier. Sistema De Reconocimiento De Objetos Removidos De Una Escena, Utilizando Visión Por Computador [en línea] (tesis). (Pregrado). Pontificia Universidad Javeriana, Bogota, Colombia. 2011. pp. 35 - 37. [Consulta: 14 diciembre 2020]. Disponible en: <https://repository.javeriana.edu.co/bitstream/handle/10554/7073/tesis535.pdf?isAllowed=y&sequence=1>.

PALOMINO, Nora La Serna; & CONCHA ROMAN, Ulises Norberto. 2014. “Técnicas de Segmentación en Procesamiento Digital de Imágenes”. Revista de ingeniería de sistemas e informática [en línea] . pp. 19. [Consulta: 31 mayo 2020]. Disponible en:

<https://www.semanticscholar.org/paper/Técnicas-de-Segmentación-en-Procesamiento-Digital-Palomino-Concha/199c8cac02503a705bf2005e3353539750f4e7ee>.

PONCE, Pilar. *Estudio De La Contaminación Lúminica Y Eficiencia Energética En Alumbrado Exterior* [en línea] 2017. pp. 83. [Consulta: 27 febrero 2021]. Disponible en: <https://repositorio.upct.es/bitstream/handle/10317/4539/tfm382.pdf?sequence=1&isAllowed=y>.

PYPI. (2020). *PyQt5 5.15.2*. [en línea] . [Consulta: 08 agosto 2020]. Disponible en: <https://pypi.org/project/PyQt5/>.

RASPBERRY. (2016). *Puertos GPIO* [en línea] . 2016. [Consulta: 14 diciembre 2020]. Disponible en: <https://prototipopi.wordpress.com/2016/05/02/puertos-gpio/>.

RASPBERRY PI. (2019). *Products Archive - Raspberry Pi*. [en línea] . [Consulta: 08 agosto 2020]. Disponible en: raspberrypi.org/products/.

RIOBAMBA. (2020). *Lugares turísticos de Riobamba*. [en línea] . [Consulta: 17 diciembre 2020]. Disponible en: <https://riobamba.com.ec/es-ec/chimborazo/riobamba/rutas-paseos/lugares-turisticos-riobamba-a1ldysrfx>.

ROCHA, Casimiro; & ESCORCIA, José. Sistema de Visión Artificial para la Detección y el Reconocimiento de Señales de Tráfico basado en Redes Neuronales. *Innovation and Development for the Americas* [en línea] , 2010, (Peru) 1(2), pp. 1 - 10. [Consulta: 03 diciembre 2020]. Disponible en: http://www.laccei.org/LACCEI2010-Peru/Papers/Papers_pdf/TS158_Rocha.pdf.

RODRÍGUEZ, Gabriel Escalas. Diseño y desarrollo de un prototipo de riego automático controlado con Raspberry Pi y Arduino [en línea] (tesis). (Pregrado). Universidad Politécnica de Cataluña, Barcelona, España. 2015. pp. 65 - 69. [Consulta: 02 junio 2020]. Disponible en: <http://hdl.handle.net/2099.1/25074>.

ROMERO, Alvaro; et al. Sistema de clasificación por visión artificial de mangos tipo Tommy. *Revista UIS Ingenierías* [en línea] , 2015, (Peru) 14(1), pp. 21-31. [Consulta: 12 agosto 2020]. Disponible en: <https://www.redalyc.org/pdf/5537/553756867002.pdf>.

ROMERO, Anna; & CAZORLA, Miguel. 2009. “Comparativa de detectores de características visuales y su aplicación al SLAM”. *Repositorio Institucional de la Universidad de Alicante*. [en línea]. pp. 62. [Consulta: 31 mayo 2021]. Disponible en: <http://rua.ua.es/dspace/handle/10045/23389>

SAQUINGA, Tatiana; & ANDRADE, Carolaen. Implementación De Un Prototipo De Sistema

Electrónico En Un Andador Para Niños Que Controle El Acceso A Lugares Inseguros, Usando Visión Artificial. [en línea] (tesis). (Pregrado). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2019. pp.26-40. [Consulta: 21 julio 2020]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/13594>.

SENPLADES. *Plan Nacional para el Buen Vivir 2017-2021: Construyendo un Estado Plurinacional e Intercultural.* [en línea], 2017 . pp.11-13. [Consulta: 23 noviembre 2020]. Disponible en: <https://www.gobiernoelectronico.gob.ec/wp-content/uploads/downloads/2017/09/Plan-Nacional-para-el-Buen-Vivir-2017-2021.pdf>.

SOLANO, Gabriela. Diseño Y Construcción De Un Dispositivo Portatil Para La Identificación De Billetes Orientado A Personas Con Discapacidad Visual Mediante El Uso De Visión Artificial [en línea] (tesis). (Pregrado). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. 2018. pp. 71-76. [Consulta: 16 enero 2021]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/9218/1/108T0260.pdf> .

TECHMAKE. (2020). Módulo GPS Ublox NEO-6M con Antena— Techmake Solutions. [en línea] . [Consulta: 22 febrero 2021]. Disponible en: <https://techmake.com/collections/modulos-de-gps/products/gpsmod00364>.

TIENDATEC, 2019. Pantallas [en línea] . 2019. Disponible en: <https://www.tiendatec.es/33-pantallas>.

VIJAYAN, Vineetha; & KP Pushpalatha. 2019. FLANN Based Matching with SIFT Descriptors for Drowsy Features Extraction. Proceedings of the IEEE International Conference Image Information Processing [en línea]. pp. 600-605. [Consulta: 23 julio 2021]. Disponible en: <https://ieeexplore.ieee.org/document/8985924>.


VILLAGOMEZ, Ximena, Resolución De Imágenes Tomográficas Mediante Matlab [en línea] (tesis). (Pregrado). Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador. [Consulta: 23 julio 2021]. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/14297>.

