



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

**“IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA  
ELECTRÓNICO EN UN ANDADOR PARA NIÑOS QUE  
CONTROLE EL ACCESO A LUGARES INSEGUROS, USANDO  
VISIÓN ARTIFICIAL”**

**TRABAJO DE TITULACIÓN**

**TIPO: PROYECTO TÉCNICO**

Presentado para optar al grado académico de:

**INGENIERA EN ELECTRÓNICA, CONTROL Y REDES  
INDUSTRIALES**

**AUTORAS: TATIANA ARACELY SAQUINGA ROBALINO**

**CAROLAENN NATHALIA ANDRADE PERALTA**

**TUTOR: ING. PABLO EDUARDO LOZADA YÁNEZ**

Riobamba – Ecuador

2019

© 2019, Tatiana Aracely Saquina Robalino; Carolaenn Nathalia Andrade Peralta

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

# ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO

## FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

### INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES INDUSTRIALES

El tribunal del trabajo de titulación certifica que: El trabajo de investigación: Proyecto Técnico “IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA ELECTRÓNICO EN UN ANDADOR PARA NIÑOS QUE CONTROLE EL ACCESO A LUGARES INSEGUROS, USANDO VISIÓN ARTIFICIAL” de responsabilidad de las señoritas TATIANA ARACELY SAQUINGA ROBALINO Y CAROLAENN NATHALIA ANDRADE PERALTA, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

Ing. Washington Gilberto Luna Encalada

**FIRMA**

**FECHA**

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

\_\_\_\_\_

Ing. Freddy Enrique Chávez Vázquez

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

\_\_\_\_\_

Ing. Pablo Eduardo Lozada Yáñez

**DIRECTOR DEL TRABAJO DE  
TITULACIÓN**

\_\_\_\_\_

Ing. Jorge Luis Paucar

**MIEMBRO DEL TRIBUNAL**

\_\_\_\_\_

Nosotras, Tatiana Aracely Saquina Robalino Y Carolaenn Nathalia Andrade Peralta, somos responsables de las ideas, doctrinas y resultados expuestos en esta tesis y el patrimonio intelectual de la Tesis de Grado pertenece a la Escuela Superior Politécnica de Chimborazo.

Tatiana Aracely Saquina Robalino

Carolaenn Nathalia Andrade Peralta

## **DEDICATORIA**

Este gran logro se lo dedico a mis padres por ser un ejemplo de perseverancia, por su apoyo incondicional en cada etapa de mi vida, por siempre confiar en mí, y especialmente por el esfuerzo que día a día hicieron para que a mis hermanos y a mí nunca nos falte nada, a mi familia y amigos por la confianza que siempre me tienen.

Tatiana Saquinga

El presente trabajo de titulación se lo dedico principalmente a Dios, por guiar mi camino y permitir cumplir mi meta llenando mi vida de bendiciones, a mis padres Silvia y Eduardo que son un pilar fundamental en mi vida, por la confianza que depositaron en mí y su apoyo incondicional en cada una de mis decisiones, a mi hermanita Chrisbellys que es mi principal motivación para no decaer, a mis Abuelitos por su cariño y palabras de aliento que lograron que jamás retroceda, a mis tíos y primos que han estado presente en cada paso de mi vida, a mis amigos por todas las experiencias vividas.

Carolaenn Andrade

## AGRADECIMIENTO

A Dios por darme las bendiciones para cumplir este gran reto, a mis padres Héctor y Cecilia por el amor y esfuerzo que hicieron para lograr mi objetivo, por ser personas humildes, perseverantes y trabajadoras, por cada consejo que me brindaron para ser una persona llena de valores. A mi hermana Maribel por apoyarme en todos los aspectos y por ser una amiga incondicional, a mi hermano Lincoln por alegrarme y llenarme de entusiasmo día tras día. A mis abuelitas Gloria y Cruz por los consejos y bendiciones a lo largo de mi vida. A mis tíos Mayra, Carlos, Ángel, Dany, Irma, Alfredo, Nelly e Irene por apoyarme y estar presente siempre con una palabra de aliento, a mis primos y demás familiares quienes me acompañaron en este largo camino. A mis amigos Jhonatan, Margarita, Marcos, Freddy, Carlos y José por el apoyo brindado y por cada momento que vivimos juntos durante todo el tiempo que duró la carrera. A Jhonsito mi amigo incondicional por estar siempre en los momentos buenos y malos de mi vida. A mi compañera del trabajo de titulación por la paciencia y dedicación que tuvo para cumplir nuestro propósito, por convertirse en una buena amiga y por ser una persona llena de optimismo.

Tatiana Saquinga

Agradezco a mis padres que son mi fortaleza, especialmente a mi mamita ya que ella es el motor de mi vida, en esos días que daba todo por perdido y quería tirar la toalla ella estuvo ahí apoyándome y dándome las fuerzas necesarias para seguir luchando por mis sueños, por su apoyo incondicional y la confianza que ha depositado en mí a ojos cerrados y sobre todo por formar mi carácter siendo mi ejemplo de mujer dedicada, soñadora y luchadora, a mi pequeña hermanita que me hace sentir orgullosa de lo que he logrado con cada palabra de amor que me da, doy gracias a mis abuelitos que son mi luz, a mi enamorado que con su amor me motivo a seguir adelante en los momentos difíciles, a José y Stalin por las largas noches de estudio donde soñábamos con este día, a mis amigas el club de las incomprendidas por sus consejos y los momentos que compartimos juntos que hicieron que se convirtieran en mi segunda familia, a los Ingenieros Pablo Lozada y Jorge Paucar por compartir con nosotras sus conocimientos siendo un gran aporte para la culminación del trabajo de titulación y un agradecimiento especial a mi compañera y amiga de tesis por su dedicación y esfuerzo, logrando complementarnos con nuestras fortalezas.

Carolaenn Andrade

## TABLA DE CONTENIDO

RESUMEN.....	xviii
SUMMARY .....	xix
INTRODUCCIÓN .....	1

### CAPÍTULO I

<b>1. MARCO TEÓRICO.....</b>	<b>5</b>
<b>1.1. Andadores.....</b>	<b>5</b>
<i>1.1.1. Lugares considerados peligrosos por el uso de andadores.....</i>	<i>6</i>
<i>1.1.2. Funcionalidad de los andadores .....</i>	<i>7</i>
<i>1.1.3. Tipos de andadores .....</i>	<i>8</i>
<b>1.2. Inteligencia artificial.....</b>	<b>9</b>
<b>1.3. Visión artificial.....</b>	<b>9</b>
<i>1.3.1. Elementos de un sistema de visión artificial.....</i>	<i>10</i>
<i>1.3.1.1. Fuente de luz .....</i>	<i>10</i>
<i>1.3.1.2. Sensor de imagen .....</i>	<i>12</i>
<b>1.4. Procesamiento Digital de Imágenes (PDI).....</b>	<b>12</b>
<i>1.4.1. La imagen digital .....</i>	<i>12</i>
<i>1.4.2. Etapas del procesamiento Digital de Imágenes .....</i>	<i>13</i>
<i>1.4.2.1. Captura .....</i>	<i>13</i>
<i>1.4.2.2. Pre-Procesamiento.....</i>	<i>14</i>
<i>1.4.2.3. Segmentación .....</i>	<i>14</i>
<i>1.4.2.4. Extracción de características.....</i>	<i>16</i>
<i>1.4.2.5. Identificación de objeto.....</i>	<i>16</i>
<b>1.5. Entrenamiento de Clasificadores .....</b>	<b>17</b>
<i>1.5.1. Descriptores.....</i>	<i>17</i>
<i>1.5.2. Clasificadores.....</i>	<i>17</i>

<b>1.5.3.</b>	<b><i>Tipos de Aprendizaje Automático</i></b> .....	18
1.5.3.1.	<i>Aprendizaje Supervisado</i> .....	18
1.5.3.2.	<i>Aprendizaje no Supervisado</i> .....	21
<b>1.6.</b>	<b>Hardware</b> .....	22
1.6.1.	<i>Raspberry PI</i> .....	22
1.6.2.	<i>Arduino</i> .....	24
1.6.3.	<i>Sensor de proximidad</i> .....	25
1.6.4.	<i>Batería</i> .....	26
1.6.5.	<i>Actuadores del sistema</i> .....	26
1.6.5.1.	<i>Servomotores</i> .....	27
<b>1.7.</b>	<b>Software Raspbian Jessie</b> .....	27
1.7.1.	<i>Python</i> .....	28
1.7.1.1.	<i>Características del lenguaje de programación Python</i> .....	28
1.7.2.	<i>Open CV</i> .....	29
<b>1.8.</b>	<b>Software Solidworks</b> .....	30

## CAPÍTULO II

<b>2.</b>	<b>MARCO METODOLÓGICO</b> .....	31
2.1.	<b>Requerimientos del sistema</b> .....	31
2.2.	<b>Concepción del diseño de hardware del prototipo de andador</b> .....	31
2.3.	<b>Diseño de las etapas del Prototipo de Andador</b> .....	32
2.3.1.	<i>Diseño de la etapa de Entrenamiento</i> .....	33
2.3.2.	<i>Diseño de la etapa de Adquisición de la Información</i> .....	33
2.3.3.	<i>Diseño de la Etapa de Procesamiento de la Información</i> .....	34
2.3.4.	<i>Diseño de la Etapa de Control</i> .....	35
2.3.5.	<i>Etapa de Alimentación</i> .....	35
2.4.	<b>Selección del Hardware del prototipo</b> .....	36
2.4.1.	<i>Selección del prototipo de andador.</i> .....	36
2.4.2.	<i>Selección del Hardware Raspberry Pi</i> .....	36
2.4.3.	<i>Selección del microcontrolador Arduino</i> .....	38
2.4.4.	<i>Selección del dispositivo de captura</i> .....	39

2.4.5.	<i>Sensor Ultrasónico HC-SR04</i> .....	40
2.4.6.	<i>Buzzer</i> .....	41
2.4.7.	<i>Selección de los servomotores</i> .....	42
2.4.8.	<i>Selección de las fuentes de alimentación</i> .....	46
2.5.	<b>Diagrama de Conexiones</b> .....	48
2.6.	<b>Software del Dispositivo</b> .....	49
2.6.1.	<i>Requerimientos del Software</i> .....	49
2.7.	<b>Software para el Procesamiento Digital de Imágenes y la etapa de Control</b> .....	49
2.7.1.	<i>Programación en el IDE de Python</i> .....	50
2.7.2.	<i>Programación en el IDE de Arduino</i> .....	57

### CAPÍTULO III

3.	<b>ANÁLISIS Y RESULTADOS</b> .....	59
3.1.	<b>Protocolo de pruebas para evaluar el algoritmo de detección de peligros, el sistema electrónico y mecánico.</b> .....	60
3.2.	<b>Prueba de algoritmo de reconocimiento de lugares peligrosos</b> .....	61
3.3.	<b>Prueba de tiempo de reconocimiento</b> .....	74
3.4.	<b>Prueba de tiempo de transmisión de datos</b> .....	76
3.5.	<b>Prueba de frenado</b> .....	77
3.6.	<b>Prueba de tiempo de duración de las baterías</b> .....	80
3.7.	<b>Análisis de costos</b> .....	83

	<b>CONCLUSIONES</b> .....	84
--	---------------------------	----

	<b>RECOMENDACIONES</b> .....	86
--	------------------------------	----

### BIBLIOGRAFÍA

### ANEXOS

## ÍNDICE DE TABLAS

<b>Tabla 1-1:</b>	Comparación de andadores según sus características.....	8
<b>Tabla 2-1:</b>	Tarjetas Controladoras y sus características .....	22
<b>Tabla 3-1:</b>	Comparación entre los tipos de sensores de proximidad.....	26
<b>Tabla 1-2:</b>	Características de modelos Raspberry Pi.....	37
<b>Tabla 2-2:</b>	Características de los modelos de la familia Arduino .....	38
<b>Tabla 3-2:</b>	Especificaciones de los dispositivos de captura .....	39
<b>Tabla 4-2:</b>	Especificaciones del sensor HC-SR04 .....	40
<b>Tabla 5-2:</b>	Características técnicas del Buzzer.....	41
<b>Tabla 6-2:</b>	Características de Servomotores.....	45
<b>Tabla 7-2:</b>	Consumo de voltaje, corriente y potencia .....	46
<b>Tabla 8-2:</b>	Especificaciones técnicas de RPI Powerpack V2.0.....	47
<b>Tabla 9-2:</b>	Especificaciones técnicas de la batería Lipo Turnigy.....	47
<b>Tabla 1-3:</b>	Datos de la detección de tomacorrientes en la mañana .....	61
<b>Tabla 2-3:</b>	Datos de la detección de tomacorrientes en la tarde.....	62
<b>Tabla 3-3:</b>	Datos de la detección de tomacorrientes en la noche .....	62
<b>Tabla 4-3:</b>	Datos de detección de errores de tomacorrientes .....	62
<b>Tabla 5-3:</b>	Detección de tomacorrientes en probabilidades de fracaso.....	63
<b>Tabla 6-3:</b>	Transformación de datos para supuesto de normalidad.....	64
<b>Tabla 7-3:</b>	Estadístico de prueba-Anova.....	65
<b>Tabla 8-3:</b>	Datos de la detección de gradas en la mañana.....	68
<b>Tabla 9-3:</b>	Datos de la detección de gradas en la tarde .....	69
<b>Tabla 10-3:</b>	Datos de la detección de gradas en la noche .....	69
<b>Tabla 11-3:</b>	Datos de detección de gradas .....	69
<b>Tabla 12-3:</b>	Detección de gradas en probabilidades de fracaso .....	70
<b>Tabla 13-3:</b>	Transformación de datos para supuesto de normalidad.....	70

<b>Tabla 14-3:</b>	Estadístico de prueba-Anova.....	71
<b>Tabla 15-3:</b>	Tiempo de reconocimiento de objetos.....	75
<b>Tabla 16-3:</b>	Distancias de frenado .....	77
<b>Tabla 17-3:</b>	Tiempo de duración de la batería RPI .....	80
<b>Tabla 18-3:</b>	Tiempo de duración de la batería Lipo.....	81
<b>Tabla 19-3:</b>	Costos de implementación.....	83

## ÍNDICE DE FIGURAS

<b>Figura 1-1:</b>	a) Escaleras. b) Tomacorrientes.....	7
<b>Figura 2-1:</b>	Iluminación Frontal.....	11
<b>Figura 3-1:</b>	Iluminación Lateral.....	11
<b>Figura 4-1:</b>	Matriz de una Imagen .....	13
<b>Figura 5-1:</b>	Etapas del Procesamiento Digital de Imágenes .....	13
<b>Figura 6-1:</b>	Segmentación orientada a bordes.....	15
<b>Figura 7-1:</b>	Segmentación orientada a regiones.....	16
<b>Figura 8-1:</b>	Algoritmo clasificadores en cascada.....	19
<b>Figura 9-1:</b>	Support Vector Machines.....	19
<b>Figura 10-1:</b>	Local Binary Patterns.....	20
<b>Figura 11-1:</b>	Algoritmo K-NN.....	20
<b>Figura 12-1:</b>	Algoritmo Regresión Lineal .....	21
<b>Figura 13-1:</b>	Estructura árbol de decisión.....	21
<b>Figura 14-1:</b>	Partes de la tarjeta Raspberry Pi3 .....	23
<b>Figura 15-1:</b>	Esquema de Puertos GPIO.....	24
<b>Figura 16-1:</b>	Partes de la tarjeta Arduino.....	25
<b>Figura 17-1:</b>	Configuración de los protocolos de comunicación .....	27
<b>Figura 18-1:</b>	GUI Sistema Operativo Raspbian Jessie.....	28
<b>Figura 19-1:</b>	Estructura de la librería OpenCV.....	29
<b>Figura 1-2:</b>	Concepción general del prototipo .....	32
<b>Figura 2-2:</b>	Etapa de entrenamiento.....	33
<b>Figura 3-2:</b>	Etapa de adquisición de la Información .....	34
<b>Figura 4-2:</b>	Etapa de Procesamiento de la Información.....	34
<b>Figura 5-2:</b>	Etapa de Control .....	35
<b>Figura 6-2:</b>	Etapa de Alimentación .....	35

<b>Figura 7-2:</b>	Andador clásico .....	36
<b>Figura 8-2:</b>	Raspberry Pi3 B .....	38
<b>Figura 9-2:</b>	Arduino Mega2560 .....	39
<b>Figura 10-2:</b>	Cámara Logitech C270 .....	40
<b>Figura 11-2:</b>	Sensor Ultrasónico .....	41
<b>Figura 12-2:</b>	Buzzer .....	41
<b>Figura 13-2:</b>	Diseño mecánico de la llanta. ....	42
<b>Figura 14-2:</b>	Diseño mecánico servo motores. ....	42
<b>Figura 15-2:</b>	Diseño mecánico del sistema de frenado. ....	43
<b>Figura 16-2:</b>	Medición de la fuerza.....	43
<b>Figura 17-2:</b>	Sistema de Fuerza de la polea.....	44
<b>Figura 18-2:</b>	Servo motor MG996R.....	45
<b>Figura 19-2:</b>	a) RPI PowerPack V2.0 b) Batería Lipo Turnigy 2S .....	47
<b>Figura 20-2:</b>	Diagrama de conexiones .....	48
<b>Figura 21-2:</b>	Python y Arduino en Raspbian .....	50
<b>Figura 22-2:</b>	Diagrama de flujo .....	51
<b>Figura 23-2:</b>	Diagrama de flujo para.....	52
<b>Figura 24-2:</b>	Diagrama de flujo para crear el índice positivo. ....	53
<b>Figura 25-2:</b>	Comandos para la creación de los archivos índice.....	54
<b>Figura 26-2:</b>	Archivo índice de imágenes negativas.....	54
<b>Figura 27-2:</b>	Archivo índice de imágenes positivas.....	54
<b>Figura 28-2:</b>	Creación del archivo vectorial de muestras. ....	55
<b>Figura 29-2:</b>	Ejecución del comando traincascade para el entrenamiento.....	55
<b>Figura 30-2:</b>	Diagrama de Flujo para detectar objetos.....	56
<b>Figura 31-2:</b>	Diagrama de Flujo para la etapa de control .....	58
<b>Figura 1-3:</b>	Prototipo final .....	59
<b>Figura 2-3:</b>	Reconocimiento del tomacorriente rango 0-30cm .....	67
<b>Figura 3-3:</b>	Reconocimiento del tomacorriente rango 31-70cm .....	67

<b>Figura 4-3:</b>	Reconocimiento del tomacorriente rango 71-120cm .....	67
<b>Figura 5-3:</b>	Reconocimiento del tomacorriente rango 121-200cm .....	68
<b>Figura 6-3:</b>	Reconocimiento del tomacorriente rango >200cm .....	68
<b>Figura 7-3:</b>	Reconocimiento del identificador de la grada rango 0-30cm .....	73
<b>Figura 8-3:</b>	Reconocimiento del identificador de la grada rango 31-70cm .....	73
<b>Figura 9-3:</b>	Reconocimiento del identificador de la grada rango 71-120cm .....	73
<b>Figura 10-3:</b>	Reconocimiento del identificador de la grada rango >200cm .....	74
<b>Figura 11-3:</b>	Datos mostrados en la consola de Python .....	74
<b>Figura 12-3:</b>	Datos mostrados en la consola de Python .....	77

## ÍNDICE DE GRÁFICOS

<b>Gráfico 1-3:</b>	Normalidad de fallas .....	64
<b>Gráfico 2-3:</b>	Interacción entre distancia respecto al tomacorriente y Luminosidad.....	66
<b>Gráfico 3-3:</b>	Normalidad de fallas .....	70
<b>Gráfico 4-3:</b>	Interacción entre Distancia respecto al tomacorriente y Luminosidad.....	72
<b>Gráfico 5-3:</b>	Tiempo Promedio de reconocimiento de Objetos .....	76
<b>Gráfico 6-3:</b>	Normalidad de distancias de frenado .....	79

## ÍNDICE DE ANEXOS

- ANEXO A:** Datasheet Raspberry Pi3 B
- ANEXO B:** Datasheet Arduino Mega 2560
- ANEXO C:** Datasheet Logitech C270
- ANEXO D:** Datasheet sensor Ultrasónico HC-SR04
- ANEXO E:** Datasheet Servomotor MG996R
- ANEXO F:** Código en Python para lectura de imágenes positivas y negativas
- ANEXO G:** Código en Python para creación de índice negativo
- ANEXO H:** Código en Python para creación de índice positivo
- ANEXO I:** Código en Python para Identificación de Objetos
- ANEXO J:** Código en Arduino para la etapa de control
- ANEXO K:** Distribución t de Student
- ANEXO L:** Diseño de la parte mecánica
- ANEXO M:** Construcción de la parte mecánica
- ANEXO N:** Implementación del sistema electrónico
- ANEXO O:** Pruebas de frenado

## ÍNDICE DE ABREVIATURAS

<b>IA:</b>	Inteligencia Artificial
<b>VA:</b>	Visión Artificial
<b>PDI:</b>	Procesamiento Digital de Imágenes
<b>INAOE:</b>	Instituto Nacional de Astrofísica, Óptica y Electrónica
<b>RAM:</b>	Random Access Memory
<b>VNC:</b>	Virtual Network Computing
<b>CAD:</b>	Diseño Asistido por Computadora
<b>HDMI:</b>	High Definition Multimedia Interface
<b>XML:</b>	Lenguaje de Marca Extensible.
<b>IDE:</b>	Entorno de Desarrollo Integrado
<b>UNE:</b>	Asociación Española de Normalización
<b>SVM:</b>	Support Vector Machines
<b>LBP:</b>	Local Binary Patterns
<b>GPIO:</b>	Entrada/Salida de Propósito General
<b>SO:</b>	Sistema Operativo

## **RESUMEN**

En el presente trabajo de titulación se implementó un prototipo de sistema electrónico en un andador para niños que controle el acceso a lugares inseguros usando visión artificial. El sistema consta de cinco etapas: entrenamiento, adquisición de la información, procesamiento, control y alimentación. En la primera etapa se realizó el entrenamiento de los clasificadores utilizando la técnica Train Cascade para obtener características únicas de los objetos en estudio (tomacorriente y gradas), en la segunda etapa se utilizó un procesador Raspberry Pi3B, el cual recibe el video en tiempo real grabado por una cámara Logitech C270 y el microprocesador Arduino Mega 2560 que recibe los datos enviados por el sensor de distancia HC-SR04, la tercera etapa es la encargada de procesar el video buscando características de los objetos obtenidas en la etapa de entrenamiento, una vez identificado el o los objetos peligrosos entra en funcionamiento la etapa de control, en la cual el Arduino envía señales a los actuadores: servo motor MG996R para impedir el desplazamiento del andador y al Buzzer encargado de emitir una señal de alarma que ponga en alerta al adulto responsable del niño. Finalmente, la quinta etapa alimenta al sistema con una batería RPI Powerpack V2.0 de 3.8A y una batería Lipo de dos celdas a 1A que brindan autonomía al prototipo. Con los datos obtenidos de las pruebas realizadas se determinó que la distancia óptima y el tiempo de reconocimiento de los objetos es de 31 a 120cm y 3.38 milisegundos respectivamente, el tiempo de transmisión de datos es de 4 microsegundos, siendo este un tiempo óptimo para que los actuadores trabajen de inmediato, además se determinó que la autonomía del prototipo es de 2 horas con 6 minutos, cumpliendo con los requerimientos establecidos.