VILLCA, Cristian; & MURGA, Mónica. Algoritmo SIFT para la detección de imágenes coincidentes. *Illuminate* [en línea] , 2017, Bolivia 9 (1), pp. 37-39. [Consulta: 05 mayo 2020]. Disponible en: http://www.revistasbolivianas.org.bo/pdf/riei/v9n1/v9n1_a05.pdf.

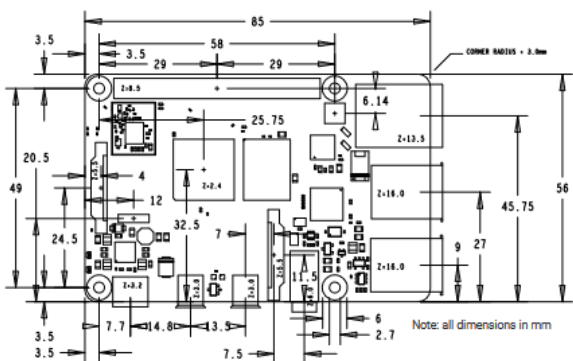
ZANUTO, Edneia; et al. *Manual de señalización turística.* [en línea], 2011. pp. 33. [Consulta: 17 diciembre 2020]. Disponible en: <https://jorgepaguay.files.wordpress.com/2011/10/manual-sec3b1alizacic3b3n-25-08-2011.pdf>.

ANEXOS

Anexo A: Datos técnicos de Raspberry PI 4B.

<p>Published in June 2019 by Raspberry Pi Trading Ltd</p> <p>www.raspberrypi.org</p> <h3>Overview</h3>  <p>Raspberry Pi 4 Model B is the latest product in the popular Raspberry Pi range of computers. It offers ground-breaking increases in processor speed, multimedia performance, memory, and connectivity compared to the prior-generation Raspberry Pi 3 Model B+, while retaining backwards compatibility and similar power consumption. For the end user, Raspberry Pi 4 Model B provides desktop performance comparable to entry-level x86 PC systems.</p> <p>This product's key features include a high-performance 64-bit quad-core processor, dual-display support at resolutions up to 4K via a pair of micro-HDMI ports, hardware video decode at up to 4Kp60, up to 4GB of RAM, dual-band 2.4/5.0 GHz wireless LAN, Bluetooth 5.0, Gigabit Ethernet, USB 3.0, and PoE capability (via a separate PoE HAT add-on).</p> <p>The dual-band wireless LAN and Bluetooth have modular compliance certification, allowing the board to be designed into end products with significantly reduced compliance testing, improving both cost and time to market.</p> <p>2 Raspberry Pi 4 Model B Product Brief</p>	<h3>Specification</h3> <p>Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz</p> <p>Memory: 1GB, 2GB or 4GB LPDDR4 (depending on model)</p> <p>Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.</p> <p>GPIO: Standard 40-pin GPIO header (fully backwards-compatible with previous boards)</p> <p>Video & sound: 2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port</p> <p>Multimedia: H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics</p> <p>SD card support: Micro SD card slot for loading operating system and data storage</p> <p>Input power: 5V DC via USB-C connector (minimum 3A) 5V DC via GPIO header (minimum 3A) Power over Ethernet (PoE)-enabled (requires separate PoE HAT)</p> <p>Environment: Operating temperature 0–50°C</p> <p>Compliance: For a full list of local and regional product approvals, please visit https://www.raspberrypi.org/documentation/hardware/raspberrypi/conformity.md</p> <p>Production lifetime: The Raspberry Pi 4 Model B will remain in production until at least January 2026.</p>
--	--

Physical Specifications

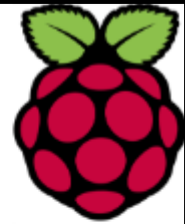


Note: all dimensions in mm

WARNINGS

- This product should only be connected to an external power supply rated at 5V/3A DC or 5.1V/ 3A DC minimum. Any external power supply used with the Raspberry Pi 4 Model B shall comply with relevant regulations and standards applicable in the country of intended use.
- This product should be operated in a well-ventilated environment and, if used inside a case, the case should not be covered.
- This product should be placed on a stable, flat, non-conductive surface in use and should not be contacted by conductive items.
- The connection of incompatible devices to the GPIO connection may affect compliance and result in damage to the unit and invalidate the warranty.
- All peripherals used with this product should comply with relevant standards for the country of use and be marked accordingly to ensure that safety and performance requirements are met. These articles include but are not limited to keyboards, monitors and mice when used in conjunction with the Raspberry Pi.
- Where peripherals are connected that do not include the cable or connector, the cable or connector must offer adequate insulation and operation in order that the relevant performance and safety requirements are met.

Anexo B: Datos técnicos Cámara pi NoIR Camera V2



Raspberry Pi NoIR Camera v2

Part number: RPI NOIR CAMERA BOARD



- 8 megapixel camera capable of taking infrared photographs of 3280 x 2464 pixels
- Capture video at 1080p30, 720p60 and 640x480p90 resolutions
- All software is supported within the latest version of Raspbian Operating System

The Raspberry Pi NoIR Camera Module v2 is a high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, featuring a fixed focus lens. It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p60/90 video. It attaches to Pi by way of one of the small sockets on the board upper surface and uses the dedicated CSI interface, designed especially for interfacing to cameras.

- 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images
- Supports 1080p30, 720p60 and 640x480p90 video
- Camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system

The board itself is tiny, at around 25mm x 23mm x 9mm. It also weighs just over 3g, making it perfect for mobile or other applications where size and weight are important. It connects to Raspberry Pi by way of a short ribbon cable.

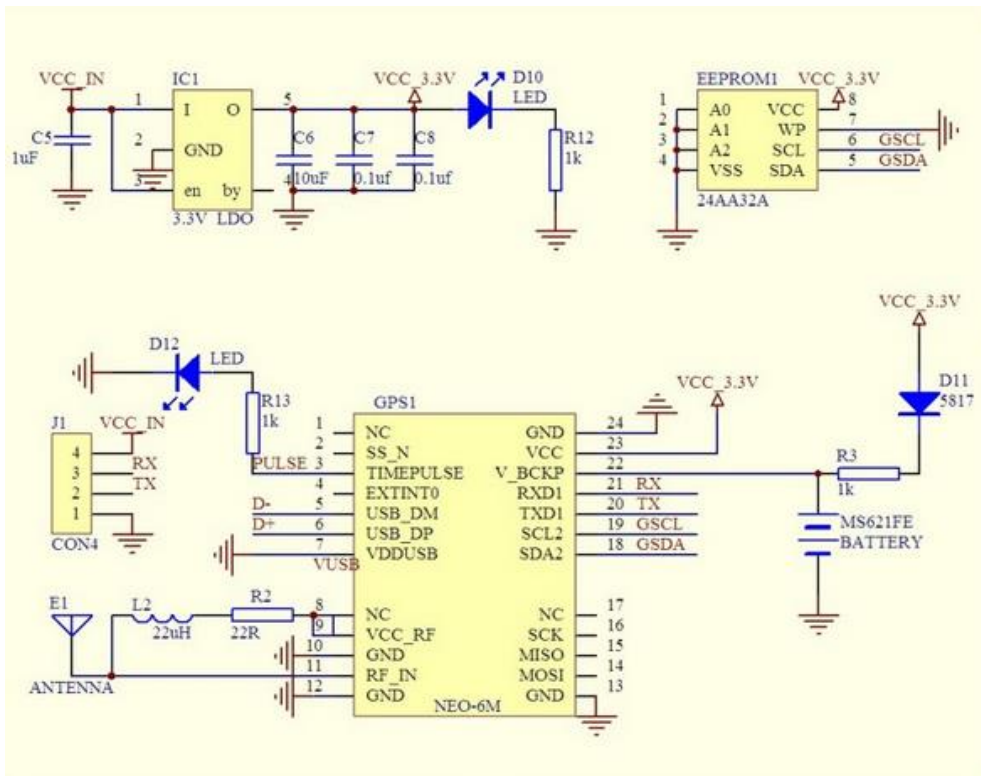
The high quality Sony sensor itself has a native resolution of 8 megapixel, and has a fixed focus lens on-board. In terms of still images, the camera is capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.

The NoIR Camera has No InfraRed (NoIR) filter on the lens which makes it perfect for doing Infrared photography and taking pictures in low light (twilight) environments.

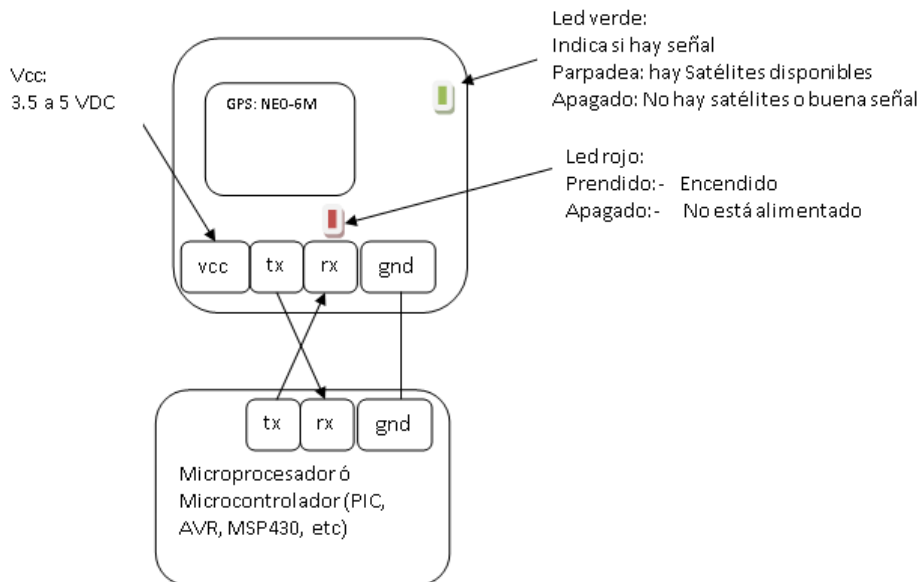
Applications

- Infrared photography
- Low light photography
- Monitoring plant growth
- CCTV security camera