**PALABRAS CLAVE:** <VISIÓN ARTIFICIAL> <NIÑEZ> <ALGORITMOS DE RECONOCIMIENTO> <TÉCNICA TRAIN CASCADE> <ANDADOR PARA NIÑOS> <PREVENCIÓN DE ACCIDENTES> <DETECCIÓN DE OBJETOS> <OBJETOS PELIGROSOS>

## SUMMARY

In the present work of titration a prototype of system in a walker for children was implemented that controls the access to insecure places using artificial vision. The system consists of five stages: training, information acquisition, processing, control and feeding. In the first stage the training of the classifiers was carried out using the Train Cascade technique to obtain unique characteristics of the objects under study (outlet and harrows), in the second stage a Raspberry Pi3B processor was used, which receives the video in real time recorded by a Logitech C270 camera and the Arduino Mega 2560 microprocessor that receives the data sent by the distance sensor HC-SR04, the third stage is in charge of processing the video looking for characteristics of the objects obtained in the training stage, once the dangerous object (s) has been identified, the control stage enters into operation, in which the Arduino sends signals to the actuators: servo motor MG996R to prevent the movement of the walker and the Buzzer in charge of issuing an alarm signal that alerts the adult responsible for the child. Finally, the fifth stage feeds the system with an RPI Powerpack V2.0 battery of 3.8A and Lipo battery of two cells to 1A that give autonomy to the prototype. With the data obtained from the tests carried out, it was determined that the optimal distance and the recognition time of the objects is from 31 to 120cm and 3.8 milliseconds respectively, the data transmission time is of 4 microseconds, this being an optimal time for the actuators work immediately, it was also determined that the autonomy of the prototype is 2 hours with 6 minutes, fulfilling the established requirements.

**KEYWORDS:** <ARTIFICIAL VISION> <CHILDHOOD> <ALGORITHMS OF RECOGNITION> <TRAIN CASCADE TECHNOLOGY > <CHILDREN FOR CHILDREN> <PREVENTION OF ACCIDENTS> <DETECTION OF OBJECTS> <DANGEROUS OBJECTS>

## INTRODUCCIÓN

Ecuador registra una tasa de natalidad de 350000 niños al año según datos emitidos por el INEC en el último censo de Población y Vivienda, los niños a los 7 meses de edad empiezan a usar andadores hasta los 11 meses aproximadamente, siendo una herramienta de ayuda para que los bebés aprendan a caminar, no obstante, en algunos países como Canadá se ha prohibido su uso y comercialización por la inseguridad que presentan (Kara et al., 2013). Al usar este aparato los bebés se desplazan con facilidad alcanzando lugares peligrosos como: escaleras donde se resbalan, lo que a menudo ocasiona fractura de huesos y lesiones de cráneo severas, siendo esta la causa más común de accidentes por el uso de andadores, de la misma forma cuando el niño se desplaza en el hogar se encuentran con tomacorrientes o puertas abiertas lo que incrementa el riesgo de accidentes (OMS y UNICEF, 2012).

Aunque no se cuenta con datos estadísticos de cuantos niños sufren accidentes usando andadores, según la Sociedad Americana de Pediatría los niños que usan estos aparatos tienen el doble de riesgo de sufrir un golpe en la cabeza, doble de posibilidades de fracturas en brazos y piernas, el doble de riesgo de quemaduras y de heridas por objetos punzocortantes y las probabilidades aumentan a cuatro cuando hablamos de caídas por las escaleras (Sociedad Americana de Pediatría, 2018).

En la actualidad se cuenta con andadores comerciales que incorpora topes para inmovilizar el andador, se han encontrado investigaciones en relación a asistentes robóticos móviles orientadas específicamente a personas de la tercera edad o personas que tengan algún tipo de dificultad para desplazarse, ya sea con silla de ruedas o con andadores pero no se ha encontrado trabajos que se relacionen con la seguridad para bebés, así por ejemplo en Barcelona-España se presentó una silla de ruedas y un andador inteligente para personas dependientes (Mínguez, 2009), además existe un estudio realizado en Venezuela de una silla robótica para el control del desplazamiento comandada por la voz (Valero et al., 2012).

A nivel nacional se han realizado diferentes estudios para la movilidad así por ejemplo en la Escuela Superior Politécnica del Ejército (ESPE) existe un prototipo inteligente para el desplazamiento de los adultos mayores (Guamushig y Jerez, 2018), y un bipedestador automático para la movilidad de personas de la tercera edad (Paredes y Toapanta, 2018).

En la Universidad Nacional Autónoma de los Andes (UNIANDES) se registra un trabajo orientado a personas con limitaciones físicas, en la misma ciudad en la Universidad Técnica de Ambato se presenta un sistema de control para la silla de ruedas eléctrica de bipedestación (Bedón, 2017).

En base a los estudios analizados se propone la Implementación de un prototipo de sistema electrónico en un andador para niños, que controle el acceso a lugares inseguros, usando visión artificial, para evitar accidentes que puedan dañar la integridad del infante.

## **FORMULACIÓN DEL PROBLEMA**

¿Cómo implementar un prototipo de sistema electrónico en un andador para niños, que controle el acceso a lugares inseguros, usando visión artificial?

## **JUSTIFICACIÓN TEÓRICA**

Los robots asistentes son prototipos que están destinados a mejorar la calidad de vida de las personas, existen asistentes dedicados al desarrollo motriz, intelectual, visual, auditivo, para discapacidad, terapias de lenguaje y movilización. Al referirse a la movilización existen una serie de investigaciones en sillas de ruedas, bastón, andadores etc., que ayudan a la movilidad y desplazamiento de personas inválidas o que tengan problemas para desplazarse, así por ejemplo encontramos un estudio realizado en Barcelona-España donde se presentó una silla de ruedas guiada por el cerebro y un andador inteligente para personas dependientes o adultos mayores con la finalidad de mejorar su autonomía, la silla es capaz de evitar todos los obstáculos que se encuentran en el camino (Mínguez, 2009).

En Venezuela un grupo de investigadores presento en el año 2012 “control de una silla robótica a través por voz”, este prototipo está basado en un módulo de interpretación de la voz, un

microcontrolador PIC para el control del desplazamiento y posicionamiento del prototipo, la voz es capturada a través de un micrófono el cual envía de forma serial la información al microcontrolador para el control de esta silla (Valero et al., 2012).

En el año 2017 en la Universidad Técnica de Ambato Bedón Luis presentó “*Sistema de control para la movilidad y extensión de una silla de ruedas Eléctrica de bipedestación*” en la misma se ha desarrollado un sistema de control, en la cual paciente puede controlarla de manera fácil, este sistema está basado en una tarjeta de desarrollo Raspberry y en un Joystick con el cual el usuario puede dirigirse a lugar que desee (Bedón, 2017).

En el año 2018 Guamushig y Jerez de la Escuela Superior Politécnica del Ejército (ESPE) presentaron un prototipo inteligente para el desplazamiento autónomo de los adultos mayores con visión reducida y problemas de movilidad, permitiendo una movilidad adecuada y evitando obstáculos como la detección de situaciones peligrosas como gradas (Guamushig y Jerez, 2018).

La mayor parte de las investigaciones están destinadas al adulto mayor o a personas con discapacidad y los aparatos móviles existentes ya sean andadores o sillas de ruedas están hechos con joystick, sensores Kinect u otros elementos por lo que se plantea implementar un sistema electrónico en un andador para niños que controle el acceso a lugares inseguros, usando visión artificial, cuyo sistema determinará si se trata o no de un lugar peligroso para que inmediatamente los actuadores trabajen evitando que el niño alcance dichos lugares que puedan dañar su integridad, la finalidad de implementar el sistema electrónico en un andador para niños es ayudar a padres de familia a tener más confiabilidad en los artefactos que adquieren para que sus hijos aprendan a caminar

## **JUSTIFICACIÓN APLICATIVA**

Gracias a la gran acogida que los andadores han tenido a lo largo del tiempo, por sus múltiples utilidades, nació la idea de crear un prototipo de andador que sea capaz de reconocer lugares peligrosos cuando el niño use este aparato.

El hardware del prototipo, tendrá incorporado en su interior un procesador, tarjeta de desarrollo, servomotores, buzzer, sensor de distancia y una cámara la cual grabará el escenario el tiempo real para que esta información sea analizada por el software del procesador, el mismo

determinará si el lugar corresponde a un lugar peligroso o no, dado que sea un lugar peligroso el procesador enviará datos a la tarjeta de desarrollo que se encargará de la etapa de control, es decir, cuando lleguen las señales a la tarjeta de desarrollo, ésta enviará señales a los actuadores: servomotores para que actúen sobre las llantas evitando el desplazamiento y sobre el buzzer para que emita una señal de alarma y así la madre pueda acudir a verificar el lugar donde se encuentra el infante.

## **OBJETIVO GENERAL**

- Implementar un prototipo de sistema electrónico en un andador para niños, que controle el acceso a lugares inseguros, usando visión artificial.

## **OBJETIVOS ESPECÍFICOS**

- Determinar los lugares peligrosos más comunes que enfrenta el niño cuando se desplaza usando un andador.
- Especificar los requerimientos que debe cumplir el prototipo a implementar.
- Dimensionar y seleccionar el sistema mecánico, software, hardware y lenguaje de programación que son adecuados para implementar un prototipo de andador capaz de resolver los problemas detectados.
- Desarrollar el protocolo de pruebas, bajo condiciones controladas, que permita evaluar el sistema mecánico, electrónico y el algoritmo de detección de peligros.

# CAPÍTULO I

## 1. MARCO TEÓRICO

En el presente capítulo se hace una revisión bibliográfica de los aspectos tomados en cuenta para llevar a cabo el trabajo de investigación. Se analiza conceptos generales acerca de los andadores, lugares peligrosos más comunes donde los niños sufren la mayor cantidad de accidentes, además también hace énfasis en el procesamiento digital de imágenes, donde se describen cada una de sus etapas y técnicas de procesamiento, finalmente se determina el hardware y software necesarios para implementar el prototipo.

### 1.1. Andadores

La marcha es una etapa importante en el desarrollo de los niños en donde el infante logra independizarse obteniendo mayor libertad de movilidad, hay padres quienes consideran que el uso de andadores es beneficioso para los niños debido a la ayuda presentada en la etapa de marcha brindándole mayor confianza para desplazarse y permitiéndole explorar su entorno, no obstante hay otros factores que influyen al momento de usar andadores como son el tiempo para vigilar al infante por parte de los padres y la poca seguridad que estos aparatos brindan al menor lo cual ocasiona diversos accidentes. Existen países donde se ha prohibido el uso de los andadores por los altos índices lesiones que han provocado en los niños, según datos recopilados en 16 centros sanitarios por el Programa de Prevención y Reporte de Lesiones del Hospital de Canadá, se estima un total de 1935 lesiones por el uso de andadores, entre abril de 1990 y abril de 2002. Resultados similares se vieron en el Hospital de Toulouse entre 2003 y 2006. En ellos, el 75% de los accidentes con andadores son caídas por las escaleras, siendo la cabeza el área más afectada (Pérez y Cuevas, 2012). Según datos de Child Prevention Trust, los andadores provocan más accidentes que cualquier otro artículo para niños y en 2002 más de 2350 bebés de Reino Unido, en su mayoría menores de un año, fueron atendidos en urgencias por esta causa (European Commission, 2009).

Su uso es bastante debatido debido a los accidentes que provocan y por tal motivo la Comisión Europea ha establecido normativas para la elaboración de su diseño y reducción de lesiones, exigiendo un control estricto en la seguridad de los andadores. La norma UNE-EN 1273 contiene los requisitos de desempeño, procedimientos de ensayo y requisitos de marcado para fomentar el uso seguro de los andadores de bebés, tiene por objetivo minimizar los accidentes en niños, que ocurren como consecuencia del uso cotidiano y probable mal uso o abuso de andadores (UNE-EN 1273, 2005).

Los padres forman una parte importante en el correcto uso de los andadores es por ello que la norma exige que existan advertencias sobre el correcto uso de los andadores ya que no se recomienda el uso de estos aparatos a niños con más de cierto peso o a niños muy pequeños que no pueden sentarse sin ayuda (European Commission, 2009).

### **1.1.1. Lugares considerados peligrosos por el uso de andadores**

El andador es considerado uno de los principales riesgos de accidentes en el segundo periodo de vida de los niños, ocasionando caídas por las escaleras, fracturas, golpes en la cabeza, quemaduras y en algunas ocasiones hasta la muerte, a pesar de estas consideraciones los padres adoptan al andador como un sistema de cuidado para los niños, pues tienen la sensación de seguridad debido a que el menor se encuentra sentado en un lugar donde no puede salir (Triviño et al., 1997).

A través de la investigación se han encontrado estudios que determinan como estos aparatos benefician o afectan a la integridad del menor dando a conocer los lugares más comunes donde los niños sufren accidentes por el uso de andadores. Según estadísticas recolectas por el Comité de Pediatría Ambulatorio de Argentina estos accidentes tiene una frecuencia de 12% a 50%, donde las lesiones habituales son golpes en la cabeza, fracturas y quemaduras ocasionadas dentro del hogar (Velasguy, 2018).

En una encuesta realizada a madres de 100 niños durante el ingreso a una sala de emergencia del departamento de Pediatría del Hospital de Baghdad se reportó que de 35 a 70% de infantes sufrieron caídas por las escaleras (Figura 1-1 a) provocando fracturas de cráneo (Al-Nouri y Al-Isami, 2006). Los niños a partir de los 4 años empiezan a tener noción de lo que significan los dos agujeros en la pared y que por ellos circula corriente eléctrica, por la curiosidad que les introducen sus dedos en los tomacorrientes (Figura 1-1 b) provocando quemaduras incluso la muerte debido a esto es responsabilidad de los adultos mantenerlos a salvo, en el Hospital

Regional de Ambato en el año 2016 se reportaron casos por quemaduras que ocurren en un 2% en menores de 14 meses (Lojano y Sacta, 2010).



**Figura 1-1:** a) Escaleras. b) Tomacorrientes.

**Realizado por:** Saquina T, Andrade C; 2019

### 1.1.2. *Funcionalidad de los andadores*

Los andadores son una herramienta muy usada, pues los artefactos ayudan con el desarrollo psicomotor, estimulando al bebe a mover sus piernas para caminar. Además otra razón para usar andadores es el apoyo de la cintura y los brazos de mamá, logrando descansar mientras el bebé se encuentra en el andador (Cárdenas et al., 2017).

La marcha es una de las etapas donde el niño para ponerse de pie necesita tener confianza en sí mismo, muchos niños no consiguen pasar del gateo por temor a caerse o por una mala experiencia y es aquí cuando los padres optan por adquirir un andador como un medio de estimulación y motivación para el menor. El sistema tiene sus pros y contras llegando a ser muy controversiales a continuación se enlista algunos de ellos:

#### **Pros**

- Se desplazan hacia cualquier punto de la casa obteniendo más libertad en sus movimientos.
- El andador también es usado como juguete para mantener ocupado y cansado al bebé y así dormirá por más tiempo evitando la somnolencia y la incomodidad del niño, esto debido a la quema de exceso de energía.
- Despierta curiosidad en el niño para explorar todo su entorno.

## Contras

- Los andadores pueden causar accidentes, muchas veces por descuido de los padres los niños acceden a lugares u objetos peligrosos.
- La velocidad que alcanzan es otro factor importante el cual implica accidentes graves de caídas o tropiezos.
- El uso prolongado de andadores puede disminuir la capacidad motora del niño.
- Producen en las piernas del niño un arqueamiento por una incorrecta postura.

### 1.1.3. Tipos de andadores

A nivel comercial existen dos tipos de diseños de andadores, los clásicos que son una especie de silla con una base cuadrada donde se introduce al bebé y los correpasillos donde el niño se encuentra de pie y empuja al coche, a continuación, en la tabla 1-1 se detallan las características de cada uno.

**Tabla 1-1:** Comparación de andadores según sus características.

Clásicos o de asiento	Corre pasillos o de pie
Posee una silla ergonómica	
Contiene 4 ruedas pequeñas para darle movilidad	Cuenta con 2 ruedas
Está rodeado por una estructura	Contiene una barra de apoyo
Son ideales para niños novatos en la etapa de marcha	Es idóneo en niños que logran ponerse de pie y mantener la estabilidad
Ayudan fortaleciendo las piernas y promoviendo la motricidad, desarrollando la actividad motriz de una forma más eficaz	Permite mayor libertad para movilizarse.

**Fuente:** (Ortopedia1 Andadores)

**Realizado por:** Saquinga T, Andrade C; 2019

Una vez analizada la tabla 1-1, se determinó que el andador clásico por su estructura, trata de evitar accidentes y tiene una base superior permitiendo al niño apoyarse y empujar para empezar con sus primeros pasos, además la altura es regulable de acuerdo tamaño de los niños, y poseen

un asiento acolchonado para mayor comodidad del bebé, por lo tanto, se decidió escoger este tipo de andadores para la implementación del sistema.

## **1.2. Inteligencia artificial**

La inteligencia artificial ha interesado tanto a filósofos como a científicos o ingenieros, desde un enfoque ingenieril la IA busca la generación de una propuesta donde un artefacto o sistema automático realice específicas actividades y presenten la capacidad de dar solución de manera inteligente a una tarea desconocida.

Según Nilsson:

*“La Inteligencia Artificial (IA), es una definición muy amplia y un tanto circular, tiene por objeto el estudio del comportamiento inteligente en las máquinas. A su vez, el comportamiento inteligente supone percibir, razonar, aprender, comunicarse y actuar en entornos complejos. Unas de las metas a largo plazo de la IA es el desarrollo de máquinas que puedan hacer todas las cosas igual, o quizá incluso mejor, que los humanos”* (Nilsson, 2001).

Un objetivo que se plantea la inteligencia artificial desde el punto de vista tecnológico es la creación de sistemas automáticos, hasta el momento estas tareas o funciones han sido exclusivas del ser humano, haciendo que estos sistemas se diseñen y construyan de tal manera que tengan capacidades similares o superiores a las del ser humano (Hidalgo, 1996).

## **1.3. Visión artificial**

Tiene como finalidad la interpretación de imágenes a partir de imágenes captadas por sensores de captura ya sea cámaras u otros dispositivos utilizando la potencia de procesamiento y demás atributos de un procesador (Somolinos, 2002).

Fu, Gonzáles y Lee, describen a la visión artificial como un proceso para obtener, caracterizar e interpretar la información de las imágenes capturadas de un mundo tridimensional (Fu et al., 1989).

En el año 2004 Vélez Serrano menciona que:

*“La visión artificial es una disciplina que intenta emular la capacidad que tienen los seres vivos para ver una escena y entenderla. Los problemas que trata suelen consistir en la*

*extracción de cierta información de una escena, que generalmente es capturada en forma de una imagen, para posteriormente tomar algún tipo de decisión” (Vélez et al., 2003).*

De igual forma Pajares y De la Cruz, define a la visión artificial como *“la capacidad de una máquina para ver el mundo que lo rodea, deduciendo la estructura del mundo tridimensional a partir de una imagen bidimensional” (Pajares y De la Cruz, 2002).*

En un sistema de visión artificial la cámara reemplaza al ojo humano y el software para el procesamiento hace las veces de cerebro, de esta forma se trata de reconocer los objetos que se encuentran en el sistema.

### **1.3.1. Elementos de un sistema de visión artificial**

Un sistema de visión artificial cuenta de dos etapas principales; la primera etapa es la adquisición de imágenes, y la segunda etapa es la encargada de procesar las imágenes adquiridas en la primera etapa. A estas etapas la conforman los siguientes elementos:

#### **1.3.1.1. Fuente de luz**

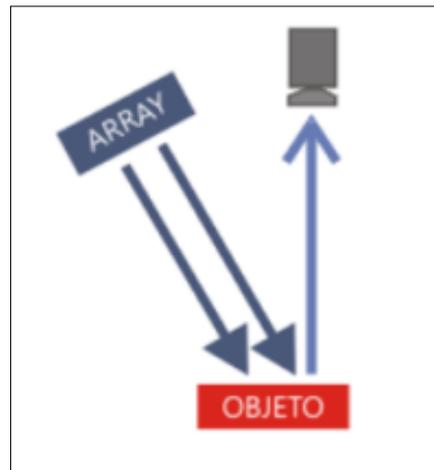
La iluminación es una parte fundamental dentro de la VA, debido a que al tener la iluminación adecuada el sistema tendrá mejor precisión al momento de resolver la aplicación. Las cámaras no solo capturan el objeto sino también las sombras generadas por la luz reflejada en los objetos y esto a su vez genera pérdida de información y productividad del sistema (Unión Europea, 2012). Para reducir este inconveniente existen diversas técnicas de iluminación a continuación se detallan las más comunes:

- **Iluminación frontal:** Esta técnica es la más usada, consiste en colocar la cámara enfocando al objeto hacia la misma dirección de la luz, de esta forma se reducen las imperfecciones y sombras del objeto, la desventaja que presenta es en objetos reflectores provocando demasiado brillo (Unión Europea, 2012). En la figura 2-1 se muestra cómo se aplica la iluminación frontal.



**Figura 2-1:** Iluminación Frontal  
**Realizado por:** (Unión Europea, 2012)

- **Iluminación lateral:** En esta técnica se coloca a la cámara frente al objeto y la luz se encuentra lateral al objeto, esta técnica se aplica para dar realce a ciertas características específicas del objeto (Unión Europea, 2012). En la figura 3-1 se muestra el esquema de la aplicación de la iluminación lateral



**Figura 3-1:** Iluminación Lateral  
**Realizado por:** (Unión Europea, 2012)

- **Iluminación estructura o láser:** Para esta técnica se usa un tipo de luz láser, y es utilizada generalmente para determinar la profundidad de los objetos, este tipo de técnica es más usada en aplicaciones de corte.

### 1.3.1.2. *Sensor de imagen*

Para la adquisición y captura de las imágenes que se van a analizar es necesario usar una cámara electrónica la cual contiene un sensor de imagen que es el encargado de convertir las ondas de luz en señales eléctricas, este sensor está constituido de varios fotositos donde cada uno de estos constituye un píxel (Costa y Fernández, 2005).

El sensor de imagen se elige de acuerdo a la necesidad y requerimientos de cada proyecto y por eso motivo se planteó dos alternativas para la selección de las cámaras; la primera es el módulo de la cámara de Raspberry Pi y la segunda opción es una cámara USB.

## 1.4. **Procesamiento Digital de Imágenes (PDI)**

Son transformaciones que sufre una imagen para obtener una nueva imagen modificada con el objetivo de extraer información para un posterior análisis (Pajares y De la Cruz, 2002).

José Ramón Mejía Vilet describe al PDI como procesos cuyas entradas y salidas son imágenes, éstos procesos se llevan a cabo en un computador y su objetivo es extraer información (contornos, bordes, etc.) de las imágenes incluyendo el reconocimiento de objetos (Mejía, 2004).

### 1.4.1. *La imagen digital*

Es una representación gráfica del mundo real percibida por el sentido de la vista., matemáticamente es una función bidimensional  $f(x, y)$  donde  $x$  e  $y$  son coordenadas en un plano y  $f$  representa la intensidad de la imagen en cualquier par de coordenadas (Mejía, 2004).

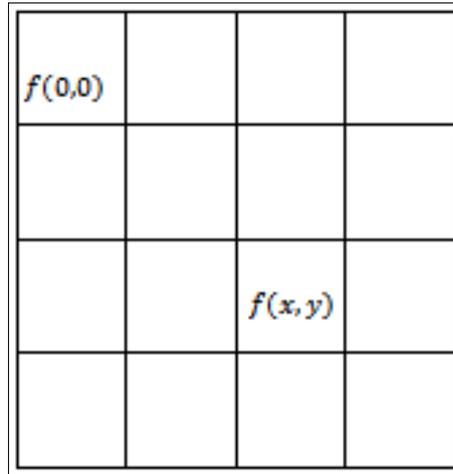
Cuando los elementos  $(x, y, f)$  son finitos entonces se dice que la imagen es digital y se representa por una matriz de  $M$  filas por  $N$  columnas (Ecuación 1-1), la intersección de éstas se denomina píxel que es el elemento básico de visualización de una imagen (Mejía, 2004).

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \dots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \dots & f(1, N - 1) \\ \dots & \dots & \dots & \dots \\ f(M - 1, 0) & f(M - 1, 1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad \text{Ecuación 1-1}$$

Donde  $x$  e  $y$  son variables discretas:  $x = 0, 1, 2, \dots, M - 1$ ;  $y = 0, 1, 2, \dots, N - 1$  y  $f$  el valor equivale a la intensidad de la imagen en dicho punto, usualmente los valores de  $M$  y  $N$  son potencias enteras de 2 (Ecuación 2-1) (Fu et al., 1989).

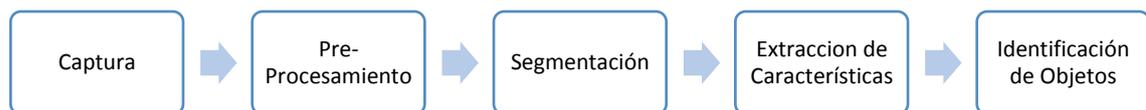
$$M, N = 2^n \quad \text{Ecuación 2-1}$$

En la figura 4-1 se observa que  $f(0,0)$  es el pixel de origen, mientras que  $f(x,y)$  es un pixel que se encuentra en la matriz de la imagen digital.



**Figura 4-1:** Matriz de una Imagen Digital  
**Realizado por:** Saquina T, Andrade C; 2019

#### 1.4.2. Etapas del procesamiento Digital de Imágenes



**Figura 5-1:** Etapas del Procesamiento Digital de Imágenes  
**Realizado por:** Saquina T, Andrade C. 2019.

##### 1.4.2.1. Captura

Es la primera etapa del PDI, donde se requiere capturar las imágenes a través dispositivos físicos como cámara fotográfica, cámara de video o dependiendo del caso también se puede usar sensores para radiografías, ecografías, o rayos X (La Serna y Román, 2014).

Una vez que la imagen es capturada, es importante digitalizarla, es decir convertir una imagen continua del mundo físico a una imagen discreta (Vélez et al., 2003).

#### 1.4.2.2. *Pre-Procesamiento*

El objetivo de esta etapa es preparar a las imágenes a través de operaciones y transformaciones que destaquen características de las mismas, para que en las etapas siguientes las imágenes sean fáciles de tratar. Las operaciones que se hacen en las imágenes sirven como filtros, es decir ingresa una imagen de entrada, se aplican los filtros y se obtiene una señal de salida (Vélez et al., 2003).

Para el pre-procesamiento de imágenes o etapa de mejoramiento se cuenta con las siguientes técnicas:

- Conversión a escala de grises
- Operaciones morfológicas
- Segmentación
- Cambio de tamaño

#### 1.4.2.3. *Segmentación*

Es un proceso en el cual la imagen es dividida en varias partes u objetos con el objetivo de separar las partes necesarias del resto de la imagen, es decir se obtiene nuevas imágenes a partir de la imagen original (La Serna y Román, 2014).

Pajares y De la Cruz mencionan que la segmentación es un proceso en el cual se obtiene información de la imagen, este proceso se basa en principios importantes como discontinuidad orientado a bordes o el de similitud orientado a regiones (Pajares y De la Cruz, 2002).

- **Método de la discontinuidad**

Consiste en dividir la imagen basándose en los cambios de nivel de gris, pues con este cambio se indica la presencia de un borde, este proceso es importante pues proporciona información de los objetos de la imagen para las siguientes etapas, es decir para la extracción de características e identificación de los objetos. Para detectar la presencia o no de bordes se utiliza derivadas de primer orden usando el operador gradiente, mientras que con la derivada de segundo orden se utiliza el operador Laplaciano (La Serna y Román, 2014).

La gradiente de una imagen  $f(x, y)$  en un cierto punto, se define como un vector bidimensional ( Ecuación 5-1 ) (Fu et al., 1989).

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad \text{Ecuación 5-1}$$

El vector  $G$  apunta a la dirección de variación máxima de  $f$  en un punto  $(x, y)$  por lo tanto puede definirse como la ecuación 6-1, con el objetivo de agilizar el procesamiento se aproxima la gradiente a través de los valores absolutos (Ecuación 7-1).

$$G[f(x, y)] = [G_x^2 + G_y^2]^{1/2} = \left[ \left( \frac{\partial f}{\partial x} \right)^2 + \left( \frac{\partial f}{\partial y} \right)^2 \right]^{1/2} \quad \text{Ecuación 6-1}$$

$$G[f(x, y)] = |G_x| + |G_y| \quad \text{Ecuación 7-1}$$

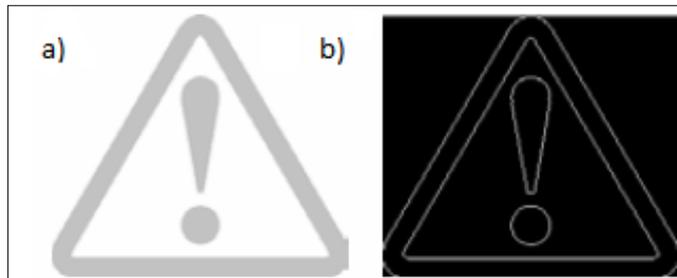
El operador Laplaciano se define como la derivada de segundo orden en un punto  $(x, y)$  (Ecuacion 12-1)

$$L = [f(x, y)] = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{Ecuación 12-1}$$

La mejor forma de encontrar una imagen de salida  $g(x, y)$  es suponer que el valor de  $g$  en dicho punto es igual a la gradiente de la imagen de entrada  $f(x, y)$  es decir:

$$g(x, y) = G([f(x, y)]) \quad \text{Ecuación 13-1}$$

En conclusión, para encontrar el operador gradiente de una imagen digital es necesario extraer las derivadas parciales de la función. Esta es una de las técnicas que más se usan para la detección de bordes, los resultados que se obtienen al aplicar esta técnica se observan en la figura 6-1.



**Figura 6-1: Segmentación orientada a bordes**  
Fuente: (La Serna y Román, 2014)

- **Método de Similitud**

Esta técnica está orientada a regiones, aquí se utilizan propiedades espaciales, es decir que la imagen es dividida en regiones conexas (Figura 7-1) en donde cada región tiene propiedades o características diferentes, este método separa los objetos que se encuentran en una imagen para que sean procesadas de forma independiente (La Serna y Román, 2014).



**Figura 7-1:** Segmentación orientada a regiones  
**Fuente:** (La Serna y Román, 2014)

#### 1.4.2.4. *Extracción de características*

Una vez que la imagen original es dividida en imágenes nuevas es importante extraer las características con alguna información que sean de interés para poder diferenciar los objetos de los otros (La Serna y Román, 2014).

El Dr. Luis Enrique Sucar docente de la Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) manifiesta que para extraer las características de las imágenes se aplican técnicas o descriptores (Sucar y Gómez, 2002, p. 38) basadas en:

- Segmentación basada en la textura, color o forma
- Contornos describiendo las formas que se basan en modelos geométricos.
- Esquinas u orillas

#### 1.4.2.5. *Identificación de objeto*

Se basa en la identificación de un objeto basado en un modelo, de manera que se pueda establecer relación entre el objeto y el modelo en base a la información que se encuentra en el sistema (Pajares y De la Cruz, 2002).

Fu, menciona que es un proceso de etiquetado, es decir, a través de los algoritmos de reconocimiento se debe identificar a los objetos que han sido segmentados de la imagen y asignarle un nombre o etiqueta (Fu et al., 1989).

*“En esta etapa se asigna una etiqueta al conjunto de objetos reconocidos basándose en la información que brindan los descriptores”*(La Serna y Román, 2014).

## **1.5. Entrenamiento de Clasificadores**

Es un proceso que consta de dos pasos: la adquisición de imágenes donde se recopila un conjunto de imágenes positivas y negativas y la creación de muestras, para tener un resultado óptimo en el entrenamiento las imágenes deben ser en blanco y negro para que estas no sean afectadas por la luminosidad del medio, una vez realizado el análisis de las características de las imágenes se clasifica un objeto desconocido a la clase que pertenece (García y Chávez, 2016).

### **1.5.1. Descriptores**

Son procesos cuya función es la extracción de características determinantes que posee una imagen las cuales generan un conjunto de valores que se almacena en los vectores de descriptores, para la toma de estos valores se pretende que las imágenes sean similares lo cual permite que los puntos de interés sean cercanos, otros factores que se consideran son el número de imágenes y la resolución de estas pues al tener sobrecarga en estos valores el vector aumenta su tamaño lo que provoca que al someterlos a un algoritmo de detección su tarea cree resultados erróneos, por esta razón la selección de los descriptores es una parte fundamental para el costo computacional que va tener el algoritmo (Aguilar, 2018, p. 15).

### **1.5.2. Clasificadores**

Machine Learning o conocido también como aprendizaje automático, es una disciplina que inicialmente se derivó de la Inteligencia Artificial pero en la actualidad trabaja de forma independiente, ésta herramienta crea sistemas para que la computadora aprenda de forma automática, es decir que identifique patrones en una infinidad de datos, sin embargo, los que realmente aprenden son algoritmos que son una serie de instrucciones para determinar la

solución de un problema a través de todos los datos almacenados y con ellos son capaces de predecir comportamientos futuros (González, 2014).

Rosario Gago Utrera lo define como un campo de las ciencias de la Computación y una rama de la Inteligencia Artificial cuyo objetivo es hacer que la computadora aprenda, es decir, generalizar comportamientos y conocimientos a partir de información que es suministrada por el usuario (Utrera, 2017).

Los clasificadores se crean con la información proporcionada por los descriptores y mediante el uso de algoritmos de detección basados en Machine Learning, el objetivo de los clasificadores es asignar un objeto a una clase donde las características utilizadas en la fase de entrenamiento tenga similitud con las características del objeto, esto se realiza con ayuda de algoritmos como SVM, LBP y HAAR los cuales reducen los tiempos de análisis y costos computacionales (Aguilar, 2018).

El proceso de clasificación viene dado por una serie de paso independientemente del tipo de clasificador que se usa:

1. Se recopilan muestras de los objetos, y se determinan puntos de interés para el reconocimiento de las características del mismo (vector de descriptores).
2. Se realiza el entrenamiento con el uso de los vectores de descriptores.
3. El clasificador decide a que clase pertenecen objetos que se quiere reconocer.

### **1.5.3. Tipos de Aprendizaje Automático**

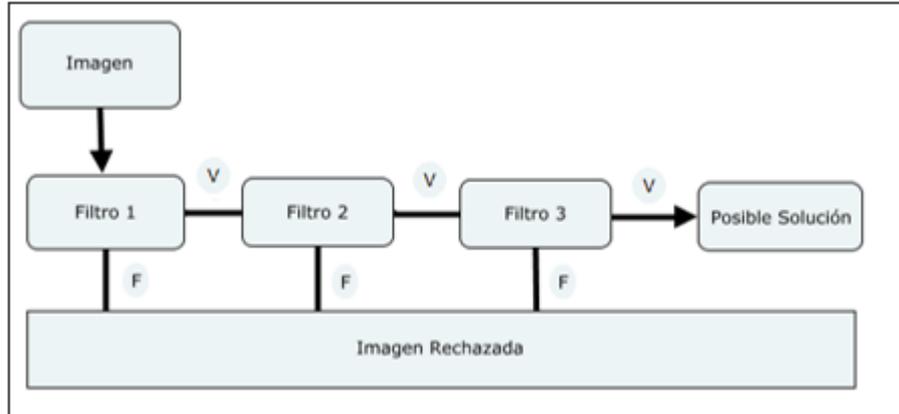
#### **1.5.3.1. Aprendizaje Supervisado**

En este tipo de aprendizaje los algoritmos trabajan con datos que previamente son etiquetados como pertenecientes a dos o más clases, y cuyo objetivo es aprender a predecir la clase de datos que no han sido etiquetados (Utrera, 2017).

Dentro del aprendizaje supervisado encontramos distintos algoritmos que se detallan a continuación:

- **Train Cascade:** Los clasificadores tipo cascada necesitan de un grupo de imágenes que contengan puntos de interés específicos para ser utilizadas como muestras positivas y un

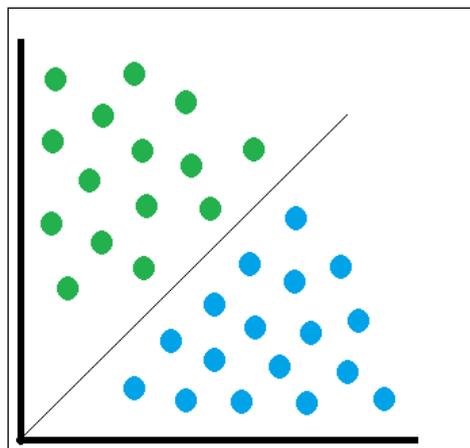
conjunto de muestras negativas donde aparezcan imágenes distintas a las requeridas (Figura 8-1), para la aplicación de este algoritmo se recomienda el uso de imágenes en escala de grises para que el entrenamiento no se vea afectado por factores como la luminosidad (Sanhueza, 2008).



**Figura 8-1:** Algoritmo clasificadores en cascada

Realizado por: Saquinga T, Andrade C; 2019

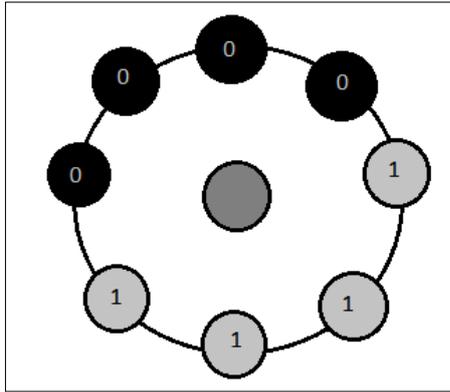
- **Support Vector Machines (VSM):** Por sus siglas en español máquinas de vector de soportes este clasificador se basa en las características por su forma y la apariencia que los objetos presentan, VSM tiene como objetivo encontrar una línea de decisión, es decir encuentra el error mínimo que separa a dos clases (De los Cobos et al., 2000, p. 25).



**Figura 9-1:** Support Vector Machines

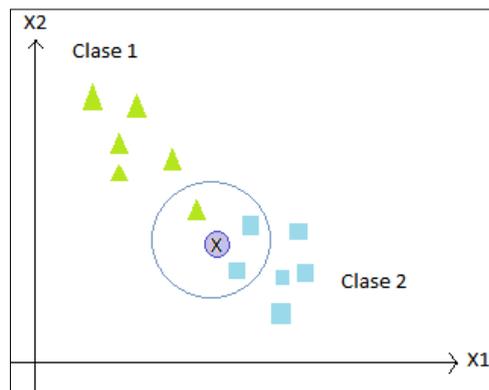
Realizado por: Saquinga T, Andrade C; 2019

- **Local Binary Patterns (LBP):** El entrenamiento de este algoritmo consiste en realizar una división de la imagen en cuadros de algunos pixeles, donde el pixel del centro es comparado con ocho de su alrededor y que se ubiquen a la misma distancia, estos valores se almacenan en un vector (Figura 10-1), si el pixel del centro es mayor se le asigna un '0' y '1' si este es menor (Tobar, 2017, p. 7).



**Figura 10-1:** Local Binary Patterns  
**Realizado por:** Saquinga T, Andrade C; 2019

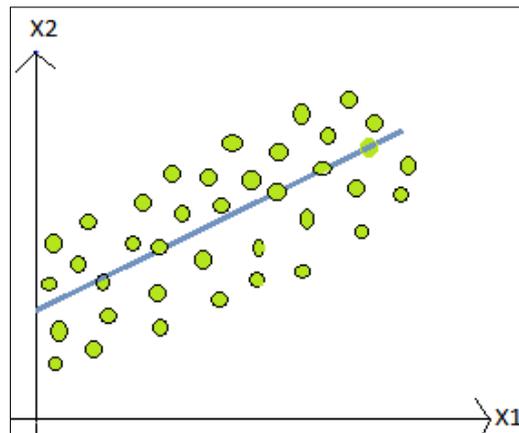
- **Vecinos más cercanos:** También conocido como algoritmo k-NN (k nearest neighbors), este método se basa en el entrenamiento mediante elementos cercanos dentro de un conjunto de entrenamiento y modelos llamados clase, en donde se estima que uno de ellos pertenezca a dicha clase (Manosalvas, 2017). En la figura 11-1 se detalla el algoritmo K-NN.



**Figura 11-1:** Algoritmo K-NN  
**Realizado por:** Saquinga T, Andrade C; 2019

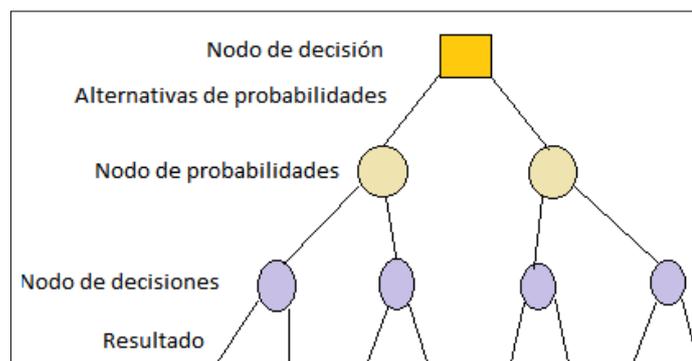
El objetivo de este algoritmo es encontrar los “k” elementos parecidos al elemento de estudio dentro de los modelos, el objeto será etiquetado dependiendo del valor que tome “k”, es decir, la clase que tenga la mayor parte de elementos parecidos al elemento de estudio, será la que le asigne la etiqueta.

- **Regresión Logística:** Es un modelo lineal que se usa para predecir la etiqueta de una variable en función de las variables independiente como se puede apreciar en la figura 12-1 (Utrera, 2017).



**Figura 12-1:** Algoritmo Regresión Lineal  
 Realizado por: Saquinga T, Andrade C; 2019

- **Árboles de decisión:** Es una técnica de búsqueda de datos, basando su funcionamiento en una estructura similar a la de un árbol, los atributos que se van a evaluar son los nodos del árbol, mientras que las hojas almacenan los resultados finales basándose en el estudio de cada nodo como se puede apreciar en la figura 13-1 (Ortuño, 2016).



**Figura 13-1:** Estructura árbol de decisión  
 Realizado por: Saquinga T, Andrade C; 2019

Si tras la construcción del árbol de decisión se tiene un modelo excesivamente grande, la evaluación del mismo involucra el uso de sentencias IF, THEN, ELSE por lo que la aplicación de este tipo de algoritmo equivale a un gran costo computacional debido a la cantidad de parámetros a evaluar (Utrera, 2017).

### 1.5.3.2. Aprendizaje no Supervisado

Los datos para el entrenamiento son un conjunto de vectores sin etiquetas, el objetivo de este tipo de aprendizaje es descubrir grupos similares dentro de los datos (Utrera, 2017).

## 1.6. Hardware

Para procesar las señales emitidas por el dispositivo de captura y el sensor es necesario utilizar una tarjeta que sea capaz de controlar dichas señales, para ello en la tabla 2-1 se analizan las características de las posibles tarjetas que puedan ser usadas.

**Tabla 2-1:** Tarjetas Controladoras y sus características

Características	Tarjetas Controladoras		
	Arduino	Raspberry Pi	Tessel
Procesador	Atmega 2560	BCM2438 64 bits	ARM Cortex-M3
Frecuencia	16MHz	1.2GHz	48Mhz
RAM	8Kb	1 GB	64Mb
Lenguaje de Programación	C	Python	JavaScript
Alimentación	5V	5V	5V
Corriente	40mA	2.5A	12mA
Puertos	14 puertos Analógicos 54 puertos Digitales	40 GPIO	2 puertos módulos de 10 pines

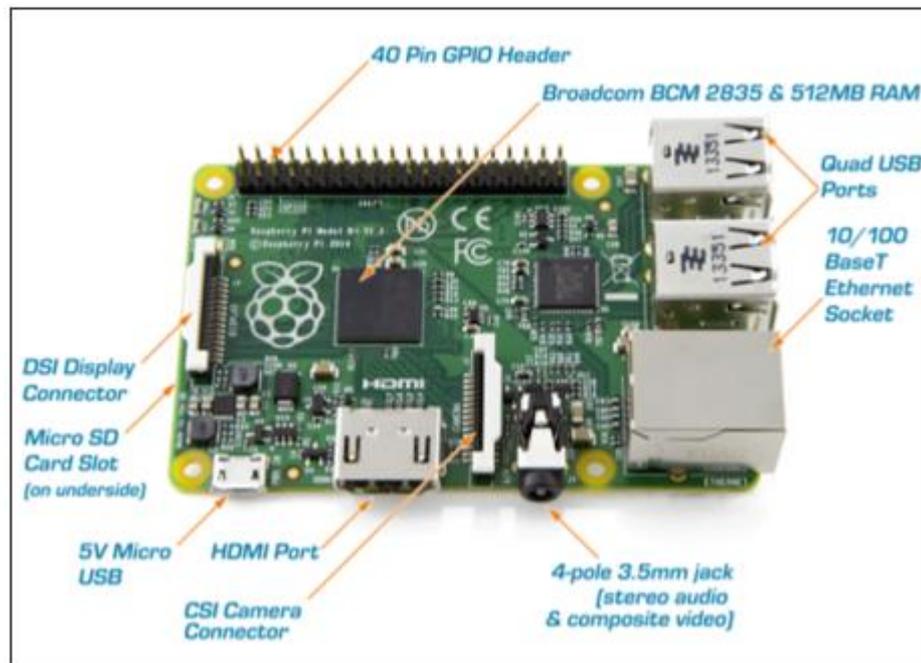
**Fuente:** (Arduino, 2019)(Raspberry Pi, 2019)(Tessel, 2018)

**Realizado por:** Saquina T, Andrade C; 2019

Después de revisar cada una de las características del Arduino, Raspberry Pi y Tessel se opta por trabajar con el Hardware Raspberry por su amplia gama de características, especialmente por su capacidad de procesamiento pues va a satisfacer las necesidades del diseño.

### 1.6.1. Raspberry PI

Es una placa pequeña que en su interior contiene varios elementos que constituyen un computador común, siendo esta tarjeta capaz de comportarse como un ordenador de tamaño reducido, ya que a esta se pueden conectar otros dispositivos tales como: teclado, mouse, cámaras, pantalla entre otros (figura 14-1). Considerando su tamaño otras de las ventajas que posee es la facilidad para transportar y usar, además de su bajo costo y consumo (Solano, 2018).

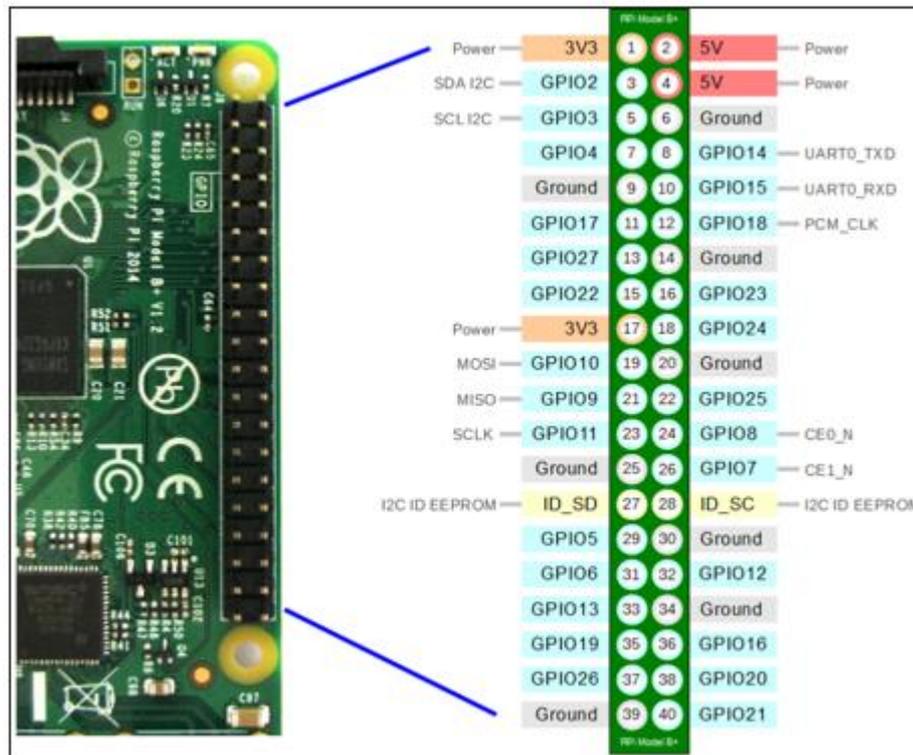


**Figura 14-1:** Partes de la tarjeta Raspberry Pi3

**Fuente:** (Cayambe y Castañeda, 2016)

En la figura 14-1 se detallan como están distribuidos los conectores de la tarjeta Raspberry Pi3 y posteriormente se especifican las funciones de cada uno de ellos:

- **Memory Card Slot:** En este puerto se colocará la tarjeta Micro SD, la cual contiene el sistema operativo instalado.
- **Conector Micro USB:** Es el conectador mediante el cual se alimentará la placa a 5VDC y 1.8A.
- **CSI Camera Serial Interface:** Este puerto es específico para conectar el Módulo de Cámara Pi de la Raspberry.
- **DSI Display Serial Interface:** Permite conectar una pantalla LCD, siendo un puerto serie de alta velocidad.
- **Puerto de Audio mini-jack:** Para la salida de audio
- **Puertos USB:** Posee cuatro puestos USB 2.0, a los cuales se puede conectar cualquier tipo de dispositivo.
- **Puerto Ethenet:** Permite a la placa Raspberry acceder a Internet mediante una red LAN Ethernet.
- **HDMI Port:** Salida de video y audio.
- **Puertos GPIO:** La tarjeta Raspberry cuenta con 40 pines de entrada y salida de propósito general como se muestra en la figura 15-1, estos pines trabajan a 3.3VDC.