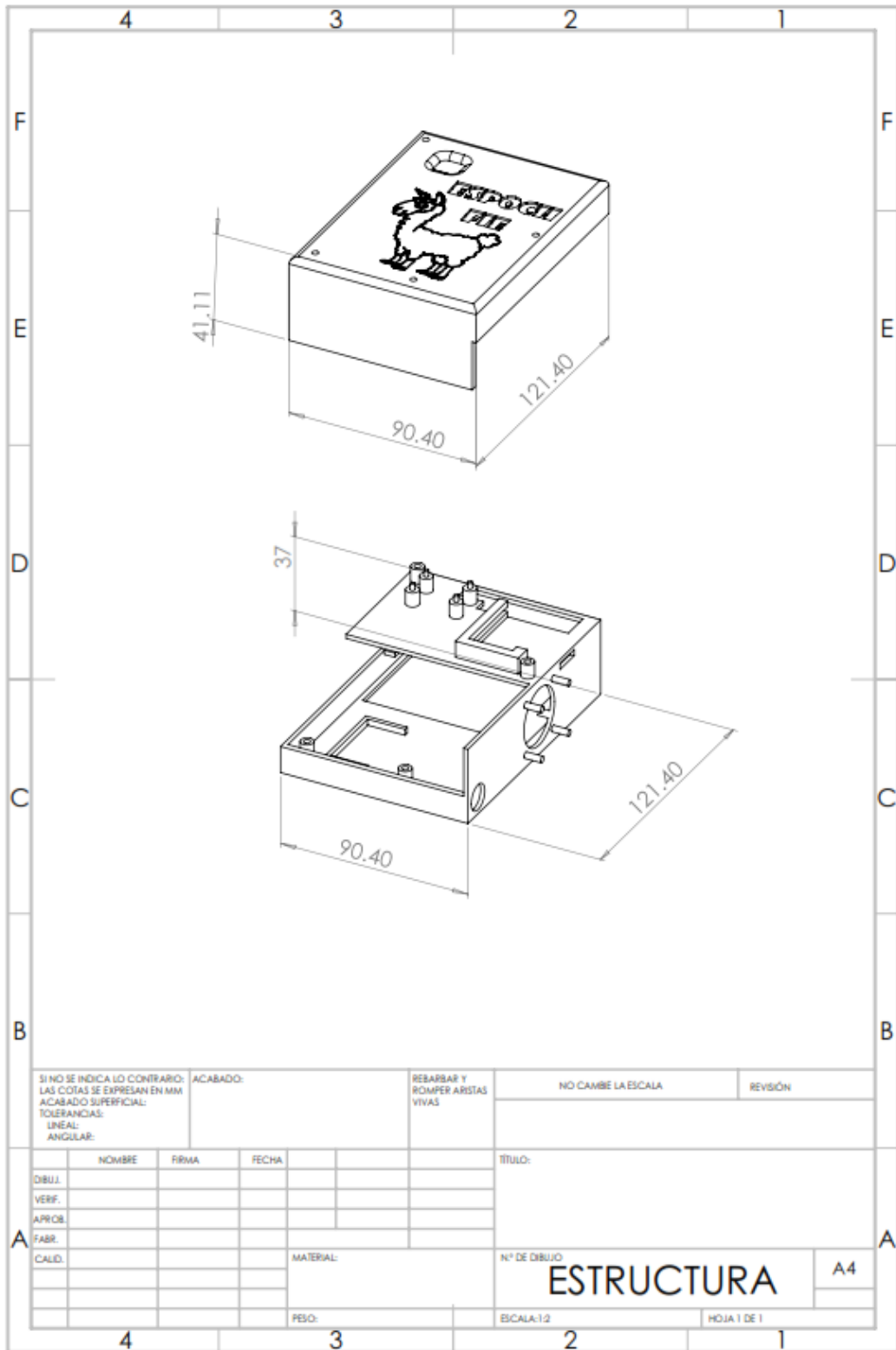
Anexo C: Esquema de conexión GPS Ublox NEO-6M



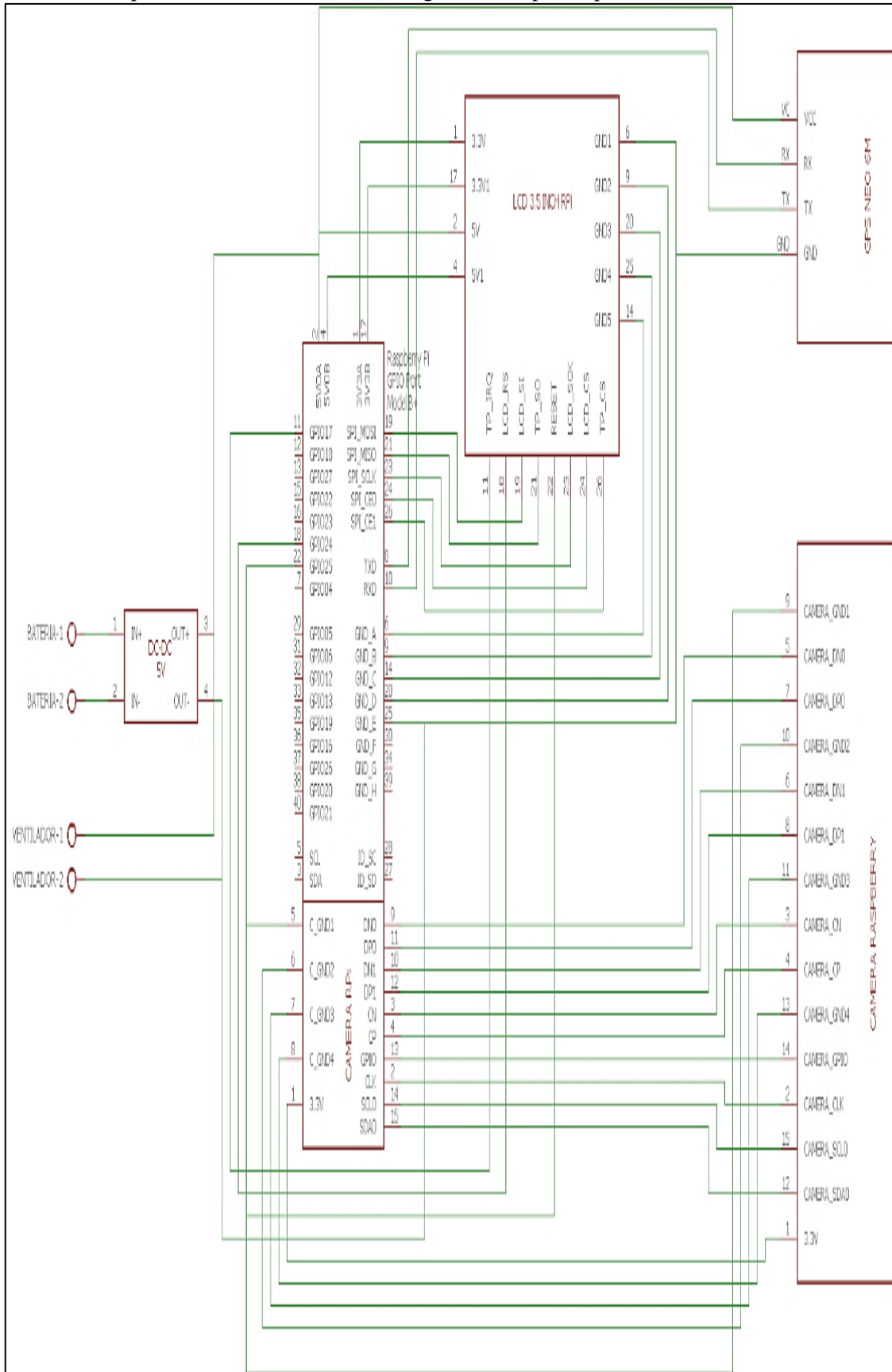
Conexión a microcontrolador



Anexo D: Medidas de la estructura del prototipo



Anexo E: Esquema de conexión del circuito general del prototipo.



Anexo F: Software del prototipo.

```
import os
import cv2
import sys
import time
import math
import serial
import string
import pynmea2
import webbrowser
import openpyxl
import numpy as np
import pytesseract
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from PyQt5.uic import loadUi
from PyQt5.QtCore import QUrl
from selenium import webdriver
from PyQt5.QtCore import QTimer
from pyvirtualdisplay import Display

lugar = ['IGLESIA', 'PARQUE', 'EDIFICIO', 'PLAZA', 'PLAZA DE TORROS', 'MUSEO']
global nom
nom=""
global informacion
global actI
global actV
global foto
global url

informacion = "
foto = "
url = "
actV = False
actI = False

print('cargo')
# *****
# ***** message box *****
class CustomMessageBox(QMessageBox):

    def __init__(self, *_args):
        QMessageBox.__init__(self)
        self.timeout = 0
        self.autoclose = False
        self.currentTime = 0

    def showEvent(self, QShowEvent):
        self.currentTime = 0
        if self.autoclose:
            self.startTimer(1000)
```

```

def timerEvent(self, *args, **kwargs):
    self.currentTime += 1
    if self.currentTime >= self.timeout:
        self.done(0)

    @staticmethod
    def showWithTimeout(timeoutSeconds, message, title, icon=QMessageBox.Information,
        buttons=QMessageBox.Ok):
        w = CustomMessageBox()
        w.autoclose = True
        w.timeout = timeoutSeconds
        w.setText(message)
        w.setWindowTitle(title)
        w.setIcon(icon)
        w.setStandardButtons(buttons)
        w.exec_()

# *****

class Loading(QMainWindow):
    def __init__(self):
        super>Loading, self).__init__()
        loadUi('pag.ui', self)
        self.btn_iniciar.clicked.connect(self.iniciar)
        self.btn_galeria.clicked.connect(self.galeria)

    def galeria(self):
        self.hide()
        self.var1 = base()
        self.var1.setWindowTitle('Base de Datos')
        self.var1.show()

    def iniciar(self):
        self.hide()
        self.var2 = inicio()
        self.var2.setWindowTitle('INICIO')
        self.var2.show()

# *****
# ***** GALERIA *****
# *****

class base(QMainWindow):
    def __init__(self):
        super>base, self).__init__()
        loadUi('Base.ui', self)
        self.btn_subir.clicked.connect(self.cargarImagen)
        self.btn_regresar.clicked.connect(self.back)
        a = os.listdir('baseIma/')
        for i in a:
            self.menu.addItem(i)

    def cargarImagen(self):

```

```

self.data = self.menu.currentText()
print(self.data)
datos = 'baseIma/' + self.data
print(datos)
img = cv2.imread(datos)
qformat = QImage.Format_Indexed8
if len(img.shape) == 2:
    qformat = QImage.Format_Grayscale8
if len(img.shape) == 3:
    if img.shape[2] == 4:
        qformat = QImage.Format_RGBA8888
    else:
        qformat = QImage.Format_RGB888
outimg = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)
outimg = outimg.rgbSwapped()
self.ima.setPixmap(QPixmap.fromImage(outimg))
self.ima.setScaledContents(True)

def back(self):
    self.hide()
    self.principal = Loading()
    self.principal.setWindowTitle('MENU PRINCIPAL')
    self.principal.show()

# *****
# ***** INICIO *****
# *****
class inicio(QMainWindow):
    def __init__(self):
        super(inicio, self).__init__()
        loadUi('Inicio.ui', self)
        self.btn_tomar.clicked.connect(self.tomarImagen)
        self.btn_refresh.clicked.connect(self.reiniciar)
        self.btn_regresar.clicked.connect(self.regre)
        self.status = True
        self.captura = cv2.VideoCapture(0)
        self.timer = QTimer(self)
        self.timer.timeout.connect(self.actualizar)
        self.timer.start(10)
        self.actE = False

def tomarImagen(self):
    t = time.time()
    self.timer.stop()
    datos2 = "
#ret, self.imagen = self.captura.read()
self.imagen = cv2.imread('imagenes/ima5.jpg')
a = os.listdir('baseIma/')
value = 0
for i in a:
    MIN_MATCH_COUNT = 5
    datos = 'baseIma/' + i
    img1 = cv2.imread(datos,0)
    img2n = self.imagen.copy()