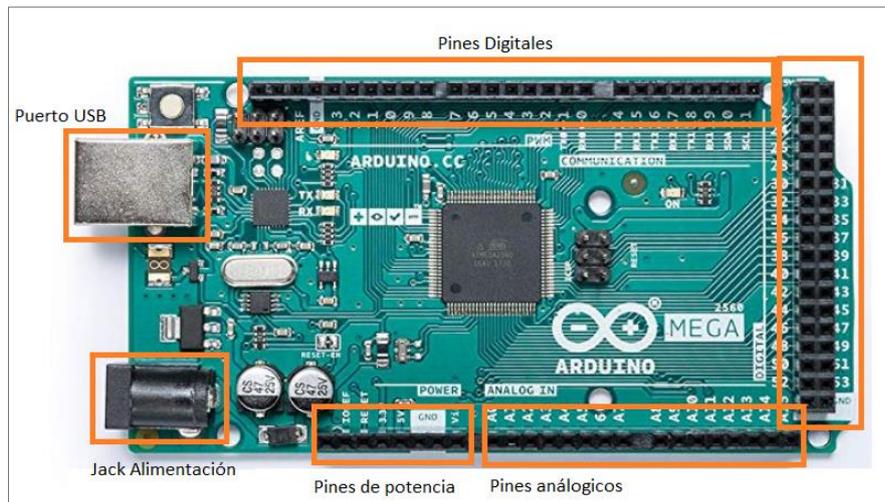


**Figura 15-1:** Esquema de Puertos GPIO  
Fuente: (Raspberry, 2016)

### 1.6.2. *Arduino*

Es una plataforma de código abierto basada en una placa de desarrollo re-programable, su software es de fácil uso debido a que su lenguaje de programación está basado en C++, además es un entorno multiplataforma ya que se puede instalar y ejecutar en varios sistemas operativos como son Windows, Mac Os y Linux. La placa contiene una serie de periféricos que le permiten interactuar con el entorno de acuerdo a las necesidades del programador, también admite la incorporación de nuevas librerías con la finalidad de crear aplicaciones que interactúen con los elementos conectados a la placa (Enríquez, 2009).

En la figura 16-1 se detalla la distribución de los periféricos de Arduino y posteriormente se describe cada uno de ellos.



**Figura 16-1:** Partes de la tarjeta Arduino

**Realizado por:** Saquinga T, Andrade C; 2019

- **Jack Alimentación:** Sirve como fuente de alimentación externa para proporcionar autonomía a la tarjeta, este periférico admite de 7 a 12VDC.
- **Pines de potencia:** Suministran alimentación de una tensión de 5V a circuitos externos a la placa.
- **Puerto USB:** Permite conectar el microcontrolador a una PC, este puerto tiene 3 funciones: transferir los programas del ordenador a la placa, sirve como comunicación y monitor serial para ver los valores de las variables del programa y también alimenta a la tarjeta a 5V.
- **Pines analógicos:** Entradas y salidas analógicas tienen la capacidad de procesar valores en el rango de 0 a 5V.
- **Pines digitales:** Entradas y salidas digitales únicamente pueden procesar voltajes de 5v y 0v.

### 1.6.3. *Sensor de proximidad*

En este proyecto una de las variables más importantes que se debe considerar es la distancia a la que se sitúa un objeto con la finalidad de controlar el espacio temporal al cual debe actuar el sistema, para esto se debe recurrir a la ayuda de un sensor de proximidad que será el encargado de detectar la presencia de algún obstáculo, este tipo de sensores realizan la detección de señales sin necesidad de contacto físico, existen una serie de sensores de proximidad en la tabla 3-1 se realiza una comparación de sus características.

**Tabla 3-1:** Comparación entre los tipos de sensores de proximidad

Características	Sensores de proximidad		
	Capacitivo	Inductivo	Ultrasónico
Material de detección	Metal y no metal	Metal	Independientemente del color y del material
Rango de distancias	1mm a 30mm	1mm a 75mm	Relativamente alto de hasta 8m
Tecnología	Campo Electroestático	Campo Electromagnético	Ondas de luz

Fuente: (Corona et al., 2014)

Realizado por: Saquinga T, Andrade C; 2019

Analizando la tabla 3-1, se optó el sensor ultrasónico puesto que detecta cualquier tipo de material y además su rango de detección de objetos es la necesaria para cumplir con los requerimientos del sistema.

#### **1.6.4. Batería**

Las baterías son elementos encargados de convertir energía química en energía eléctrica, una batería puede contener más de una celda o pila y dos terminales que incorporan los polos positivos y negativos, son usadas para proporcionar autonomía al sistema. Existen diversos tipos de baterías y las que comúnmente se puede encontrar en el mercado son: baterías de plomo ácido, baterías de níquel, baterías de litio (Fonseca, 2011).

Para la correcta selección de una batería que se adapte a los requerimientos del sistema, existen diversos parámetros que se deben de considerar al momento de realizar los cálculos estos son: tensión nominal (V), corriente (A) y potencia (W).

#### **1.6.5. Actuadores del sistema**

Para el sistema de frenado del andador es necesario incorporar actuadores estos pueden ser tipo mecánico, hidráulico o eléctrico de acuerdo a las necesidades del proyecto, para el sistema que

se desea implementar es preciso usar actuadores eléctricos ya que se requiere mantener el control total del movimiento, dentro de este grupo se encuentran los motores de corriente directa, servomotores, electroválvulas entre otros.

#### 1.6.5.1. *Servomotores*

Este es un tipo de motores a los cuales se puede controlar tanto su velocidad como su posicionamiento, los servomotores no giran 360 grados como lo motores comunes, solo giran su eje de 0 a 180 grados y debido a esto sus movimientos son controlados, en su interior están compuestos por un motor de corriente continua y una caja de engranajes la cual permite fijar al motor en una posición determinada y por esa razón son muy usados en la robótica.

### 1.7. **Software Raspbian Jessie**

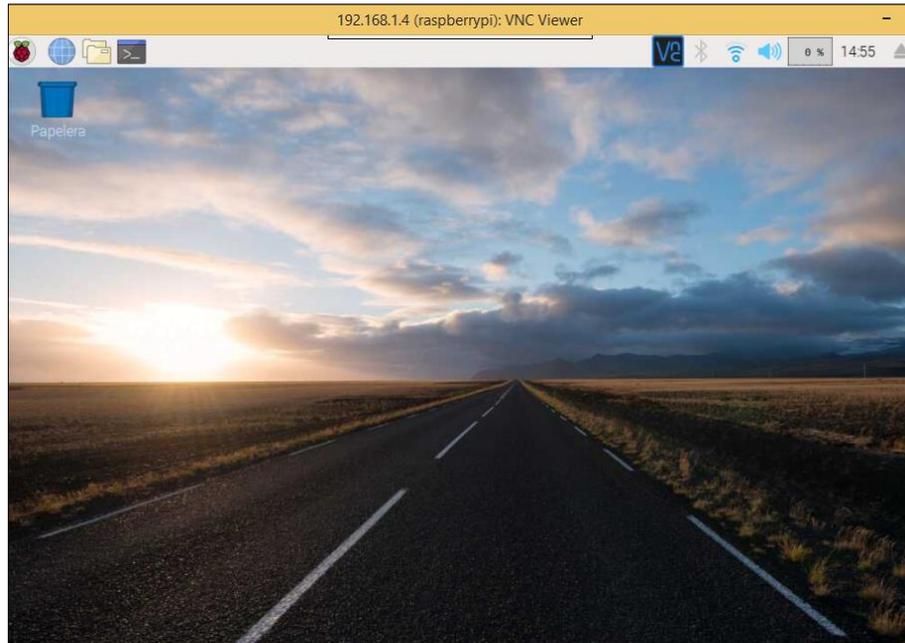
Es un sistema Operativo (SO) amigable con el usuario pues se puede manejar de dos formas, la primera es a través de su consola insertando líneas de códigos y la segunda es usando el entorno grafico del SO, es decir accediendo de forma remota, para esto se hace uso del software VNC Viewer sin necesidad de conectarla a periféricos como pantalla, mouse o teclado, para esto se debe activar el protocolo SSH y VNC, en la figura 17-1 se observa la configuración del dispositivo para acceder de forma remota desde el computador.



**Figura 17-1:** Configuración de los protocolos de comunicación

**Realizado por:** Saquinga T, Andrade C; 2019

Cuando la Raspberry tenga hecha las configuraciones antes mencionadas se puede ingresar a través de VNC Viewer insertando la dirección IP de la Raspberry, para luego ingresar el usuario y contraseña de del dispositivo que será manipulado como se muestra en la figura 18-1.



**Figura 18-1:** GUI Sistema Operativo Raspbian Jessie

**Realizado por:** Saquina T, Andrade C; 2019

El sistema operativo incluye aplicaciones como LibreOffice y Claws Mails para satisfacer las necesidades de los usuarios. Libre Office es similar que Microsoft Office pues posee un procesador de texto, hoja de cálculo y programas para bases de datos, cabe recalcar que todos los archivos generados en las aplicaciones antes mencionadas son compatibles con Microsoft Office (Eltechs, 2017).

### **1.7.1. Python**

Es un lenguaje de programación de alto nivel y fácil de aprender, posee estructuras de datos eficientes, que permiten realizar tareas con pocas líneas de código y de forma clara (Challenger et al., 2014).

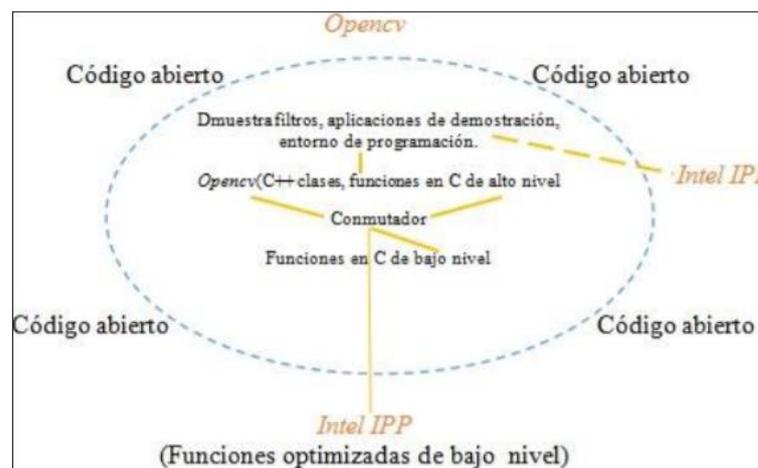
#### *1.7.1.1. Características del lenguaje de programación Python*

- **Lenguaje de propósito general:** Es un lenguaje que no está orientado a un fin concreto, es decir, se puede crear todo tipo de programas.

- **Multiparadigma:** Existen estilos de programación para llegar a un fin deseado, es decir, el usuario puede elegir como va a llevar a cabo la programación ya sea con sentencias de bucle, programación con funciones y métodos o programación orientada a objetos.
- **Lenguaje Interpretado:** En el lenguaje de programación Python existe un intérprete que es el que ejecuta el programa, éste se basa en el código que es escrito por el usuario.
- **Multiplataforma:** Aunque inicialmente Python fue creado para Unix, en la actualidad el lenguaje de programación es compatible con varios sistemas operativos siempre y cuando exista un intérprete programado para Python (Alvarez, 2003).
- **Interactivo:** Cuando el programador ingresa su línea de código, cada sentencia se ejecuta y produce un resultado, lo que hace que el usuario vaya verificando los resultados de la ejecución por porciones de código (Alvarez, 2003).
- **Orientado a Objetos:** Es una gran ventaja puesto que se puede crear programas fáciles y sencillos con componentes reutilizables gracias a las características de la programación orientada a objetos como herencia y polimorfismo (Alvarez, 2003).

### 1.7.2. Open CV

Open Source Computer Vision Library está enfocado principalmente a la visión por computadora siendo una librería basada en aplicaciones en tiempo real para el tratamiento de imágenes. Las áreas de aplicación en las que se enfocan Open CV son: detección y segmentación de imágenes, reconocimiento de gestos faciales, seguimiento del objeto entre otras. Una de las ventajas de Open CV son las diversas interfaces para otros lenguajes y entornos de aplicación con las que cuenta, en la figura 19-1 se detalla la estructura de la librería Open CV (Arévalo et al., 2016).



**Figura 19-1:** Estructura de la librería OpenCV.

Fuente: (Arévalo et al., 2016)

## 1.8. Software Solidworks

Es un software utilizado para el modelamiento de diseños mecánicos en 2D y 3D, es de tipo CAD (diseño asistido por computadora), utiliza un entorno grafico diseñado para el sistema operativo Microsoft Windows, el software realiza animaciones en tiempo real permitiendo tener una idea del desarrollo del producto antes de su fabricación (Fernández y Díaz, 2015).

Solidworks cuenta con tres tipos de diseños que son:

- **Pieza:** Este módulo es el bloque principal, ya que cuenta con los elementos fundamentales para la creación de las figuras en 3D partiendo de planos en 2D, el conjunto de estas piezas creará los ensambles y dibujos.
- **Ensamblaje:** Este módulo está constituido por el conjunto de piezas o subensamblajes que presenta relación entre sí, el ensamblaje permite determinar si existen interferencias o errores entre las piezas creadas para conformarlo.
- **Dibujo:** En este apartado se pueden obtener las vistas o plantillas de las piezas o ensamblajes.

Los tres módulos se encuentran entrelazados entre sí, es decir que al realizar cambios en uno de ellos los otros también sufren cambios en su composición se actualiza de manera automática.

## CAPÍTULO II

### 2. MARCO METODOLÓGICO

En el presente capítulo se detallan las etapas que conforman el sistema y los elementos que son útiles para llevar a cabo el trabajo de titulación además se especifican las características y el diseño del hardware y software del prototipo.

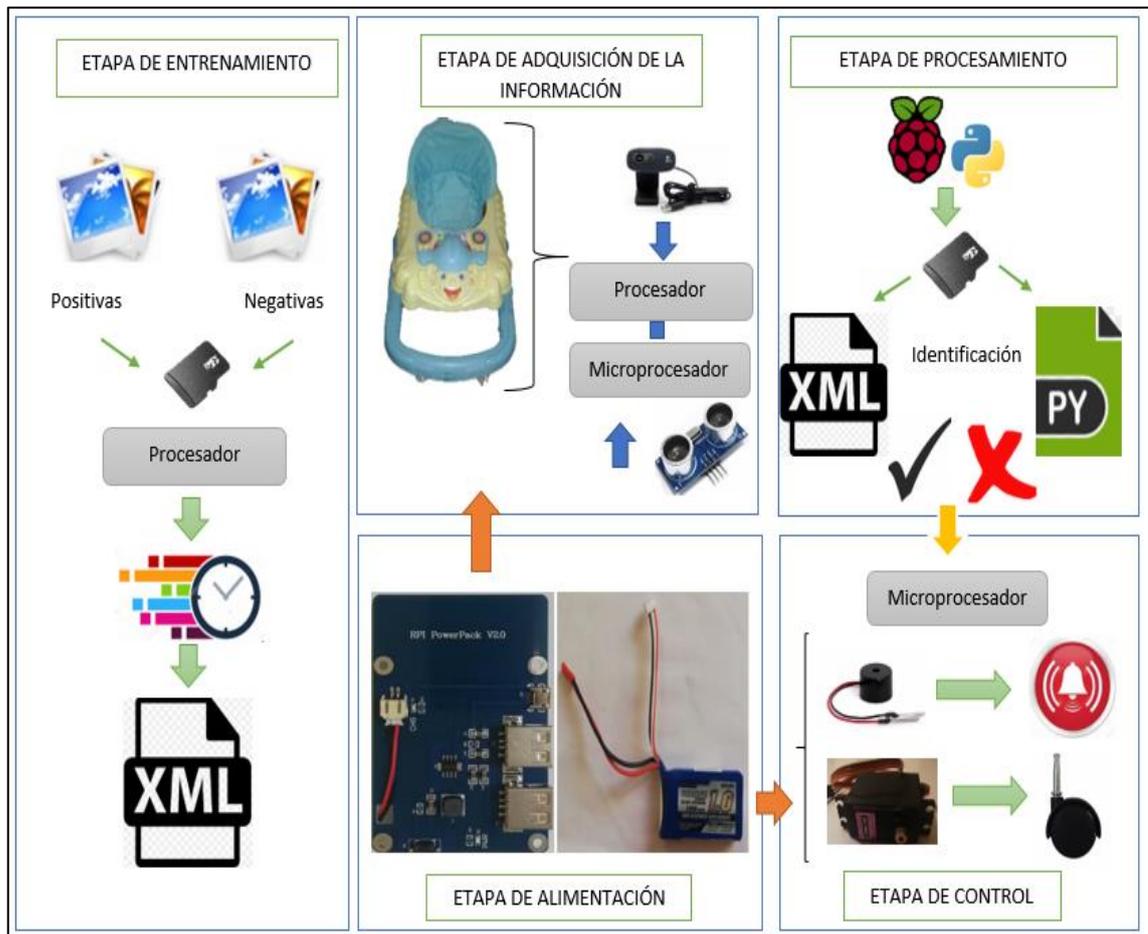
#### 2.1. Requerimientos del sistema

- Seleccionar un procesador que sea eficiente para el procesamiento de imágenes.
- Seleccionar un dispositivo de captura que cuente con buena resolución.
- Diseñar un sistema mecánico de bloqueo para las llantas del andador.
- Seleccionar una fuente de alimentación para que el sistema sea autónomo hasta por 1 hora.

#### 2.2. Concepción del diseño de hardware del prototipo de andador

En la figura 1-2 se observa la concepción general del prototipo, el diseño consta de un procesador y un microcontrolador los cuales recibirán información de los sensores: cámara y sensor de distancia respectivamente, el procesador se usa para la etapa de procesamiento digital de imágenes, para esto se toman fotos de las áreas consideradas peligrosas según datos analizados en capítulo I, y se las almacenan en la tarjeta SD del controlador Raspberry donde se realizan modificaciones a las imágenes para someterlas a un algoritmo de entrenamiento del cual se obtiene un archivo XML (lenguaje de marca extensible) con características únicas del objeto, posteriormente la cámara grabará en tiempo real y el software determinará si los objetos que aparecen en la escena tienen relación con las características obtenidas después del entrenamiento, de ser así el procesador enviará datos al microcontrolador a través del puerto Serial, para que este envíe señales a los actuadores: servomotores que trabajarán sobre las

llantas para bloquearlas, y buzzer el cual emitirá una señal de alarma para que la madre acuda a verificar la posición del andador.



**Figura 1-2:** Concepción general del prototipo

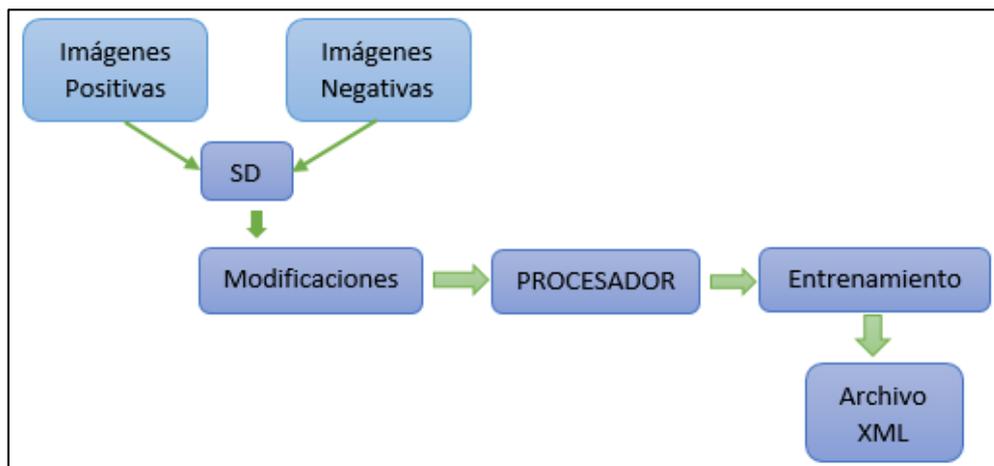
Realizado por: Saquina T, Andrade C; 2018

### 2.3. Diseño de las etapas del Prototipo de Andador

Una vez definido la concepción general del sistema se presenta el diseño de las etapas del prototipo de sistema electrónico.

### 2.3.1. *Diseño de la etapa de Entrenamiento*

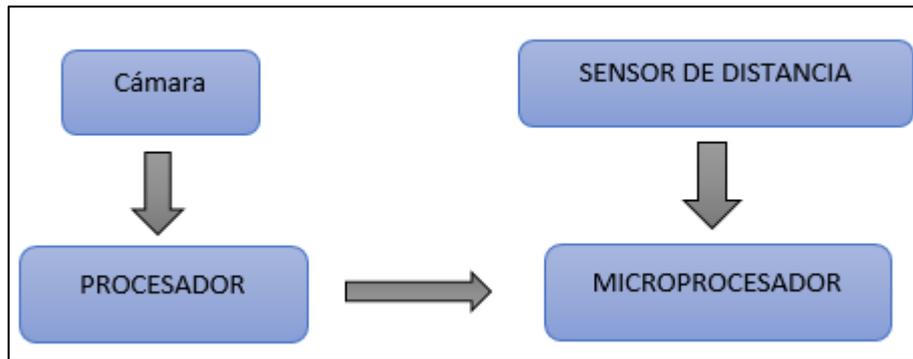
En la figura 2-2 se observa el diagrama de bloques de la etapa de entrenamiento, la cual inicia con la recolección de imágenes positivas (imágenes del objeto en estudio) y negativas (imágenes cualquiera que no contengan el objeto que se estudia) en formato JPG, que son almacenadas en la tarjeta SD del procesador, para llevar a cabo esta etapa se usan aproximadamente 450 imágenes positivas y 900 imágenes negativas que pertenecen al grupo de imágenes para el entrenamiento del clasificador del tomacorriente, de la misma forma se tomaron 833 imágenes positivas y las mismas imágenes negativas del proceso anterior que corresponden al grupo de imágenes para el entrenamiento del clasificador de las gradas, dicho conjunto de imágenes se transforman en tamaño y color con el objetivo de reducir el costo computacional del procesador, cuando las imágenes positivas y negativas hayan sido transformadas pasan al proceso de entrenamiento del cual se obtiene un archivo XML con las características del objeto en estudio, dicha actividad puede tardar de 10 a 12 horas.