```

```

img2 = cv2.cvtColor(img2n, cv2.COLOR_BGR2GRAY)
sift = cv2.xfeatures2d.SIFT_create()
kp1, des1 = sift.detectAndCompute(img1, None)
kp2, des2 = sift.detectAndCompute(img2, None)
FLANN_INDEX_KDTREE = 0
index_params = dict(algorithm = FLANN_INDEX_KDTREE, trees = 5)
search_params = dict(checks=50) # or pass empty dictionary
flann = cv2.FlannBasedMatcher(index_params, search_params)
matches = flann.knnMatch(des1, des2, k=2)
matchesMask = [[0,0] for i in range(len(matches))]
goodMatch = []
for j, (m, n) in enumerate(matches):
    if m.distance < 0.8*n.distance:
        matchesMask[j]=[1,0]
        goodMatch.append(m)
if len(goodMatch) > value and len(goodMatch) > MIN_MATCH_COUNT:
    value = len(goodMatch)
    try:
        tp = []
        qp = []
        for m in goodMatch:
            qp.append(kp2[m.trainIdx].pt)
            tp.append(kp1[m.queryIdx].pt)
        tp, qp = np.float32((tp, qp))
        H, status = cv2.findHomography(tp, qp, cv2.RANSAC, 3.0)
        h, w = img1.shape
        trainBorder = np.float32([[[[0, 0], [0, h - 1], [w - 1, h - 1], [w - 1, 0]]]])
        queryBorder = cv2.perspectiveTransform(trainBorder, H)
        m, n, p = queryBorder.shape
        if m == 1 and n == 4 and p == 2:
            img4 = cv2.polylines(img2n, [np.int32(queryBorder)], True, (0, 0, 255), 3)
            a = i.find('.jpg')
            datos2 = i[0:a - 1]
            global nom
            nom = i[0:a - 1]
            he = queryBorder[0][0][0]
            wi = queryBorder[0][0][1]
            org = (int(he), int(wi))
            img4 = cv2.putText(img4, datos2, org, cv2.FONT_HERSHEY_SIMPLEX, 1.2,
(255, 0, 0), 6, cv2.LINE_AA)
            if True:
                draw_params = dict(matchColor = (0,255,0), singlePointColor = (255,0,0),
matchesMask = matchesMask, flags = 0)
                img4 =
cv2.drawMatchesKnn(img1, kp1, img2, kp2, matches, None, **draw_params)
                cv2.imshow('name', img4)
                self.mostrar(img4)
            except:
                print('error')
        else:
            #print("No se encontro coincidencia")
            pass
    e = time.time() - t
    print(e)

```

```

if datos2 != "":
    QMessageBox.about(self, "AVISO", "La señalectica es : " + datos2)
    self.mostrarLocal()

else:
    QMessageBox.about(self, "AVISO", "No se encontro coincidencia se usara
reconocimiento de palabras")
    self.OCR(img2n)

def OCR(self, img):
    image=img.copy()
    img_to_yuv = cv2.cvtColor(img,cv2.COLOR_BGR2YUV)
    img_to_yuv[:, :,0] = cv2.equalizeHist(img_to_yuv[:, :,0])
    result = cv2.cvtColor(img_to_yuv, cv2.COLOR_YUV2BGR)
    img = cv2.cvtColor(result, cv2.COLOR_RGB2GRAY)
    kernel = np.ones((2, 2), np.uint8)
    img = cv2.dilate(img, kernel, iterations=1)
    ret,img = cv2.threshold(img,10,80,cv2.THRESH_TRUNC)
    custom_config = r'--oem 3 --psm 6'
    text = pytesseract.image_to_string(img, config=custom_config)
    texto=text.upper()
    print(texto)
    datos=""
    j=0
    self.lugaresDect.addItem("")
    for i in lugar:
        a=texto.find(i)
        if a != -1:
            j = j+1
            datos=i
            self.lugaresDect.addItem(datos)

    if j == 1:
        font = cv2.FONT_HERSHEY_SIMPLEX
        he, wi=img.shape[0:2]
        org = (int(wi/2-(wi*0.1)), int(he*0.1))
        print(org)
        fontScale = 2
        color = (0, 0, 255)
        thickness = 8
        cv2.putText(image, datos, org, font, fontScale, color, thickness, cv2.LINE_AA)
        self.mostrar(image)
        QMessageBox.about(self, "AVISO", "La señalectica es :" + datos)
        global nom
        nom=datos
        self.mostrarLocal()

    if j > 1:
        QMessageBox.about(self, "AVISO", "Existe mas de una señal o lugar detectado
porfavor escoja uno de ellos")
        self.actE = True
        self.timer.start(10)

    else:
        QMessageBox.about(self, "AVISO", "No se encontro coincidencia textuales")
        self.mostrar(image)

```

```

def reiniciar(self):
    self.timer.start(10)

def actualizar(self):
    if self.actE == False:
        ret, self.imagen = self.captura.read()
        #self.imagen = cv2.imread('imagenes/7.jpg')
        self.mostrar(self.imagen)
    else:
        data = self.lugaresDect.currentText()
        print(data)
        if data != "":
            global nom
            nom = data
            self.mostrarLocal()
            self.timer.stop()

def mostrar(self, img):
    qformat = QImage.Format_Indexed8
    if len(img.shape) == 3:
        if img.shape[2] == 4:
            qformat = QImage.Format_RGBA8888
        else:
            qformat = QImage.Format_RGB888
    outimg = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)
    outimg = outimg.rgbSwapped()
    self.pantalla.setPixmap(QPixmap.fromImage(outimg))
    self.pantalla.setScaledContents(True)

def detect(self, img):
    pass

def regre(self):
    self.captura.release()
    self.hide()
    self.principal = Loading()
    self.principal.setWindowTitle('MENU PRINCIPAL')
    self.principal.show()

def mostrarLocal(self):
    self.captura.release()
    self.hide()
    self.var1 = mLocal()
    self.var1.setWindowTitle('Menu de Rutas')
    self.var1.show()
# *****
# ***** LOCALIZACION *****
# *****
class mLocal(QMainWindow):
    def __init__(self):
        super(mLocal, self).__init__()

```



```

loadUi('localizacion.ui', self)
self.btn_iniciarNave.clicked.connect(self.iniciarNavegacion)
self.btn_iniciarRuta.clicked.connect(self.iniRuta)
self.btn_regresarRu.clicked.connect(self.regre3)
self.senal.setText(nom)
lista = os.listdir('lugares/')
listaN=[]
for i in lista:
    a = i.find('.xlsx')
    ap = i[0:a]
    listaN.append(ap)
l=""
for i in listaN:
    if nom == i:
        l=i
        break
path = 'lugares/'+l+'.xlsx'
wb = openpyxl.load_workbook(path)
self.hoja = wb.active
self.fil = self.hoja.max_row
for i in range(1, self.fil + 1):
    datos = self.hoja.cell(row=i, column=1)
    print(datos.value)
    self.listaLugares.addItem(str(datos.value))

port = '/dev/ttyAMA0'
self.ser = serial.Serial(port, baudrate=9600, timeout=1)
act = True
lat = 0
lng = 0
while act:
    try:
        newdata = self.ser.readline().decode('utf-8')
        if newdata[0:6] == '$GPRMC':
            newmsg = pynmea2.parse(newdata)
            lat = newmsg.latitude
            lat = round(lat,6)
            lng = newmsg.longitude
            lng = round(lng,6)
            gps = 'Latitud=' + str(lat) + ' Longitud=' + str(lng)
            print(gps)
            lat=-1.6529528
            lng=-78.6682284
    except:
        print('no hay señal')
        if lng!=0 and lat != 0:
            act=False
            self.lat_Ob = 0
self.lng_Ob = 0
self.lat_La = 0
self.lng_La = 0
self.timer1 = QTimer(self)
self.timer1.timeout.connect(self.lectu)
self.timer1.start(100)
self.paso=0

```

```
self.actB = False
self.da = 0
```

```
def lectu(self):
```

```
    data = self.listaLugares.currentText()
    actC = False
    for i in range(1, self.fil + 1):
        datos = self.hoja.cell(row=i, column=1)
        if datos.value == data:
            self.latitud = self.hoja.cell(row=i, column=2)
            self.longitud = self.hoja.cell(row=i, column=3)
            direccion = self.hoja.cell(row=i, column=4)
            self.latL.setText(str(self.latitud.value))
            self.longL.setText(str(self.longitud.value))
            self.dir.setText(str(direccion.value))
            self.lat_La = self.latitud.value
            self.lng_La = self.longitud.value
            global informacion
            informacion = str(self.hoja.cell(row=i, column=5).value)
            informacion = data + '\n \n' + informacion
            global foto
            foto = self.hoja.cell(row=i, column=6).value
            actC = True
```

```
lat = 0
```

```
lng = 0
```

```
try:
```

```
    newdata = self.ser.readline().decode('utf-8')
    if newdata[0:6] == '$GPRMC':
        newmsg = pynmea2.parse(newdata)
        lat = newmsg.latitude
        lat = round(lat,6)
        lng = newmsg.longitude
        lng = round(lng,6)
        lat=-1.6529528
        lng=-78.6682284
        gps = 'Latitud=' + str(lat) + ' Longitud=' + str(lng)
```

```
except:
```

```
    print('no hay señal')
```

```
if lat!=0 and lng!= 0:
```

```
    self.lat_Ob = lat
```

```
    self.lng_Ob = lng
```

```
    self.lat_pos.setText(str(self.lat_Ob))
```

```
    self.long_pos.setText(str(self.lng_Ob))
```

```
    if actC:
```

```
        self.dis=self.distancia(self.lat_Ob, self.lng_Ob, self.lat_La, self.lng_La)
```

```
        self.dis=round(self.dis, 3)* 1000
```

```
        self.long_pos_2.setText(str(self.dis))
```

```
if self.actB:
```

```
    self.paso=self.paso+1
```

```
    print(self.paso)
```

```
    if self.paso > 120:
```

```

        global actV
        actV = True
        self.paso = 0
        self.da = self.da + 1
        dirA = str(self.lat_Ob)+' '+str(self.lng_Ob)
        dirL = str(self.lat_La)+' '+str(self.lng_La)
        global url

url='https://www.google.com.ec/maps/dir/'+dirA+' '+dirL+'/@'+dirA+',21z/data=!4m5!4m4!1m1!4e1!1m0!5i1'
        if self.dis<=10:
            self.actB = False
            global actI
            actI = True
            actV = False

def distancia(self, lat1, lon1, lat2, lon2):
    rad = math.pi / 180
    dlat = lat2 - lat1
    dlon = lon2 - lon1
    R = 6372.795477598
    a = (math.sin(rad * dlat / 2)) ** 2 + math.cos(rad * lat1) * math.cos(rad * lat2) * (
        math.sin(rad * dlon / 2)) ** 2
    dist = 2 * R * math.asin(math.sqrt(a))
    return dist

def iniRuta(self):
    dirA = str(self.lat_Ob)+' '+str(self.lng_Ob)
    dirL = str(self.lat_La)+' '+str(self.lng_La)
    global url

url='https://www.google.com.ec/maps/dir/'+dirA+' '+dirL+'/@'+dirL+',17z/data=!4m5!4m4!1m1!4e1!1m0!5i1'
    QMessageBox.about(self, "AVISO", "Espere muy pronto se mostrara la ruta")
    self.mUbi()

def iniciarNavegacion(self):

    dirA = str(self.lat_Ob)+' '+str(self.lng_Ob)
    dirL = str(self.lat_La)+' '+str(self.lng_La)
    global url

url='https://www.google.com.ec/maps/dir/'+dirA+' '+dirL+'/@'+dirL+',21z/data=!4m5!4m4!1m1!4e1!1m0!5i1'
    self.actB = True
    global actV
    actV = True
    QMessageBox.about(self, "AVISO", "Espere muy pronto se mostrara inciar la
navegacion")