**Figura 2-2:** Etapa de entrenamiento  
Realizado por: Saquina T, Andrade C; 2019

### 2.3.2. *Diseño de la etapa de Adquisición de la Información*

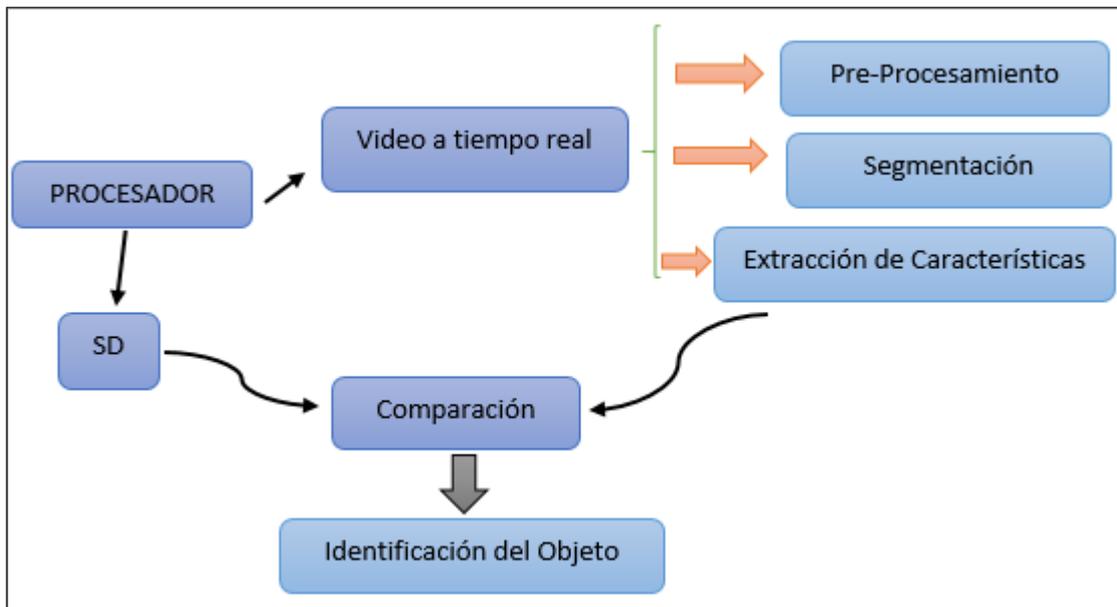
En la figura 3-2 se presenta el diagrama de bloques de la etapa de adquisición de la información, inicia con la captura del video a tiempo real por medio de la cámara que se encuentra conectada en uno de los puertos USB del controlador Raspberry, y a la vez el microprocesador recibirá los datos enviados por el sensor de distancia.



**Figura 3-2:** Etapa de adquisición de la Información  
 Realizado por: Saquina T, Andrade C; 2019

### 2.3.3. *Diseño de la Etapa de Procesamiento de la Información*

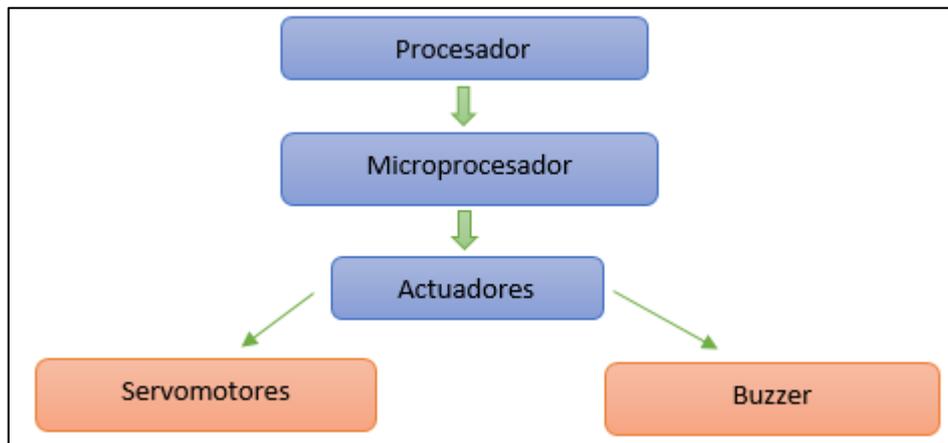
En la figura 4-2 se presenta el diagrama de bloques de la etapa de Procesamiento de la Información, inicia cuando el video es transmitido al procesador para procesarlo, segmentarlo y extraer sus características con las que se determinará si las mismas tienen relación con los datos que se encuentran en el archivo XML generado después del entrenamiento, con el fin de identificar el objeto, una vez hecho este proceso, el controlador determinará si la imagen en análisis corresponde a un lugar peligroso o no.



**Figura 4-2:** Etapa de Procesamiento de la Información  
 Realizado por: Saquina T, Andrade C; 2019

### 2.3.4. *Diseño de la Etapa de Control*

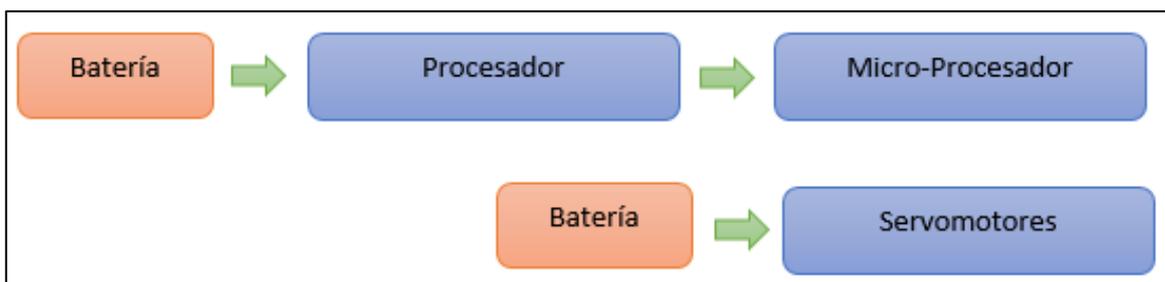
En la figura 5-2 se presenta el diagrama de bloque de la etapa de control, una vez que el procesador determina si el objeto es peligroso o no enviará señales a la tarjeta de desarrollo, la cual se encargará de dar órdenes a los actuadores para emitir una alarma y bloquear las llantas del prototipo evitando daños en la integridad del infante.



**Figura 5-2:** Etapa de Control  
Realizado por: Saquinga T, Andrade C; 2019

### 2.3.5. *Etapa de Alimentación*

En la figura 6-2 se muestra el diagrama de bloques de la etapa de Alimentación, en esta etapa el procesador y el microprocesador serán alimentados por una fuente de tensión portátil de acuerdo a la corriente que consuman, mientras que los servomotores serán alimentados de forma independiente con otra batería.



**Figura 6-2:** Etapa de Alimentación  
Realizado por: Saquinga T, Andrade C; 2019

## 2.4. Selección del Hardware del prototipo

En esta parte se realiza la selección de los elementos que conforman el Hardware del dispositivo.

### 2.4.1. Selección del prototipo de andador.

En el mercado existe una gran variedad de andadores, los mismos poseen diferentes características, sin embargo, en el país se registra que el andador clásico o de asiento (figura 7-2) es el más comercial, el cual es usado en este trabajo de investigación gracias a su amplia gama de particularidades mencionadas en la tabla 1-1 del capítulo anterior, las cuales hacen que el cliente adquiera este producto.



**Figura 7-2:** Andador clásico  
Realizado por: Saquinga T, Andrade C, 2019

### 2.4.2. Selección del Hardware Raspberry Pi

En la Tabla 1-2 se presentan las características de los últimos modelos lanzados por la fundación Raspberry Pi, siendo éstos: Raspberry Pi2 B+, Raspberry Pi3 B, y Raspberry Pi3 B+, con las mismas se hace una comparación para posteriormente seleccionar el Hardware adecuado.

**Tabla 1-2:** Características de modelos Raspberry Pi

<b>Características</b>	<b>Raspberry Pi 2 B+</b>	<b>Raspberry Pi3 B</b>	<b>Raspberry Pi3 B+</b>
<b>Procesador</b>	BCM 2836 ARM v7 32 bits	BCM2438 64 bits	BCM2837B0 ARM v8 64 Bits
<b>Frecuencia</b>	900Mhz	1.2GHz	1.4GHz
<b>Memoria RAM</b>	1GB	1GB	1 GB
<b>Conectividad Inalámbrica</b>	IEE 802.11B	IEEE 802.11b, Bluetooth 4.1	IEEE 802.11b, Bluetooth 4.2
<b>Conectividad de Red</b>	Ethernet 100Mbps	Gigabit Ethernet USB 2.0	Gigabit Ethernet USB 2.0 (300Mbps)
<b>Puertos USB 2.0</b>	4	4	4
<b>Salidas de Video</b>	DSI (Pantalla Táctil) HDMI	DSI (Pantalla Táctil) HDMI	DSI (Pantalla Táctil) HDMI
<b>Almacenamiento</b>	Tarjeta SD	Tarjeta SD	Tarjeta SD
<b>Periféricos de Bajo Nivel</b>	40 x GPIO	40 x GPIO	40 x GPIO
<b>Corriente</b>	1.5A	2.5A	2.5A
<b>Alimentación</b>	5V	5V	5V
<b>Dimensiones</b>	8.5x5.35cm	8.5x5.35cm	8.5x5.35cm

**Fuente:** (Raspberry Pi, 2019)

**Realizado por:** Saquina T, Andrade C; 2019

Con la tabla 1-2 se pudo deducir que los últimos modelos de Raspberry Pi es decir la Raspberry Pi3 B (Figura 8-2) y la Raspberry Pi3 B+ presentan similares y mejores características, excepto en la frecuencia de trabajo ya que el primer modelo tiene una frecuencia de trabajo de 1.2GHz, mientras que en el segundo modelo existe una leve diferencia pues éste trabaja a 1.4GHz de frecuencia, además también se observa que existe diferencia en la versión del módulo bluetooth, sin embargo, esta característica no se toma en cuenta para la elección del dispositivo pues no se usa dicho complemento en el trabajo de titulación. Por lo mencionado anteriormente se concluye que los dos tipos de Raspberry resultan ser efectivas ya que la frecuencia de trabajo de

los dos dispositivos es suficiente para cumplir con los requerimientos establecidos. En el anexo A se aprecia el datasheet de la Raspberry seleccionada.



**Figura 8-2:** Raspberry Pi3 B  
Realizado por: Saquina T, Andrade C; 2019

#### 2.4.3. Selección del microcontrolador Arduino

En la tabla 2-2 se observan las características de los modelos más conocidos de la familia Arduino (Arduino Nano, Arduino Uno y Arduino Mega2560), con las cuales se puede hacer una selección del hardware necesario.

**Tabla 2-2:** Características de los modelos de la familia Arduino

Características	Arduino NANO	Arduino UNO	Arduino Mega 2560
<b>Microcontrolador</b>	Atmega328	Atmega328P-PU	Atmega 2560
<b>Voltaje de Operación</b>	5V	5V	5V
<b>Entradas y salidas digitales</b>	14(6PWM)	14 (5PWM)	54 (15 PWM)
<b>Entradas Analógicas</b>	8	6	16
<b>Corriente DC2IO/Pin</b>	40mA	40mA	20mA
<b>Frecuencia</b>	16MHz	16MHz	16MHz
<b>Memoria</b>	16KB	32KB	256KB

**Fuente:** (Arduino Products, 2019)  
**Realizado por:** Saquina T, Andrade C; 2019

Con los datos de la tabla 2-2 se deduce que los tres tipos de Arduino trabajan con la misma frecuencia y son óptimos para cumplir con los requerimientos, sin embargo, el microcontrolador Arduino Mega 2560 (Figura 9-2) posee mayor memoria que los otros productos Arduino y por tal motivo es el escogido, las especificaciones técnicas completas se observa en el anexo B.



**Figura 9-2: Arduino Mega2560**  
**Realizado por:** Saquina T, Andrade C; 2019

#### 2.4.4. Selección del dispositivo de captura

En la Tabla 3-2 se observa las características de las cámaras web Logitech C170 y C270, pues con estos detalles se puede elegir la que mejor nos convenga.

**Tabla 3-2:** Especificaciones de los dispositivos de captura

Características	Logitech C170	Logitech C270
<b>Captura de Video</b>	640x480 pixeles	HD (1280x720) pixeles
<b>Captura de Imágenes</b>	3MP	3 MP
<b>Conexión</b>	Cable de 1.5m al puerto USB	Cable de 1.5m al puerto USB
<b>Voltaje</b>	5V	5V
<b>Corriente</b>	500mA	500mA
<b>Campo Visual</b>	60°	60°

**Fuente:** (Logitech C170, 2011)

**Realizado por:** Saquina T, Andrade C; 2019

Según las especificaciones la tabla 3-2 se elige la cámara Logitech C270 (figura 10-2) puesto que posee una resolución de video más alta que la cámara Logitech C170, mientras que en el anexo C se puede ver las especificaciones técnicas completas.



**Figura 10-2: Cámara Logitech C270**  
Realizado por: Saquina T, Andrade C; 2019

#### 2.4.5. Sensor Ultrasónico HC-SR04

Su función principal es medir distancia, en el anexo D se muestra las características técnicas completas del sensor HC-SR04, es muy utilizado en diversos proyectos pues posee las siguientes características:

- Bajo costo.
- Tamaño pequeño.
- Buena precisión.
- Bajo consumo energético

En la tabla 4-2 se observa las especificaciones de dicho sensor.

**Tabla 4-2:** Especificaciones del sensor HC-SR04

Características	HC-SR04
Voltaje de operación	5V
Corriente de trabajo	15mA
Rango de medida	2cm-400cm
Precisión	$\pm 3mm$
Frecuencia de ultrasonido	40KHz

Fuente: (Elec Freaks, 2019)  
Realizado por: Saquina T, Andrade C; 2019

En la figura 11-2 se muestra el sensor utilizado en este proyecto.



**Figura 11-2:** Sensor Ultrasónico  
Realizado por: Saquinga T, Andrade C; 2019

#### 2.4.6. *Buzzer*

Es un dispositivo electrónico que produce sonido o zumbido, ideal para emitir alarmas, en la tabla 5-2 se observan las características técnicas de este dispositivo.

**Tabla 5-2:** Características técnicas del Buzzer

Características	Valor
Voltaje	3-5V
Corriente de consumo	25mA
Impedancia	16Ω
Frecuencia	2.7KHz

Fuente: (ABC Components LTD, 2016)  
Realizado por: Saquinga T, Andrade C; 2019

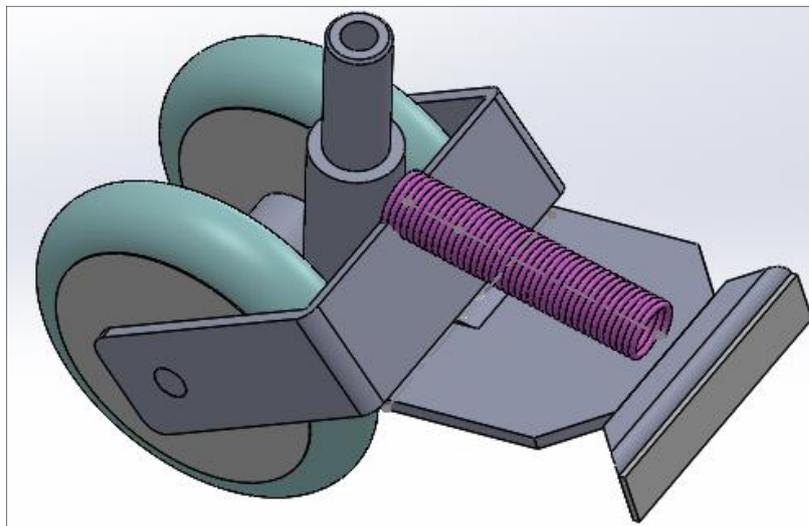


**Figura 12-2:** Buzzer  
Realizado por: Saquinga T, Andrade C; 2019

#### 2.4.7. Selección de los servomotores

Antes de seleccionar los servomotores es necesario detallar el sistema mecánico diseñado para el sistema de frenado, en la figura 13-2 se observa el diseño de la parte de las llantas el cual consta de:

- **Resorte:** se colocó con la finalidad que la zapata retorne a la posición inicial cuando el motor deje de ser accionado.
- **Zapata:** obstruye la rotación de la llanta
- **Cable:** cuya función es alzar o bajar la zapata.

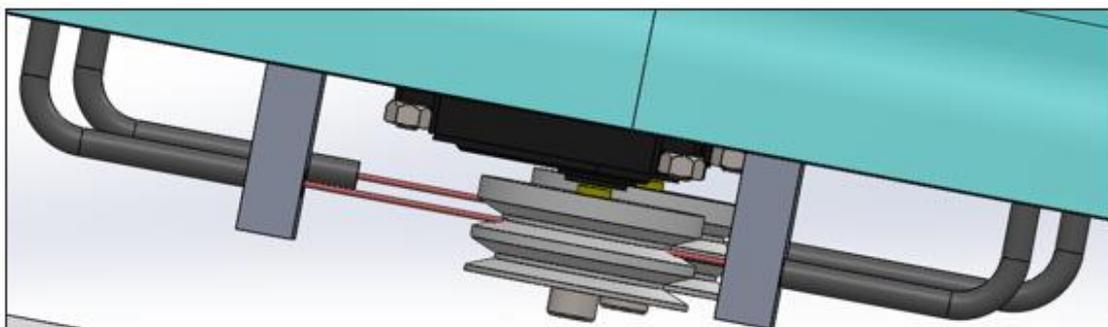


**Figura 13-2:** Diseño mecánico de la llanta.

Realizado por: Saquina T, Andrade C; 2019

Mientras que en la figura 14-2 se observa el diseño mecánico para la parte de los servos motores, este diseño está compuesto de:

- **Polea:** para el enrollamiento del cable
- **Servo:** su objetivo es girar para tensionar el cable.



**Figura 14-2:** Diseño mecánico servo motores.

Realizado por: Saquina T, Andrade C; 2019

Finalmente, el diseño mecánico completo se puede observar en la figura 15-2.



**Figura 15-2:** Diseño mecánico del sistema de frenado.

**Realizado por:** Saquinga T, Andrade C; 2019

Una vez que se diseñó el mecanismo, se procede a elegir los servomotores, para ello se debe analizar la tensión requerida en el cable a través de un dinamómetro, mediante un ensayo práctico se encontró la fuerza requerida para cada rueda igual a 3lbf (Figura 16-2).



**Figura 16-2:** Medición de la fuerza

**Realizado por:** Saquinga T, Andrade C; 2019

Luego de encontrar la fuerza requerida es necesario calcular el torque o momento de torsión que es la capacidad que tiene una fuerza para hacer girar un objeto, cuando éste actúa sobre el (Tippens, 2009, p. 123). Con la ecuación 1-2 se puede encontrar el valor del momento de torsión de un objeto, cuya unidad es Newton por metro ( $N \cdot m$ )

$$T = Fxr \quad \text{Ecuación 1-2}$$

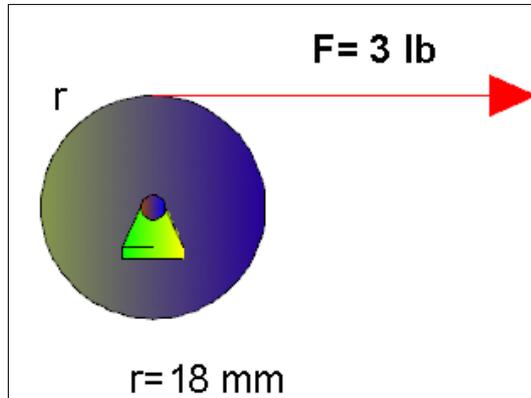
Donde:

T=Torque

F=Fuerza

r=distancia

La figura 17-2 representa el sistema de fuerzas de la polea.



**Figura 17-2:** Sistema de Fuerza de la polea  
Realizado por: Saquina T, Andrade C; 2019

Aplicando la ecuación 1-2 se determina el Torque necesario para el sistema.

Trabajando en el sistema internacional se tiene que:

$$F = 3\text{ lbf} \left( \frac{4.448\text{ N}}{1\text{ lbf}} \right) = 13.34\text{ N}$$

$$r = 18\text{ mm} \left( \frac{1\text{ m}}{1000\text{ mm}} \right) = 0.018\text{ m}$$

$$T = Fxr$$

$$T = (13.34 * 0.018) = 0.24\text{ Nm}$$

Cabe mencionar que para encontrar el torque total necesario se multiplica por 2 puesto que un par de llantas comparten el mismo servo motor.

$$T = 0.24 \times 2 = 0.48\text{ Nm}$$

En la tabla 6-2 se puede observar las características de los servomotores que pueden ser usados en el sistema mecánico.

**Tabla 6-2:** Características de Servomotores

Características		SG-90	MG955	MG 996R
Peso		13.4g	55g	55g
Torque	4.8V	1.8Kg.cm	8.5Kg.cm	9.4Kg.cm
	6V	2.2Kg.cm	10Kg.cm	11Kg.cm
Voltaje de Operación		4.8V-6V	4.8V-7.2V	4.8V-7.2V
Corriente		900mA	900mA	900mA
Rotación		120°	180°	180°
Velocidad de Giro	4.8V	0.1s/60°	0.2 s/60°	0.17s/60°
	6V	0.08s/60°	0.16s/60°	0.14s/60°

**Fuente:** (Electrónicos Caldas MG90S, 2019) (Electrónicos Caldas MG995, 2019) (Electrónicos Caldas MG996R, 2019)  
**Realizado por:** Saquinga T, Andrade C; 2019

Como se observa en la tabla 6-2 el servo SG90 nos brinda un torque de 1.8Kg.cm o 0.1764Nm, mientras que el torque que nos ofrece el servo motor Mg955 es 8.5Kg.cm equivalente a 0.833Nm, y el servo MG996R nos brinda un toque de 9.4Kg.cm o 0.9212Nm, como ya se analizó anteriormente el torque necesario para el sistema mecánico del andador es 0.48Nm lo que significa que los 2 últimos servomotores analizados son suficientes para poder tensionar el cable y por ende obstruir la rotación de la llanta, sin embargo se elige al servo motor MG996R (Figura 18-2) por la velocidad de giro pues es una versión mejorada del servo motor MG995. En el anexo E se puede observar las características técnicas del servo motor MG996R.



**Figura 18-2:** Servo motor MG996R

**Realizado por:** Saquinga T, Andrade C; 2019

#### 2.4.8. Selección de las fuentes de alimentación

En la tabla 7-2 se muestra los valores de voltaje, corriente y potencia de cada uno de los dispositivos que se usan para la implementación del sistema electrónico con los cuales se puede determinar las fuentes de energía necesaria para que el sistema tenga autonomía.

**Tabla 7-2:** Consumo de voltaje, corriente y potencia

Dispositivos	Voltaje (V)	Corriente (A)	Potencia(W)
Raspberry	5	0.800	4
Arduino Mega 2560	5	0.2	1
Logitech C270	5	0.5	2.5
HC-SR04	5	0.015	0.075
Buzzer	5	0.025	0.125
Servos motores	5	0.4	2

Realizado por: Saquinga T, Andrade C; 2019

Según la tabla 7-2 el sistema electrónico consumirá 1.54A (Raspberry, Arduino, Cámara, HC-SR04 y Buzzer) por lo que se seleccionó la batería RPI Powerpack V2.0 (figura 19-2 a), pues es una batería destinada específicamente para Raspberry Pi, cuya capacidad es de 3800mA, mientras que la parte de servo motores consumirá 0.4A por lo que se eligió una batería Lipo de 2 celdas, es decir, 7.4V a 1A (Figura 19-2 b). Aplicando la ecuación 2-2 se obtiene el tiempo de autonomía para la parte electrónica y para los servos motores.

$$Autonomía = \frac{Capacidad\ de\ la\ batería}{I\ control} \quad \text{Ecuación 2-2}$$

$$Aut_{electrónica} = \frac{3.8Ah}{1.54A} = 2.46h = 2h27m$$

$$Aut_{servos} = \frac{1Ah}{0.4A} = 2.5h = 2h30m$$

En la tabla 8-2 se observan las especificaciones técnicas de la batería para la parte electrónica y en la tabla 9-2 se encuentran las especificaciones técnicas de la batería Lipo para los servos.

**Tabla 8-2:** Especificaciones técnicas de RPI Powerpack V2.0

<b>Especificaciones técnicas de RPI Powerpack V2.0</b>	
<b>Voltaje</b>	5V
<b>Capacidad</b>	3800mA
<b>Corriente de carga</b>	1A
<b>Corriente de descarga máxima</b>	1.8A
<b>Salidas USB</b>	2

Fuente: (Altronics, 2019)

Realizado por: Saquina T, Andrade C; 2019

**Tabla 9-2:** Especificaciones técnicas de la batería Lipo Turnigy

<b>Especificaciones técnicas de batería Lipo Turnigy</b>	
<b>Celdas</b>	2
<b>Voltaje</b>	7.4V
<b>Capacidad</b>	1000mA
<b>Tasa de descarga</b>	20-30C
<b>Conector de Carga</b>	JST-XH
<b>Conector de descarga</b>	XT60

Fuente: (AV Electronics, 2019)

Realizado por: Saquina T, Andrade C; 2019

En la figura 19-2 se observa las baterías seleccionadas.