```

```

self.Nave()

def regre3(self):
    self.hide()
    self.principal = Loading()
    self.principal.setWindowTitle('MENU PRINCIPAL')
    self.principal.show()

def mInfo(self):
    self.var1 = Info()
    self.var1.setWindowTitle('Informacion')
    self.var1.show()

def mUbi(self):
    self.var1 = Ubi()
    self.var1.setWindowTitle('Ruta')
    self.var1.show()

def Nave(self):

    self.var1 = Navega()
    self.var1.setWindowTitle('Navegacion')
    self.var1.show()

# *****
# *****
# ***** ruta *****
# *****
class Ubi(QMainWindow):
    def __init__(self):
        super(Ubi, self).__init__()
        loadUi('Ubicacion.ui', self)
        self.btn_regresarU.clicked.connect(self.regreU)
        print('Starting ...')
        display = Display(visible=0, size=(2500, 1000))
        display.start()
        driver = webdriver.Chrome('/usr/lib/chromium-browser/chromedriver')
        global url
        print(url)
        driver.get(url)
        time.sleep(10)
        driver.save_screenshot('datosRuta/ubi.png')
        driver.quit()
        print('target page loaded adnd screenshot taken')
        print('Done')
        img1 = cv2.imread('datosRuta/ubi.png')
        he, wi=img1.shape[0:2]

        img = img1[50:he-50, 410:wi-50]
        cv2.imwrite('datosRuta/mapa.jpg',img)
        img = cv2.imread('datosRuta/mapa.jpg')
        #cv2.imwrite('datosRuta/ubiN.jpg')
        qformat = QImage.Format_Indexed8
        if len(img.shape) == 2:

```

```

        qformat = QImage.Format_Grayscale8
    if len(img.shape) == 3:
        if img.shape[2] == 4:
            qformat = QImage.Format_RGBA8888
        else:
            qformat = QImage.Format_RGB888
    outimg = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)
    outimg = outimg.rgbSwapped()
    self.label.setPixmap(QPixmap.fromImage(outimg))
    self.label.setScaledContents(True)

def regreU(self):
    self.hide()

# *****
# *****
# ***** ruta *****
# *****

class Navega(QMainWindow):
    def __init__(self):
        super(Navega, self).__init__()
        loadUi('rutaDir.ui', self)
        print('Starting ...')
        self.display = Display(visible=0, size=(2500, 1000))
        self.display.start()
        self.driver = webdriver.Chrome('/usr/lib/chromium-browser/chromedriver')
        self.timer2 = QTimer(self)
        self.timer2.timeout.connect(self.graficar)
        self.timer2.start(500)

def graficar(self):
    global actV
    global actI
    if actV:
        actV = False
        print('actualiza')
        global url
        self.driver.get(url)
        time.sleep(10)
        self.driver.save_screenshot('datosRuta/ubi.png')
        print('ruta actualizada')
        img1 = cv2.imread('datosRuta/ubi.png')
        he, wi=img1.shape[0:2]
        img = img1[50:he-50, 410:wi-50]
        cv2.imwrite('datosRuta/mapa.jpg',img)
        img = cv2.imread('datosRuta/mapa.jpg')
        #cv2.imwrite('datosRuta/ubiN.jpg')
        qformat = QImage.Format_Indexed8
        if len(img.shape) == 2:
            qformat = QImage.Format_Grayscale8
        if len(img.shape) == 3:
            if img.shape[2] == 4:
                qformat = QImage.Format_RGBA8888
            else:
                qformat = QImage.Format_RGB888

```

```

        outimg = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)
        outimg = outimg.rgbSwapped()
        self.mapaR.setPixmap(QPixmap.fromImage(outimg))
        self.mapaR.setScaledContents(True)
    if actI:
        actI=False
        self.driver.close()
        QMessageBox.about(self, "AVISO", "!!!LLEGO AL LUGAR ACONTINUACION !!!")
        self.mInfo2()

def mInfo2(self):
    self.hide()
    self.var1 = Info()
    self.var1.setWindowTitle('Informacion')
    self.var1.show()

def regreU(self):
    self.hide()
    self.var1 = mLocal()
    self.var1.setWindowTitle('Menu de Rutas')
    self.var1.show()

# *****

# *****
# *****
# ***** informacion *****
# *****

class Info(QMainWindow):
    def __init__(self):
        super(Info, self).__init__()
        loadUi('ruta.ui', self)
        self.btn_regresar.clicked.connect(self.regre4)
        global foto
        foto = 'lugares/imagenes/'+ str(foto)
        print(foto)
        self.informa.setText(str(informacion))
        img = cv2.imread(foto)
        qformat = QImage.Format_Indexed8
        if len(img.shape) == 2:
            qformat = QImage.Format_Grayscale8
        if len(img.shape) == 3:
            if img.shape[2] == 4:
                qformat = QImage.Format_RGBA8888
            else:
                qformat = QImage.Format_RGB888
        outimg = QImage(img, img.shape[1], img.shape[0], img.strides[0], qformat)
        outimg = outimg.rgbSwapped()
        self.infoIma.setPixmap(QPixmap.fromImage(outimg))
        self.infoIma.setScaledContents(True)

```

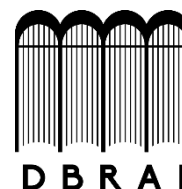
```
def regre4(self):
    self.hide()

# *****

if __name__ == '__main__':
    QApplication.setStyle("fusion")
    app = QApplication(sys.argv)
    window = Loading()
    window.setWindowTitle('MENU PRINCIPAL')
    window.show()
    sys.exit(app.exec())
```



**ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO
DIRECCIÓN DE BIBLIOTECAS Y RECURSOS
PARA EL APRENDIZAJE Y LA
INVESTIGACIÓN**



**UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA**

Fecha de entrega: 26/08/2021

INFORMACIÓN DEL AUTOR/A (S)

Nombres – Apellidos: BERRONES ORTIZ ALVARO ALEJANDRO

CRESPATA ALMACHI CRISTIAN SANTIAGO

INFORMACIÓN INSTITUCIONAL

Facultad: INFORMÁTICA Y ELECTRÓNICA

Carrera: INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

Título a optar: INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

f. Analista de Biblioteca responsable: Ing. CPA. Jhonatan Rodrigo Parreño Uquillas. MBA.



26-08-2021

1133-DBRA-UTP-2021