**Figura 19-2:** a) RPI PowerPack V2.0 b) Batería Lipo Turnigy 2S

Realizado por: Saquina T, Andrade C; 2019



- El sensor ultrasónico HC-SR04 es alimentado con Vcc y Gnd del Arduino, la señal del Trigger se encuentra en pin 10 y la señal del echo se encuentra en el pin 9.
- El Buzzer para la señal de alarma está conectado a tierra mientras que la señal se encuentra en el pin digital 2 del Arduino.
- Finalmente se conecta la Raspberry y el Arduino a la batería RPI Powerpack v2.0.

## **2.6. Software del Dispositivo**

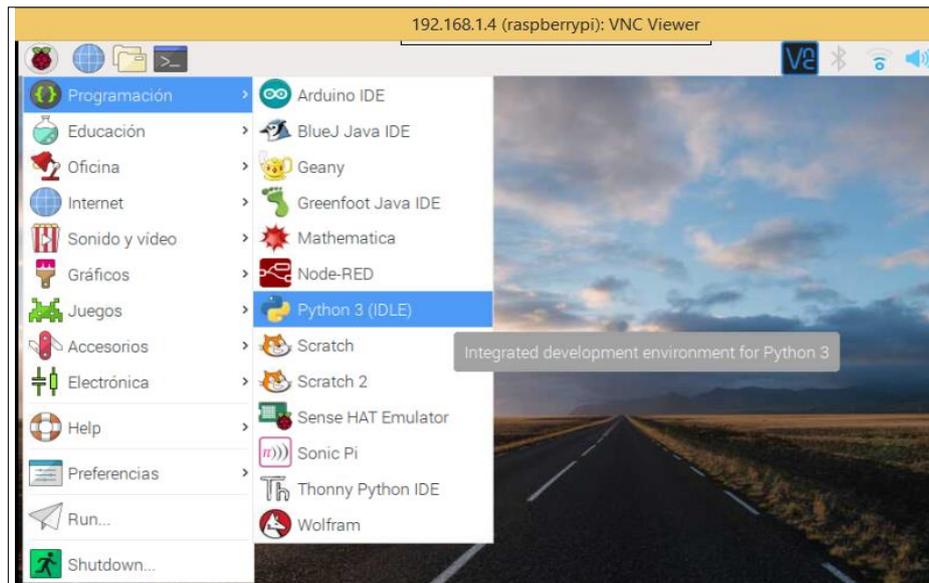
En esta parte de la investigación se presenta el software adecuado para la adquisición y procesamiento de las imágenes, así como el software para la etapa de control.

### **2.6.1. *Requerimientos del Software***

- Sistema Operativo y librerías gratuitas
- Compatibilidad de las librerías con el sistema operativo.
- Establecer comunicación serial Raspberry-Arduino.
- Obtener datos del sensor de distancia.
- Crear algoritmo para reconocimiento de objetos.
- Enviar respuestas a los actuadores para permitir o no el acceso a lugares inseguros.
- Emitir una alarma cuando el sistema de bloqueo se active.

## **2.7. Software para el Procesamiento Digital de Imágenes y la etapa de Control**

El Software usado para la programación del procesamiento digital de las imágenes es Python 3.5.0, que viene instalado por defecto en el sistema operativo de la Raspberry, mientras que el software usado para la etapa de control es Arduino cuyo lenguaje de programación se basa en C++ como se observa en la figura 21-2.



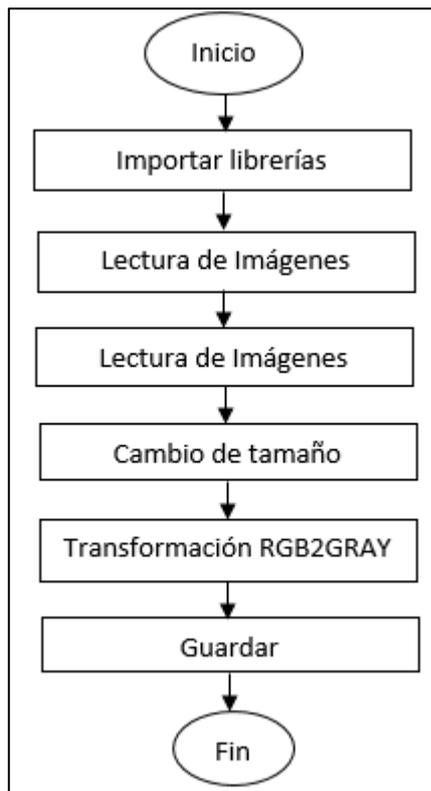
**Figura 21-2:** Python y Arduino en Raspbian  
**Realizado por:** Saquinga T, Andrade C; 2019

### 2.7.1. Programación en el IDE de Python

La programación para el procesamiento de las imágenes se hizo en 3 partes, la primera es para la lectura y creación de archivos índices de las imágenes positivas y negativas la segunda parte es para crear muestras, y finalmente la tercera parte es el entrenamiento.

En la figura 22-2 se puede observar el diagrama de flujo para la lectura de las imágenes positivas o negativas (cuyo código generado en Python se puede observar en el anexo F), su proceso se detalla a continuación:

- Inicio.
- Importar librerías para lectura.
- Lectura de imágenes.
- Cambio de tamaño.
- Transformación de las imágenes de RGB a Escala de Grises.
- Guardar imágenes transformadas en un nuevo directorio.
- Fin.



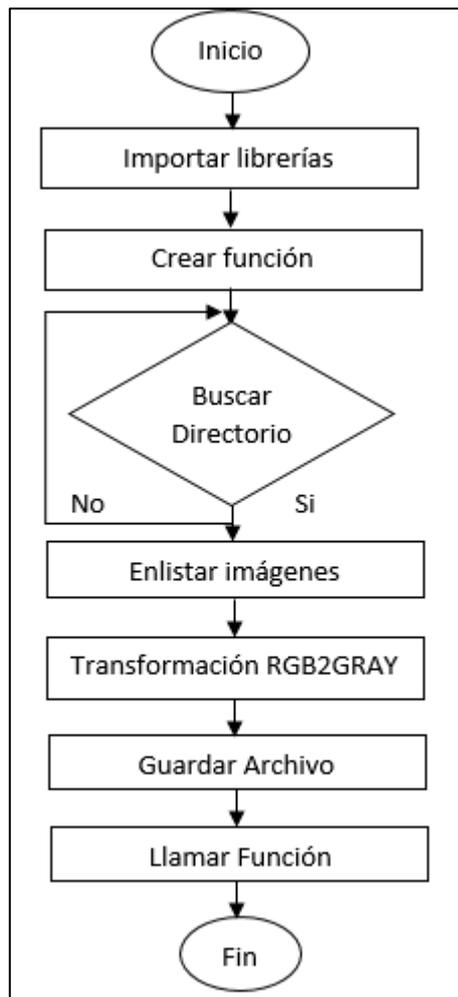
**Figura 22-2:** Diagrama de flujo para la lectura de imágenes  
**Realizado por:** Saquinga T, Andrade C; 2019

La librería y función usadas para cumplir este procedimiento son:

- **Open CV:** es una librería de visión artificial, pues posee algoritmos que hace que sea más sencillo el análisis y tratamiento de imágenes.
- **glob:** es una función que busca los archivos de cierto directorio.

En la misma etapa es necesario crear los archivos índices con las imágenes leídas (positivas y negativas), en el diagrama de flujo de la figura 23-2 se detalla el procedimiento para generar el índice negativo y cuyo código se muestra en el anexo G, los pasos de este procedimiento se mencionan a continuación:

- Inicio.
- Importar librerías.
- Crear función
- Búsqueda de imágenes en el directorio establecido.
- Enlistar las características de la imagen.
- Crear un archivo de texto que contenga el listado de imágenes con sus características.
- Guardar archivo.
- Llamar función.
- Fin.



**Figura 23-2:** Diagrama de flujo para crear el índice negativo.

Realizado por: Saquinga T, Andrade C; 2019

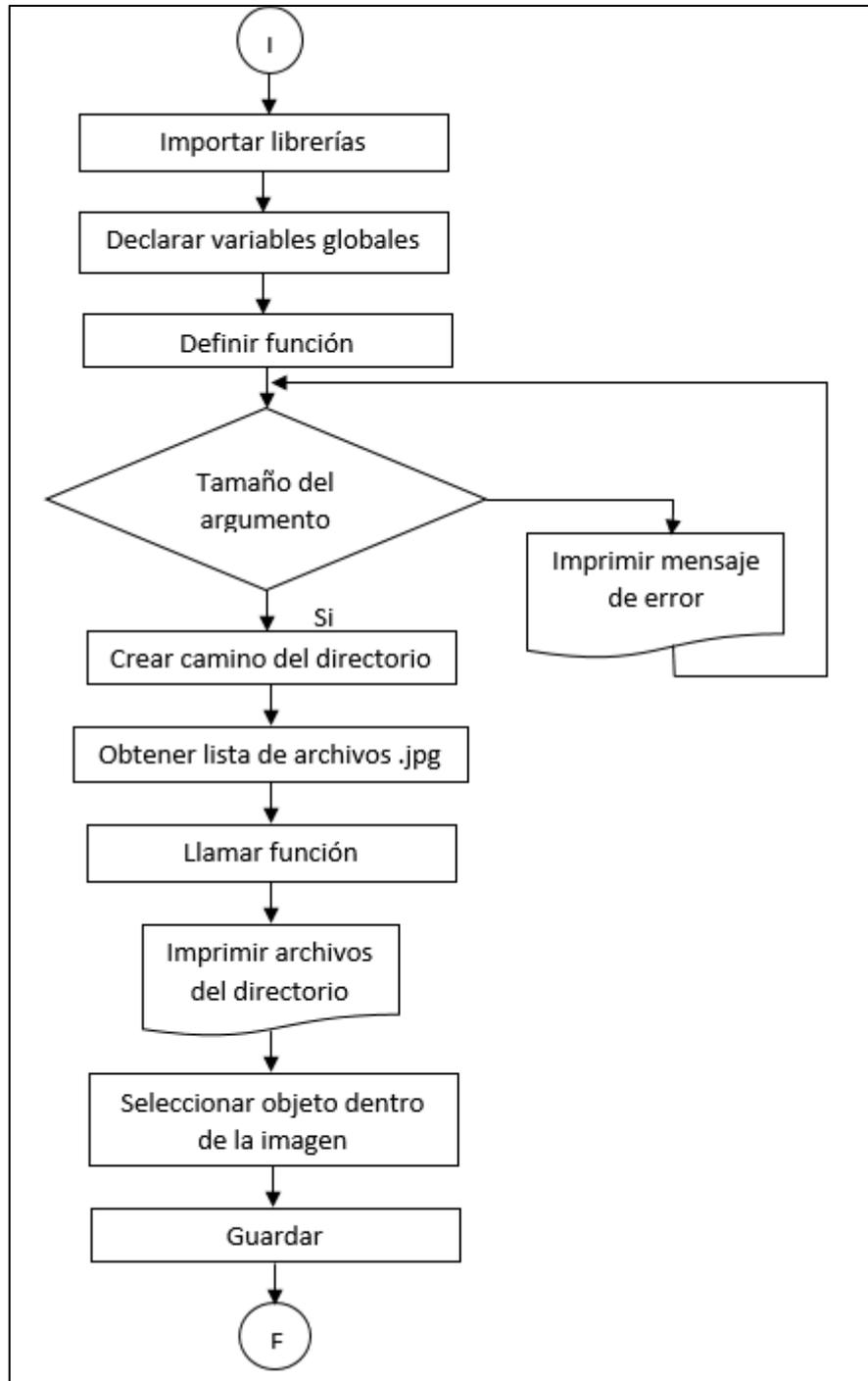
Las librerías que se usaron para cumplir con el procedimiento establecido son:

- **Os:** es un método que sirve para crear archivos dentro de una carpeta, mientras que `os.listdir` enlista el contenido de una carpeta.

Mientras que en el diagrama de flujo de la figura 24-2 se detalla el procedimiento para generar el índice positivo, en el anexo H se muestra el código para generar dicho índice y cuyos pasos se mencionan a continuación:

- Inicio.
- Importar librerías.
- Declarar variables usadas en todo el proceso.
- Crear función.
- Comparar tamaño del argumento.
- Obtener lista de archivos jpg.
- Crear ventana para visualizar las imágenes y llamar a la función.

- Imprimir los archivos encontrados en el directorio establecido.
- Seleccionar el objeto de estudio dentro de la imagen.
- Guardar.
- Fin.



**Figura 24-2:** Diagrama de flujo para crear el índice positivo.

Realizado por: Saquina T, Andrade C; 2019

Las librerías y funciones que se usaron para cumplir con el procedimiento establecido son:

- **Open CV:** librería descrita anteriormente con el diagrama de flujo de la figura 22-2.

- **sys:** es un módulo que permite acceder a las funciones y objetos que se relacionan con el intérprete de Python.
- **glob:** descrita en el diagrama de la figura 22-2.

Una vez creados los archivos en Python con los procedimientos antes mencionados es necesario ejecutar código en la consola de Raspberry como se observa en la figura 25-2.

```

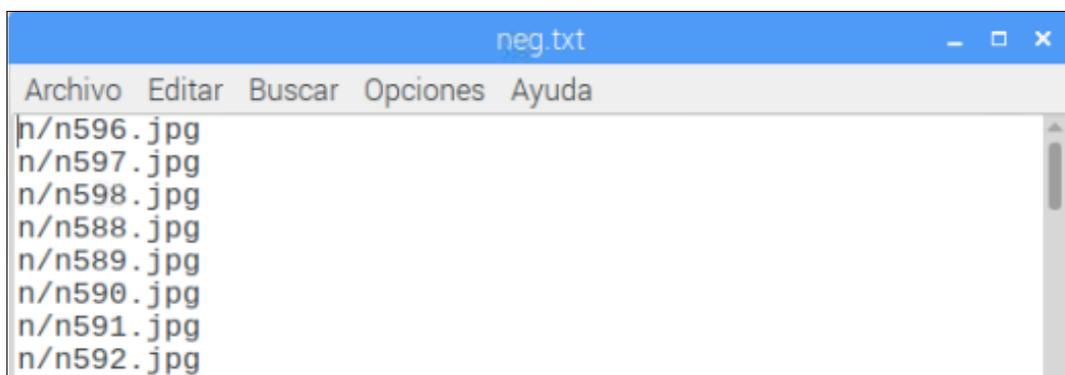
pi@raspberrypi:~ $ cd opencv-3.4.1
pi@raspberrypi:~/opencv-3.4.1 $ cd apps
pi@raspberrypi:~/opencv-3.4.1/apps $ python generacion_indices_negativos.py
pi@raspberrypi:~/opencv-3.4.1/apps $ python generacion_indice_positivo.py info pos.lst

```

**Figura 25-2:** Comandos para la creación de los archivos índice.

**Realizado por:** Saquinga T, Andrade C; 2019

Una vez que se ejecuta el código de la figura anterior (línea 3) se obtiene un archivo índice que contiene el nombre de la carpeta, nombre de la imagen y el formato (Figura 26-2).



**Figura 26-2:** Archivo índice de imágenes negativas

**Realizado por:** Saquinga T, Andrade C; 2019

Mientras que para las imágenes positivas se debe ejecutar el código de la figura 25-2 (línea 4), con el mismo se genera un archivo que lleva en su interior el nombre de la carpeta, nombre de la imagen, formato, numero de objetos positivos y la ubicación del objeto dentro de la imagen, los primeros dos valores de la ubicación son las coordenadas x e y mientras que los siguientes son largo y ancho del objeto, las últimas dos variables antes mencionadas se obtienen cuando determinamos el objeto a encontrar dentro de la imagen, es decir manualmente (Figura 27-2).



**Figura 27-2:** Archivo índice de imágenes positivas

**Realizado por:** Saquinga T, Andrade C; 2019

Ahora es importante crear las muestras, es decir, juntar los archivos creados anteriormente en un archivo vectorial, para esto se usa la herramienta `createsamples` y se ejecuta el código de la figura 28-2.

```
pi@raspberrypi:~/opencv-3.4.1/apps $ opencv_createsamples -info/pos.lst -num 600  
-w 20 -h 20 -vec pos_samples.vec
```

**Figura 28-2:** Creación del archivo vectorial de muestras.

Realizado por: Saquinga T, Andrade C; 2019

Donde:

- **info:** Es la carpeta que contiene el archivo del índice positivo.
- **pos.lst:** índice positivo.
- **num:** número de imágenes positivas
- **w, h:** ancho y alto de la muestra.
- **pos\_samples.vec:** nombre del vector.

Finalmente, para cumplir el proceso de entrenamiento se usa una herramienta llamada `traincascade`, para esto se ejecuta el código de la figura 29-2.

```
pi@raspberrypi:~/opencv-3.4.1/apps $ opencv_traincascade -data data -vec pos_samples.vec  
-neg.txt -numPos 600 -numNeg 900 -numStages 20 -w 20 -h 20
```

**Figura 29-2:** Ejecución del comando `traincascade` para el entrenamiento

Realizado por: Saquinga T, Andrade C; 2019

Donde:

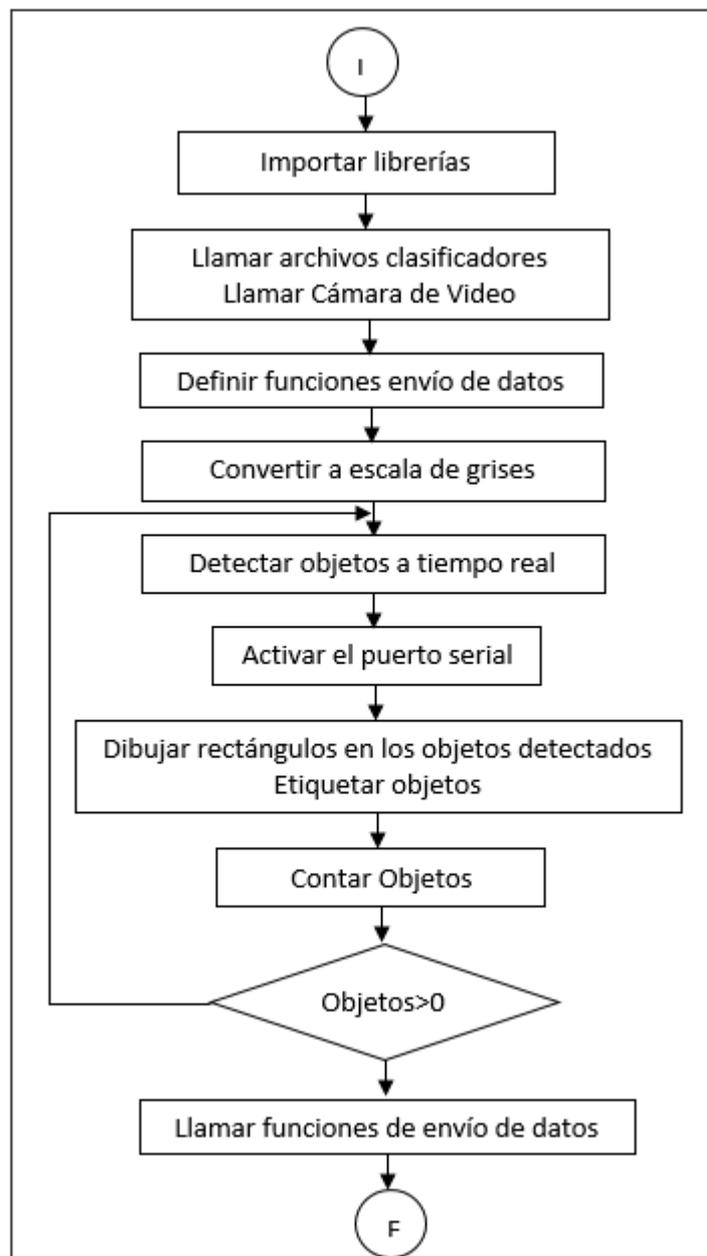
- **pos\_samples.vec:** vector de muestras.
- **neg.txt:** archivo índice negativo.
- **numPos:** número de imágenes positivas.
- **numNeg:** número de imágenes negativas.
- **numStages:** número de escenarios para el entrenamiento (mientras mayor sea este número mejor será el entrenamiento)

Cuando el proceso de entrenamiento termine se obtiene un archivo XML el cual contiene los datos del entrenamiento, este archivo se usa para la identificación de los objetos.

En el diagrama de flujo de la figura 30-2 se observa el proceso para identificar los objetos, mientras que en el anexo I se tiene el código completo en Python y cuyos pasos se detallan a continuación:

- Inicio.
- Importar Librerías.
- Llamar archivos clasificadores.
- Llamar Cámara de Video.

- Definir funciones de envío de datos hacia el Arduino.
- Convertir a escala de grises el video.
- Detectar objetos peligrosos a tiempo real.
- Activar el puerto serial para el envío de datos.
- Dibujar rectángulos en los objetos detectados.
- Etiquetar objetos.
- Definir número de objetos encontrados.
- Llamar funciones de envío de datos.
- Fin.



**Figura 30-2:** Diagrama de Flujo para detectar objetos  
 Realizado por: Saquina T, Andrade C; 2019

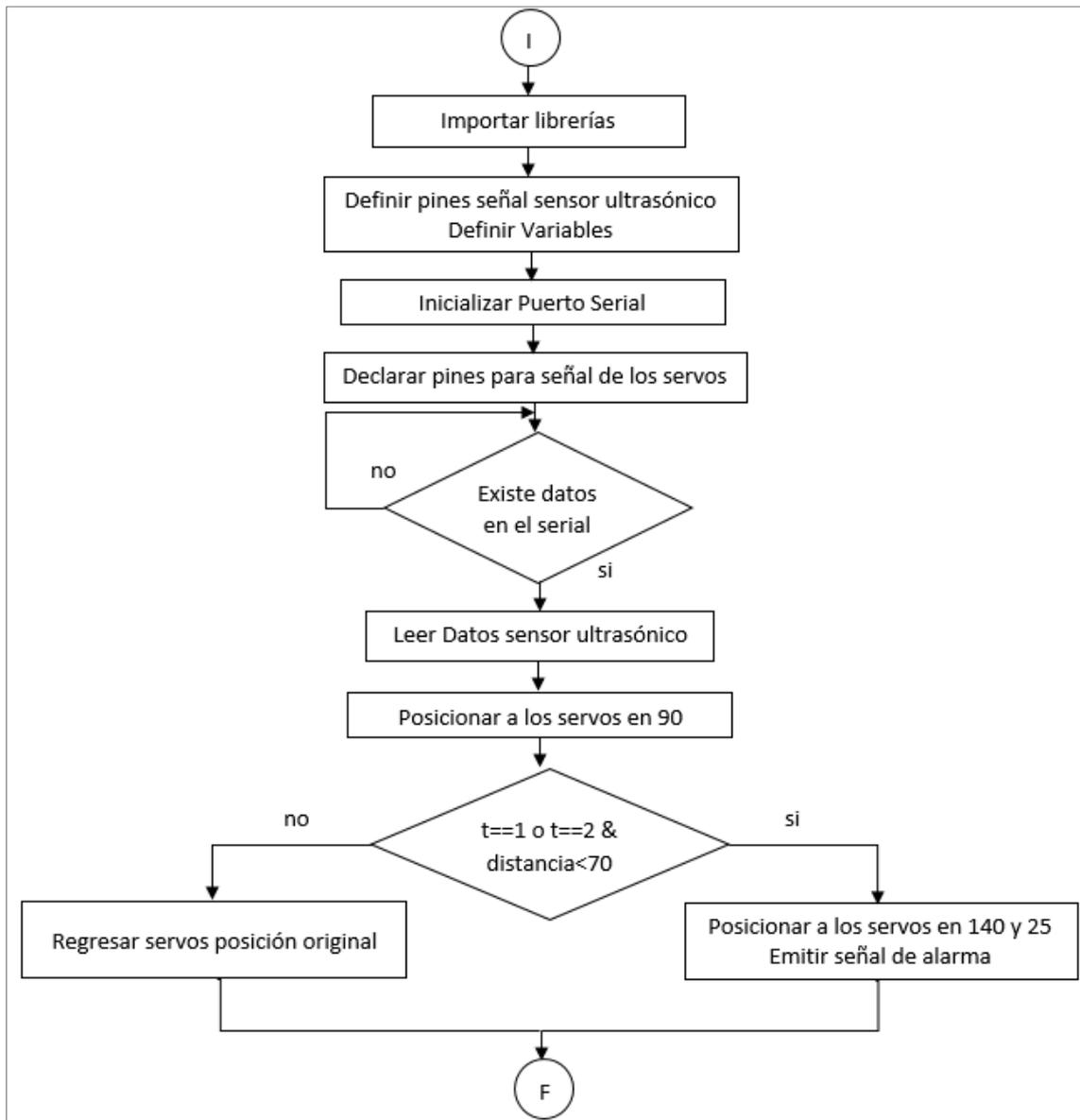
Las librerías que se usaron para el proceso de detección de objetos son:

- **Numpy:** Es uno de los paquetes fundamentales de Python capaz de dotar capacidades de cálculo matemáticas y vectoriales similares a las capacidades de Matlab, desde las funcionalidades más simples hasta las más complejas (Herrera y Sánchez, 2013).
- **Serial:** Es un módulo de comunicación entre Arduino y Python.
- **Os:** descrita en el diagrama de la figura 23-2.
- **Open CV:** descrita en el diagrama de la figura 22-2.
- **Time:** Es una librería propia de Python que sirve para contabilizar el tiempo.

### **2.7.2. Programación en el IDE de Arduino**

Finalmente, en el diagrama de flujo de la figura 31-2 se tiene el proceso para la etapa de control, es decir, el envío de señales hacia los actuadores para el proceso de frenado y para la señal de alarma, en el anexo J se tiene el código completo y cuyos pasos se detallan a continuación:

- Inicio.
- Importar Librerías.
- Definir pines para la señal del sensor ultrasónico.
- Definir variables.
- Inicializar el puerto Serial.
- Declarar pines para enviar señales a los servos.
- Buscar datos en el puerto serial.
- Leer datos del puerto serial.
- Posicionar a los servos en 90°.
- Comparar si existe un 1 en el puerto serial y la distancia que mide el sensor ultrasónico es menos a 70cm para enviar señales a los actuadores
- Posicionar a los servos en 140° y 25° y emitir una señal de alarma en el Buzzer.
- Fin.



**Figura 31-2:** Diagrama de Flujo para la etapa de control  
**Realizado por:** Saquinga T, Andrade C; 2019

Para cumplir con este proceso se usaron los siguientes módulos:

- **Ultrasonic.h:** permite la lectura del sensor ultrasónico HC-SR04.
- **Servo.h:** útil para el funcionamiento de los servos motores.

## CAPÍTULO III

### 3. ANÁLISIS Y RESULTADOS

En el presente capítulo se presentan las pruebas hechas al dispositivo, con el objetivo de validarlo a través de pruebas experimentales realizadas a cada parte constitutiva del sistema como: software (algoritmo de detección de peligros, tiempo de reconocimiento, tiempo de transmisión de datos), hardware (duración de la batería, etapa de frenado) y finalmente se analiza los costos de la implementación del prototipo.

En la figura 1-3 se observa el prototipo final conformado por la estructura del andador, sistema electrónico y sistema mecánico.



**Figura 1-3:** Prototipo final  
**Realizado por:** Saquina T, Andrade C; 2019

### **3.1. Protocolo de pruebas para evaluar el algoritmo de detección de peligros, el sistema electrónico y mecánico.**

#### **Protocolo de pruebas para evaluar el algoritmo de detección de peligros**

El protocolo que se aplicó nos permite validar el algoritmo de detección de peligros basándose en que su eficiencia debe estar en el orden del 95%, a su vez, se validó el sistema en la mañana, tarde y noche lo que nos permite introducir perturbaciones como luminosidad y distancia, para lo cual se procedió a realizar el siguiente proceso:

1. Abrir el archivo del algoritmo de reconocimiento de objetos.
2. Ejecutar el archivo del algoritmo de reconocimiento de objetos en Python.
3. Dirigir el prototipo hacia el objeto peligroso en la mañana tarde y noche para obtención de datos.
4. Capturar las imágenes reconocidas.
5. Enlistar las fallas.

#### **Protocolo de pruebas para evaluar el sistema electrónico**

Este protocolo nos permite validar los tiempos de reconocimiento de objetos y transmisión de datos basándose en que los tiempos de respuesta debe ser menor a 1 segundo, para lo cual se realizó el siguiente proceso:

1. Ejecutar el archivo del algoritmo de reconocimiento de objetos en Python.
2. Abrir el puerto serial en Arduino.
3. Dirigir el prototipo hacia el objeto peligroso para la obtención de datos.
4. Capturar el tiempo de reconocimiento de objetos.
5. Capturar el tiempo de transmisión.

## Protocolo de pruebas para evaluar el sistema mecánico

Este protocolo nos permite validar la distancia de frenado a distintas velocidades basándose en que la distancia optima entre el objeto y el prototipo debe comprender el rango de 30 a 70cm, para lo cual se procedió a realizar el siguiente proceso:

1. Ejecutar el archivo del algoritmo de reconocimiento de objetos en Python.
2. Dirigir el prototipo hacia el objeto peligroso para la obtención de datos.
3. Lanzar el prototipo de una distancia de 2,10 metros a diferentes velocidades
4. Tomar el tiempo de frenado con un cronómetro.
5. Calcular la velocidad.

### 3.2. Prueba de algoritmo de reconocimiento de lugares peligrosos

Siguiendo el protocolo de pruebas establecido se obtuvieron los datos de detección de tomacorrientes en la mañana, tarde y noche, lo cual introduce una perturbación lumínica los resultados se detallan en las tablas 1-3, 2-3 y 3-3.

**Tabla 1-3:** Datos de la detección de tomacorrientes en la mañana

<b>Detección Tomacorriente en la mañana</b>				
<b>Hora</b>	<b>Distancia</b>	<b>Intentos</b>	<b>Fallas</b>	<b>Porcentaje de error</b>
<b>09H30</b>	0-30cm	30	7	23.33%
<b>09H55</b>	31-70cm	30	0	0%
<b>10H20</b>	71-120cm	30	0	0%
<b>10H50</b>	121-200cm	30	1	3.33%
<b>11H20</b>	>200cm	30	5	16.67%

Realizado por: Saquinga T, Andrade C; 2019

**Tabla 2-3:** Datos de la detección de tomacorrientes en la tarde

<b>Detección Tomacorriente en la tarde</b>				
<b>Hora</b>	<b>Distancia</b>	<b>Intentos</b>	<b>Fallas</b>	<b>Porcentaje de error</b>
<b>15H55</b>	0-30cm	30	8	26.67%
<b>16H15</b>	31-70cm	30	0	0%
<b>16H40</b>	71-120cm	30	1	3.33%
<b>17H00</b>	121-200cm	30	2	6.67%
<b>17H30</b>	>200cm	30	5	16.67%

Realizado por: Saquinga T, Andrade C; 2019

**Tabla 3-3:** Datos de la detección de tomacorrientes en la noche

<b>Detección Tomacorriente en la noche</b>				
<b>Hora</b>	<b>Distancia</b>	<b>Intentos</b>	<b>Fallas</b>	<b>Porcentaje de error</b>
<b>19H00</b>	0-30cm	30	7	23.33%
<b>19H25</b>	31-70cm	30	1	3.33%
<b>19H50</b>	71-120cm	30	1	3.33%
<b>20H20</b>	121-200cm	30	3	10.00%
<b>20H45</b>	>200cm	30	5	16.67 %

Realizado por: Saquinga T, Andrade C; 2019

La tabla 4-3 muestra un resumen de los datos de los errores recopilados de las tablas anteriores:

**Tabla 4-3:** Datos de detección de errores de tomacorrientes

<b>Distancia</b>	<b>Luminosidad</b>		
	<b>Mañana</b>	<b>Tarde</b>	<b>Noche</b>
<b>0-30cm</b>	7	8	7
<b>31-70cm</b>	0	0	1
<b>71-120cm</b>	0	1	1

<b>121-200cm</b>	1	2	3
<b>&gt;200cm</b>	5	5	5

Realizado por: Saquinga T, Andrade C; 2019

Con los datos de la tabla 4-3 se aplicó la técnica de diseño de experimentos mixto 5x3, con el objetivo de comprobar si las variables influyen en el reconocimiento de los objetos, para llevar a cabo esta técnica se establecieron los siguientes parámetros:

**Variables independientes:** Variable A= Distancia, Variable B=Luminosidad

**Variable dependiente:** Número de fallas

**Tipo de variable:** Discreta

**Niveles:** 5 niveles para la variable A, 3 niveles para la variable B

**Número de repeticiones:** 30

**Número de corridas experimentales:** 450.

**Distribución de los datos:** Distribución Binomial (es decir existen dos posibilidades)

$$p = \text{éxito} \rightarrow \text{si reconoce el objeto}$$

$$1 - p = \text{fracaso} \rightarrow \text{no reconoce el objeto}$$

Una vez definido los parámetros que se necesitan se encontró la probabilidad de no reconocer el objeto como se observa en la tabla 5-3.

**Tabla 5-3:** Detección de tomacorrientes en probabilidades de fracaso

<b>Distancia(A)</b>	<b>Luminosidad(B)</b>		
	Mañana	Tarde	Noche
<b>0-30cm</b>	0,23	0,267	0,233
<b>31-70cm</b>	0,00	0,000	0,033
<b>71-120cm</b>	0,00	0,033	0,033
<b>121-200cm</b>	0,03	0,067	0,100
<b>&gt;200cm</b>	0,17	0,167	0,167

Realizado por: Saquinga T, Andrade C; 2019

Para saber el tipo de test estadístico que se debe aplicar, los datos tienen que cumplir el supuesto de normalidad, es decir que no deben alejarse de la recta de normalidad (gráfico 1-3), sin embargo, la tabla 5-3 muestra datos de una distribución binomial por lo que se aplicó la ecuación 1-3 para que la variable sea continua y siga una distribución normal como se observa en la tabla 6-3.

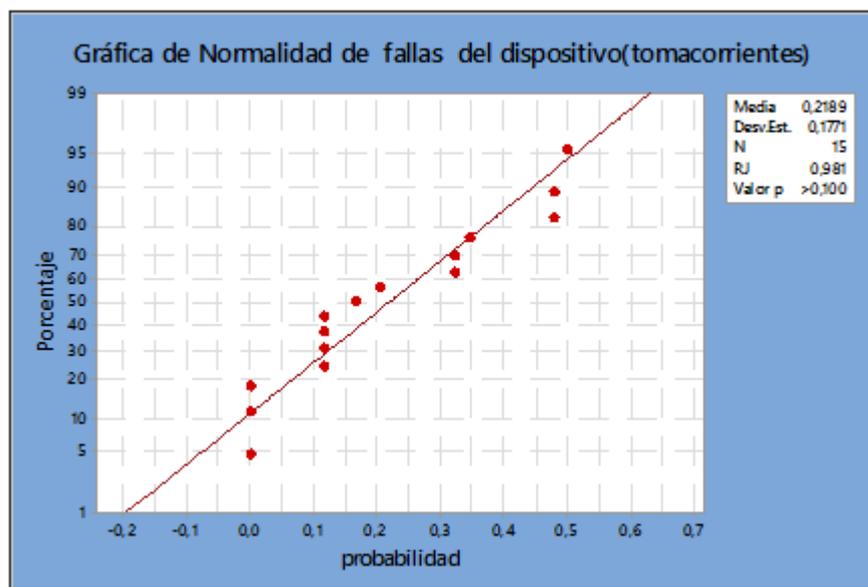
Transformación de datos para que siga distribución Normal  $\sim N(\mu, \sigma)$ , siendo la media  $\mu = 0$  y la varianza  $\sigma = 1$

$$Trans = \left(\frac{2}{\pi}\right) * \text{ArcoSen}\sqrt{(\text{probabilidad de fracaso})} \quad \text{Ecuación 1-3}$$

**Tabla 6-3:** Transformación de datos para supuesto de normalidad

Distancia(A)	Luminosidad(B)		
	mañana	tarde	Noche
0-30cm	0,32	0,35	0,32
31-70cm	0,00	0,00	0,12
71-120cm	0,00	0,12	0,12
121-200cm	0,12	0,17	0,20
>200cm	0,27	0,27	0,27

Realizado por: Saquinga T, Andrade C; 2019



**Gráfico 1-3:** Normalidad de fallas

Realizado por: Saquinga T, Andrade C; 2019

Los datos observados cumplen el supuesto de normalidad, con un valor de p de 0,10 a un nivel de confianza del 90%, por lo que se realizó el Anova que es un estudio en el cual se evalúa si las variables (distancia, luminosidad) influyen en el reconocimiento del objeto.

### Hipótesis Estadísticas

$$H_0: A = 0 \quad H_1: A \neq 0$$

$$H_0: B = 0 \quad H_1: B \neq 0$$

$$H_0: AB=0 \quad H_1: AB \neq 0$$

### Hipótesis Textuales

$H_0$ : La distancia a la que se encuentre el prototipo, respecto al tomacorriente no influye en el reconocimiento.

$H_1$ : La distancia a la que se encuentre el prototipo, respecto al tomacorriente influye en el reconocimiento.

$H_0$ : La luminosidad no influye en el reconocimiento

$H_1$ : La luminosidad influye en el reconocimiento

$H_0$ : La interacción de los dos factores: distancia y luminosidad no influyen en el reconocimiento

$H_1$ : La interacción de los dos factores: distancia y luminosidad influyen en el reconocimiento

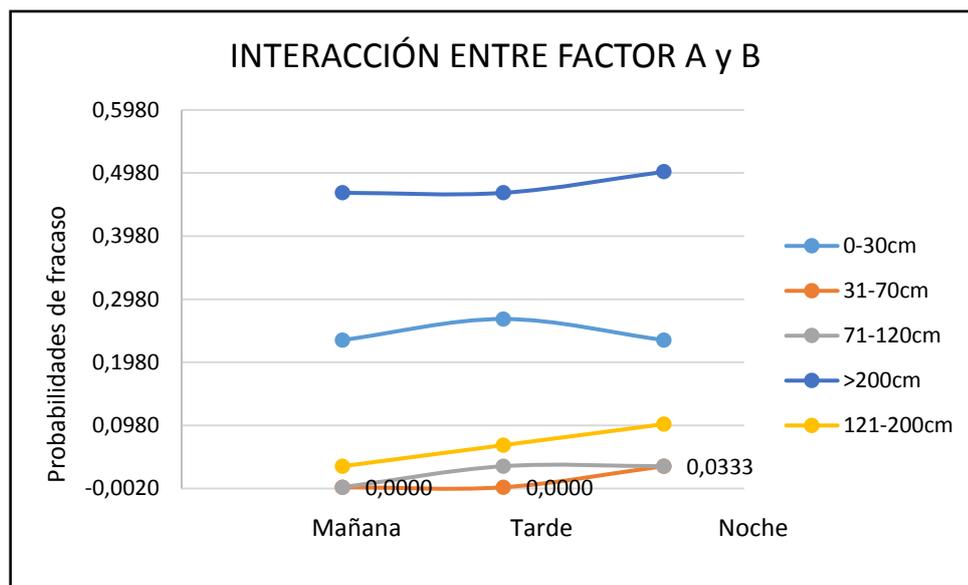
Los factores analizados fueron las distancias a la que se encuentra el prototipo, respecto al tomacorriente y la luminosidad del día, para determinar el reconocimiento, cuya variable respuesta fue el número de fallas del mismo.

**Tabla 7-3:** Estadístico de prueba-Anova

FV	SC	GL	CM	F	Valor-p
<b>EFEECTO A</b>	0,006	1	0,00604293	4,091582483	0,044
<b>EFEECTO B</b>	0,000	2	0,00017449	0,118147019	0,889
<b>EFEECTO AB</b>	0,000	11	3,6467E-05	0,024691576	1,000
<b>ERROR</b>	0,642	435	0,00147692		
<b>TOTAL</b>	0,64925	449			

Realizado por: Saquinga T, Andrade C; 2019

Con un valor de  $p$  igual a 0,04, se rechazó la hipótesis nula, concluyendo que el factor A (distancias de pruebas aplicadas) afecta de manera significativa en el reconocimiento mientras que el factor B (luminosidad) con un valor de  $p$  de 0,88, no influye en el reconocimiento, es decir que el prototipo reconocerá el tomacorriente sin ningún problema, sin importar la luminosidad del día, finalmente, la interacción de los factores juntos con un valor de  $p$  de 1 no afecta en la variable respuesta (número de fallas). Los resultados fueron medidos a un nivel de significancia de 0,05 y un nivel de confianza del 95%.



**Gráfico 2-3:** Interacción entre distancia respecto al tomacorriente y Luminosidad  
**Realizado por:** Saquinga T, Andrade C; 2019

Se determinó que la distancia óptima para el reconocimiento es de 31 a 120 centímetros a diferente luminosidad del día (mañana, tarde, noche) ya que el factor luminosidad no afecta en el reconocimiento, con un porcentaje de falla de 1,67, es decir que el algoritmo de reconocimiento es 98,3% efectivo.

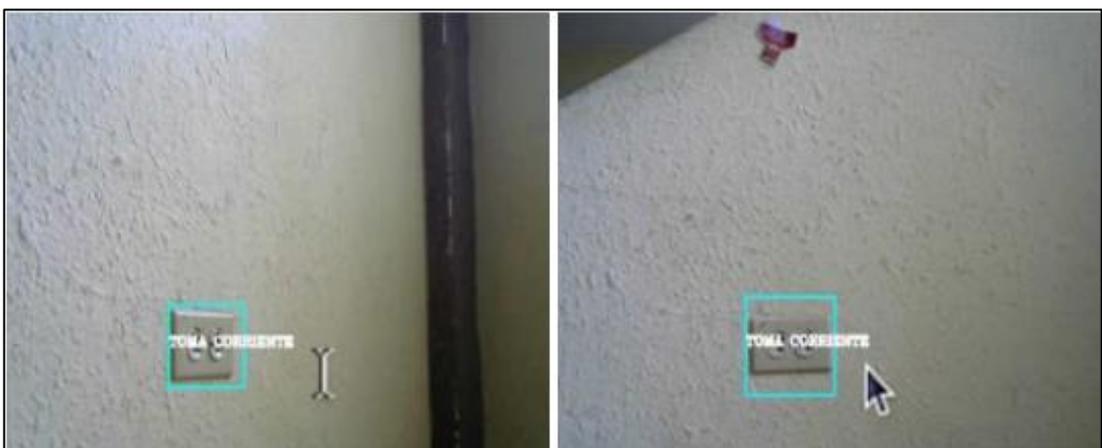
A continuación, se presentan imágenes del reconocimiento del tomacorriente a diferentes distancias.



**Figura 2-3:** Reconocimiento del tomacorriente rango 0-30cm  
Realizado por: Saquinga T, Andrade C; 2019



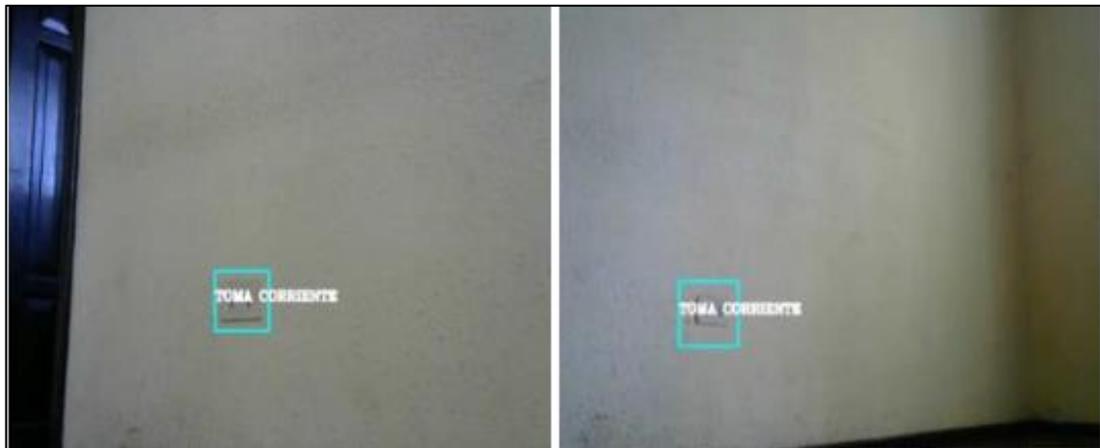
**Figura 3-3:** Reconocimiento del tomacorriente rango 31-70cm  
Realizado por: Saquinga T, Andrade C; 2019



**Figura 4-3:** Reconocimiento del tomacorriente rango 71-120cm  
Realizado por: Saquinga T, Andrade C; 2019



**Figura 5-3:** Reconocimiento del tomacorriente rango 121-200cm  
**Realizado por:** Saquinga T, Andrade C; 2019



**Figura 6-3:** Reconocimiento del tomacorriente rango >200cm  
**Realizado por:** Saquinga T, Andrade C; 2019

Continuando el protocolo de pruebas se obtuvieron los datos de detección del identificador del tomacorriente en la mañana, tarde y noche detallados en las tablas 8-3, 9-3 y 10-3.

**Tabla 8-3:** Datos de la detección de gradas en la mañana

<b>Detección gradas en la mañana</b>				
<b>Hora</b>	<b>Distancia</b>	<b>Intentos</b>	<b>Fallas</b>	<b>Porcentaje de error</b>
<b>08H00</b>	0-30cm	30	10	33.33%
<b>08H30</b>	31-70cm	30	0	0.00%
<b>08H55</b>	71-120cm	30	1	3.33%
<b>09H25</b>	121-200cm	30	2	6.67%

**Realizado por:** Saquinga T, Andrade C; 2019

**Tabla 9-3:** Datos de la detección de gradas en la tarde

<b>Detección gradas en la tarde</b>				
<b>Hora</b>	<b>Distancia</b>	<b>Intentos</b>	<b>Fallas</b>	<b>Porcentaje de error</b>
<b>14H05</b>	0-30cm	30	11	36.67%
<b>14H30</b>	31-70cm	30	1	3.33%
<b>14H50</b>	71-120cm	30	1	3.33%
<b>15H20</b>	121-200cm	30	4	13.33%

Realizado por: Saquina T, Andrade C; 2019

**Tabla 10-3:** Datos de la detección de gradas en la noche

<b>Detección gradas en la noche</b>				
<b>Hora</b>	<b>Distancia</b>	<b>Intentos</b>	<b>Fallas</b>	<b>Porcentaje de error</b>
<b>19H30</b>	0-30cm	30	10	33.33%
<b>19H55</b>	31-70cm	30	1	3.33%
<b>20H25</b>	71-120cm	30	2	6.67%
<b>20H50</b>	121-200cm	30	3	10.00%

Realizado por: Saquina T, Andrade C; 2019

**Tabla 11-3:** Datos de detección de gradas

<b>Distancia</b>	<b>Luminosidad</b>		
	<b>mañana</b>	<b>tarde</b>	<b>noche</b>
<b>0-30cm</b>	10	11	10
<b>31-70cm</b>	0	1	1
<b>71-120cm</b>	1	1	2
<b>121-200cm</b>	2	4	3

Realizado por: Saquina T, Andrade C; 2019

**Tabla 12-3:** Detección de gradas en probabilidades de fracaso

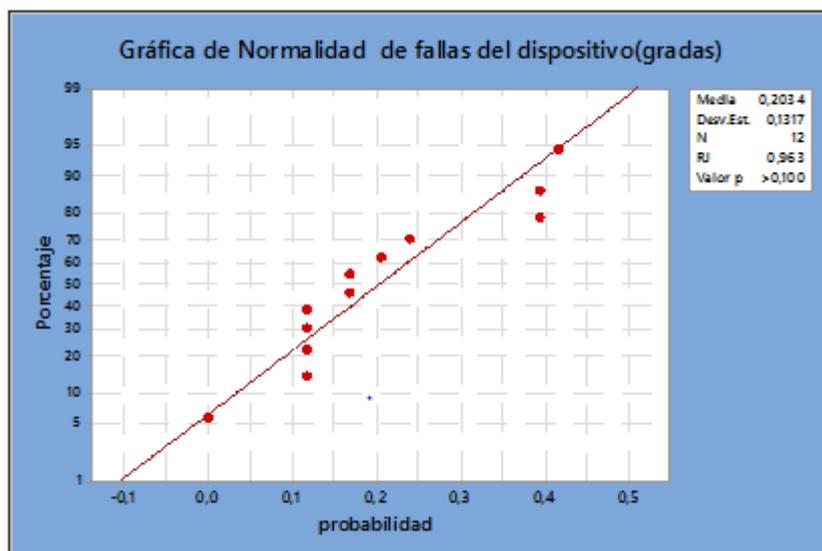
Distancia(A)	Luminosidad(B)		
	mañana	tarde	noche
<b>0-30cm</b>	0,33	0,367	0,333
<b>31-70cm</b>	0,00	0,033	0,033
<b>71-120cm</b>	0,03	0,033	0,067
<b>121-200cm</b>	0,07	0,133	0,100

Realizado por: Saquinga T, Andrade C; 2019

**Tabla 13-3:** Transformación de datos para supuesto de normalidad

Distancia(A)	Luminosidad(B)		
	mañana	tarde	noche
<b>0-30cm</b>	0,39	0,41	0,39
<b>31-70cm</b>	0,00	0,12	0,12
<b>71-120cm</b>	0,12	0,12	0,17
<b>121-200cm</b>	0,17	0,24	0,20

Realizado por: Saquinga T, Andrade C; 2019



**Gráfico 3-3:** Normalidad de fallas

Realizado por: Saquinga T, Andrade C; 2019

Los datos observados cumplen el supuesto de normalidad, con un valor de p de 0,10 a un nivel de confianza del 90%, por lo cual se procede a realizar el Anova.

### Hipótesis Estadísticas

$$H_0: A = 0 \quad H_1: A \neq 0$$

$$H_0: B = 0 \quad H_1: B \neq 0$$

$$H_0: AB=0 \quad H_1: AB \neq 0$$

### Hipótesis Textuales

$H_0$ : La distancia a la que se encuentre el prototipo, respecto a las gradas no influye en el reconocimiento

$H_1$ : La distancia a la que se encuentre el prototipo, respecto a las gradas influye en el reconocimiento

$H_0$ : La luminosidad no influye en el reconocimiento

$H_1$ : La luminosidad influye en el reconocimiento

$H_0$ : La interacción de los dos factores: distancia y luminosidad no influyen en el reconocimiento

$H_1$ : La interacción de los dos factores: distancia y luminosidad influyen en el reconocimiento

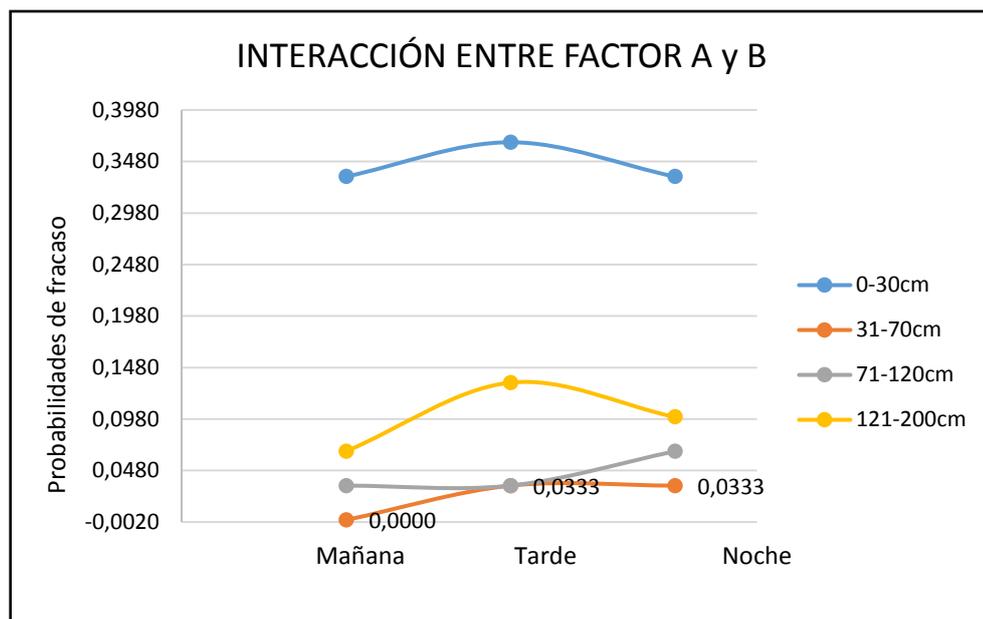
Los factores analizados fueron las distancias a la que se encuentra el prototipo, respecto a las gradas y la luminosidad del día, para determinar el reconocimiento de objetos, cuya variable respuesta fue el número de fallas del mismo.

**Tabla 14-3:** Estadístico de prueba-Anova

<b>FV</b>	<b>SC</b>	<b>GL</b>	<b>CM</b>	<b>F</b>	<b>Valor-p</b>
<b>EFEECTO A</b>	0,009	1	0,00920985	6,032370182	0,014
<b>EFEECTO B</b>	0,000	2	9,6052E-05	0,062913491	0,939
<b>EFEECTO AB</b>	0,000	11	2,3867E-05	0,015632436	1,000
<b>ERROR</b>	0,664	435	0,00152674		
<b>TOTAL</b>	0,6738	449			

Realizado por: Saquinga T, Andrade C; 2019

Con un valor de  $p$  igual a 0,01, se rechazó la hipótesis nula, concluyendo que el factor A (distancias de pruebas respecto al identificador gradas) afecta de manera significativa en el reconocimiento, mientras que el factor B (luminosidad) con un valor de  $p$  de 0,93, no influye en el reconocimiento, es decir que el prototipo reconocerá el identificador gradas sin ningún problema, sin importar la luminosidad del día, finalmente, la interacción de los factores juntos con un valor de  $p$  de 1 no afecta en la variable respuesta. Los resultados fueron medidos a un nivel de significancia de 0,05 y un nivel de confianza del 95%



**Gráfico 4-3:** Interacción entre Distancia respecto al tomacorriente y Luminosidad  
**Realizado por:** Saquinga T, Andrade C; 2019

Se determinó que la distancia óptima para el reconocimiento es de 31 a 120 centímetros a diferente luminosidad del día (mañana, tarde, noche) ya que el factor luminosidad no afecta en el reconocimiento, con una probabilidad de que el algoritmo no reconozca de 0,033, lo que equivale al 3,33% de error, es decir el algoritmo es 96,67% efectivo.

A continuación, se presentan imágenes del reconocimiento del identificador de las gradas a diferentes distancias



**Figura 7-3:** Reconocimiento del identificador de la grada rango 0-30cm  
Realizado por: Saquina T, Andrade C; 2019



**Figura 8-3:** Reconocimiento del identificador de la grada rango 31-70cm  
Realizado por: Saquina T, Andrade C; 2019



**Figura 9-3:** Reconocimiento del identificador de la grada rango 71-120cm  
Realizado por: Saquina T, Andrade C; 2019



**Figura 10-3:** Reconocimiento del identificador de la grada rango >200cm  
 Realizado por: Saquina T, Andrade C; 2019

### 3.3. Prueba de tiempo de reconocimiento

Para recolectar datos del tiempo de reconocimiento se utilizó la librería time, esta información se muestra en la consola de Python como se observa en la figura 11-3 y se aplicó el protocolo de pruebas.

```

etiqueta3.py x
47
48     for (x, y, w, h) in gradas:
49         cv2.rectangle(img, (x, y), (x+w, y+h)
Shell
tiempo de reconocimiento 0.006565
tiempo de reconocimiento 0.003202
tiempo de reconocimiento 0.003233
tiempo de reconocimiento 0.006969
tiempo de reconocimiento 0.003128
tiempo de reconocimiento 0.003097
tiempo de reconocimiento 0.003225
tiempo de reconocimiento 0.003464
tiempo de reconocimiento 0.003496
tiempo de reconocimiento 0.007555
tiempo de reconocimiento 0.003249
tiempo de reconocimiento 0.003508
tiempo de reconocimiento 0.007649
tiempo de reconocimiento 0.003216
tiempo de reconocimiento 0.003379
tiempo de reconocimiento 0.003300
tiempo de reconocimiento 0.003015
tiempo de reconocimiento 0.006625
tiempo de reconocimiento 0.003021
tiempo de reconocimiento 0.003007
tiempo de reconocimiento 0.003202
tiempo de reconocimiento 0.003017
tiempo de reconocimiento 0.003004
tiempo de reconocimiento 0.003086
tiempo de reconocimiento 0.003027
tiempo de reconocimiento 0.003039
tiempo de reconocimiento 0.010679
tiempo de reconocimiento 0.003104
tiempo de reconocimiento 0.003558
  
```

**Figura 11-3:** Datos mostrados en la consola de Python  
 Realizado por: Saquina T, Andrade C; 2019

En la tabla 15-3 se muestran los datos recolectados del tiempo de reconocimiento de los objetos en estudio.

**Tabla 15-3:** Tiempo de reconocimiento de objetos

Prueba	Tiempo (segundos)		
	1 objeto	2 objetos	3 objetos
1	0.003102	0.006565	0.012516
2	0.003202	0.006432	0.015462
3	0.003233	0.006938	0.023145
4	0.003205	0.006969	0.017425
5	0.003128	0.007102	0.013521
6	0.003097	0.006598	0.017423
7	0.003225	0.006842	0.018974
8	0.003464	0.006245	0.012357
9	0.003496	0.007423	0.011451
10	0.003354	0.007555	0.013452
11	0.003249	0.007285	0.013692
12	0.003508	0.007159	0.021421
13	0.003562	0.007649	0.012542
14	0.003216	0.008952	0.019142
15	0.003379	0.007412	0.018110
16	0.003300	0.006521	0.016237
17	0.003015	0.007326	0.013121
18	0.003751	0.006625	0.014562
19	0.003021	0.007482	0.018945
20	0.003007	0.006654	0.012783
21	0.003202	0.006211	0.014231
22	0.003017	0.006612	0.011859

<b>23</b>	0.003004	0.006514	0.011412
<b>24</b>	0.003086	0.006605	0.015268
<b>25</b>	0.003027	0.006745	0.017031
<b>26</b>	0.003039	0.006323	0.014161
<b>27</b>	0.003301	0.008005	0.010379
<b>28</b>	0.003104	0.006523	0.012563
<b>29</b>	0.003558	0.007231	0.021031
<b>30</b>	0.003106	0.007536	0.014152
<b>Promedio</b>	0.003379	0.007001	0.015279

Realizado por: Saquinga T, Andrade C; 2019

Según los datos presentados en la tabla 15-3 se determinó que el tiempo promedio de reconocimiento de un objeto es de 3.38ms y este aumenta a medida que detecta más objetos (gráfico 1-3), esto se debe a que el software tarda en procesar las imágenes que aparecen en la escena.

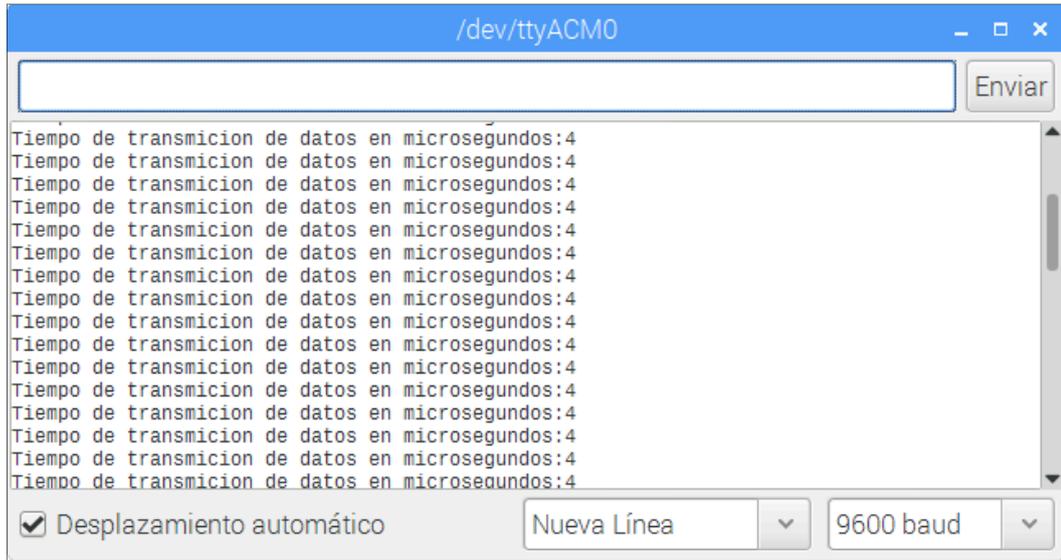


**Gráfico 5-3:** Tiempo Promedio de reconocimiento de Objetos

Realizado por: Saquinga T, Andrade C; 2019

### 3.4. Prueba de tiempo de transmisión de datos

Para calcular el tiempo de transmisión de datos desde la Raspberry hacia el Arduino se usa la función micros y para visualizarlo es necesario seguir el protocolo antes mencionado.



**Figura 12-3:** Datos mostrados en la consola de Python

Realizado por: Saquinga T, Andrade C; 2019

Con los datos que se observan en la figura 12-3, se determinó que el tiempo de transmisión de datos es 4 microsegundos, lo que significa que es un tiempo óptimo para que los actuadores cumplan su función.

### 3.5. Prueba de frenado

La tabla 16-3 muestra las distancias de frenado recogidas mediante pruebas que se realizaron aplicando el protocolo antes mencionado.

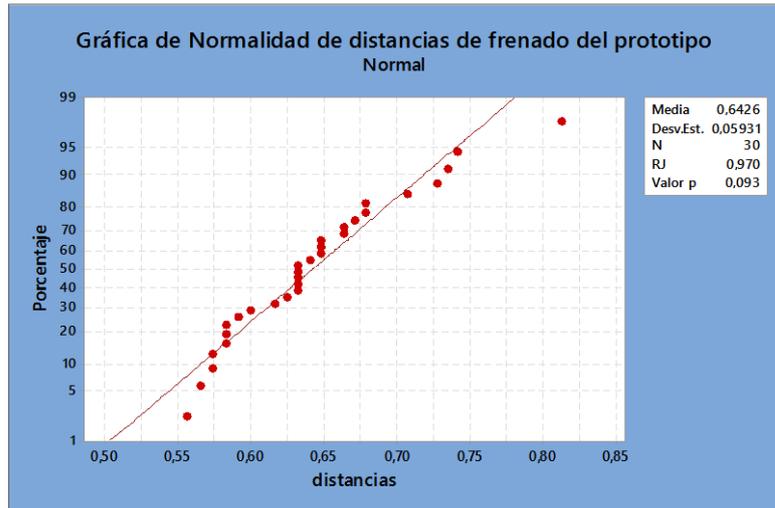
**Tabla 16-3:** Distancias de frenado

N	Distancia de lanzamiento	Distancia de frenado	Diferencia	Tiempo	Velocidad
1	2,1	1,77	0,33	1,80	0,98
2	2,1	1,70	0,40	2,32	0,73
3	2,1	1,72	0,38	1,37	1,26
4	2,1	1,69	0,41	2,46	0,69
5	2,1	1,76	0,34	2,12	0,83
6	2,1	1,70	0,40	1,19	1,43
7	2,1	1,75	0,35	1,61	1,09

8	2,1	1,70	0,40	1,29	1,32
9	2,1	1,68	0,42	2,67	0,63
10	2,1	1,65	0,45	4,51	0,37
11	2,1	1,79	0,31	1,40	1,28
12	2,1	1,77	0,33	2,06	0,86
13	2,1	1,76	0,34	1,31	1,34
14	2,1	1,71	0,39	2,43	0,70
15	2,1	1,78	0,32	2,10	0,85
16	2,1	1,76	0,34	2,45	0,72
17	2,1	1,66	0,44	2,93	0,57
18	2,1	1,64	0,46	3,40	0,48
19	2,1	1,55	0,55	4,31	0,36
20	2,1	1,68	0,42	2,97	0,57
21	2,1	1,74	0,36	3,20	0,54
22	2,1	1,70	0,40	3,93	0,43
23	2,1	1,44	0,66	4,24	0,34
24	2,1	1,60	0,50	3,95	0,41
25	2,1	1,56	0,54	3,79	0,41
6	2,1	1,68	0,42	3,71	0,45
27	2,1	1,70	0,40	2,64	0,64
28	2,1	1,66	0,44	3,99	0,42
29	2,1	1,64	0,46	4,06	0,40
30	2,1	1,57	0,53	4,44	0,35

Realizado por: Saquinga T, Andrade C; 2019

Con los datos de la tabla anterior se pudo evidenciar que el prototipo se demora en frenar a mayor velocidad, siendo la velocidad mínima de 0,34 *m/s* y máxima de 1,43 *m/s*, con una distancia de frenado de 1,44 y 1,70 respectivamente. La diferencia entre la distancia total de lanzamiento del prototipo y distancia de frenado, da como resultado la distancia entre el prototipo y tomacorriente. El valor promedio de las diferencias fue de 0,42 centímetros.



**Gráfico 6-3:** Normalidad de distancias de frenado  
**Realizado por:** Saquinga T, Andrade C; 2019

Los datos observados cumplen el supuesto de normalidad, con un valor de p de 0,09 a un nivel de confianza del 90%.

### Hipótesis estadísticas

$$H_0: \text{distancias}=0 \quad H_1: \text{distancias} \neq 0$$

### Hipótesis textuales

$H_0$ : Sistema de frenado deficiente (distancia de frenado cercana al tomacorriente)

$H_1$ : Sistema de frenado eficiente (distancia de frenado alejada del tomacorriente)

### Estadístico de prueba

$$t_{n-1} = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$

$$t_{n-1} = \frac{0,6426 - 0}{0,059/\sqrt{30}}$$

$$t_{n-1} = 59,65$$

$$t_{1-\frac{\alpha}{2}} = 2$$

$$tc > t_{1-\frac{\alpha}{2}} \text{ Ho se rechaza}$$

$$28,7 > 2 \text{ Ho se rechaza}$$

Con un valor del estadístico de prueba  $t$  igual a 28,7 y un valor crítico de 2 tomado del anexo K se rechaza  $H_0$ , es decir que el sistema de frenado es eficiente, con una adecuada distancia entre el dispositivo y el objeto de estudio, identificado como zona de peligro, con un valor  $p$  de 0,001, un nivel de significancia de 0,05 y de confianza del 95%

### 3.6. Prueba de tiempo de duración de las baterías

Para recolectar datos de tiempo de duración de las baterías que alimentan al sistema electrónico se hicieron 30 pruebas, la información se presenta en las siguientes tablas:

**Tabla 17-3:** Tiempo de duración de la batería RPI

<b>Duración de batería RPI Powerpack V2.0</b>			
<b>Prueba</b>	<b>Inicio</b>	<b>Fin</b>	<b>Tiempo de duración</b>
1	17H15	19H30	2H15
2	10H45	13H03	2H18
3	20H36	22H56	2H20
4	08H25	10H30	2H05
5	13H28	15H38	2H10
6	19H30	21H40	2H10
7	07H26	09H38	2H12
8	14H02	15H58	2H00
9	20H25	22H47	2H22
10	09H34	11H28	1H54
11	16H28	18H32	2H04
12	08H20	10H42	2H22
13	15H36	17H27	1H51
14	19H23	21H41	2H18
15	09H11	11H31	2H20

16	14H20	16H46	2H26
17	20H16	22H27	2H11
18	10H12	12H15	2H03
19	16H22	18H42	2H20
20	08H06	10H16	2H10
21	13H22	15H22	2H00
22	18H30	20H28	1H58
23	09H37	11H42	2H05
24	14H42	16H59	2H17
25	20H21	22H27	2H06
26	08H28	10H41	2H13
27	14H15	16H33	2H18
28	19H52	21H54	2H02
29	10H45	12H59	2H14
30	16H35	18H43	2H08
<b>Promedio</b>			2H06

Realizado por: Saquina T, Andrade C; 2019

**Tabla 18-3:** Tiempo de duración de la batería Lipo

<b>Duración de batería Lipo 2S</b>			
<b>Prueba</b>	<b>Inicio</b>	<b>Fin</b>	<b>Tiempo de duración</b>
1	17H15	19H40	2H25
2	10H45	13H05	2H20
3	20H36	22H59	2H23
4	08H25	10H42	2H17
5	13H28	15H43	2H15

6	19H30	21H48	2H18
7	07H26	09H52	2H26
8	14H02	16H22	2H24
9	20H25	22H51	2H26
10	09H34	11H52	2H18
11	16H28	18H38	2H10
12	08H20	10H45	2H25
13	15H36	17H46	2H10
14	19H23	21H44	2H21
15	09H11	11H34	2H23
16	14H20	16H46	2H26
17	20H16	22H35	2H19
18	10H12	12H36	2H24
19	16H22	18H44	2H22
20	08H06	10H21	2H15
21	13H22	15H38	2H16
22	18H30	20H53	2H23
23	09H37	12H02	2H25
24	14H42	17H01	2H19
25	20H21	22H32	2H11
26	08H28	10H50	2H22
27	14H15	16H38	2H23
28	19H52	22H06	2H14
29	10H45	13H02	2H17
30	16H35	19H00	2H25
<b>Promedio</b>			2H20

Realizado por: Saquinga T, Andrade C; 2019

En las tablas 17-3 y 18-3 se puede apreciar los valores de la duración de las baterías en horas, por lo que se determinó que el tiempo promedio de autonomía que posee el sistema es el tiempo promedio de duración de la batería RPI Powerpack igual a 2 horas con 6 minutos, siendo esta una duración mayor a la establecida en los requerimientos mínimos del prototipo, la razón por la cual se eligió este valor es porque si se agota la energía de la batería antes mencionada el controlador y microcontrolador se apagan dejando sin funcionamiento a los motores.

### 3.7. Análisis de costos

En la tabla 19-3 se detalla el costo de cada uno de los componentes utilizados en la implementación del prototipo.

**Tabla 19-3:** Costos de implementación

Componentes	Cantidad	Costo (\$)
Andador clásico	1	30.00
Raspberry Pi3 B	1	75.00
Arduino Mega 2560	1	15.00
Cámara Logitech C270	1	35.00
Servo motor MG996R	2	24.00
Batería RPI Powerpack V2.0	1	25.00
Batería Lipo 2S 1 <sup>a</sup>	1	20.00
Sensor HC-SR04	1	3.00
Buzzer	1	0.70
Estructura mecánica	1	30.00
<b>Total</b>		<b>257.7</b>

Realizado por: Saquinga T, Andrade C; 2019

Como se observa en la tabla 19-3 el costo total del prototipo es \$257.70 siendo un costo aceptable ya que brinda mayor seguridad que los andadores convencionales y también porque puede ser reusable hasta por tres generaciones.

## CONCLUSIONES

- Se implementó un prototipo de sistema electrónico en un andador para niños que controla el acceso a los lugares peligrosos más comunes detallados en el ítem 1.1.1 del capítulo I que enfrenta el menor al desplazarse en el andador, usando visión artificial.
- Los elementos que conforman el hardware fueron seleccionados después de analizar las funciones que debe realizar cada uno de ellos descritos en las tablas 1-2 hasta la 9-2, con el objetivo de cumplir los requerimientos del prototipo.
- Para construir la parte mecánica del andador se diseñó cada una de las partes que conforman el sistema de frenado en el software SolidWorks, para esto se tomó como referencia el mecanismo de frenado de las bicicletas y se acopló a un andador comercial.
- Se determinó que el tiempo promedio de reconocimiento de un objeto es de 3.38ms y este aumenta a medida que el algoritmo detecta más objetos, además el tiempo de transmisión de datos desde la Raspberry hacia el Arduino es de 4 microsegundos, lo que significa que el tiempo de respuesta de los actuadores es inmediato.
- Se determinó que la distancia ideal de reconocimiento está entre 31 - 120 cm, porque es la distancia donde el campo visual de la cámara tiene mayor cobertura del entorno, permitiendo al algoritmo obtener mejores detalles del objeto para su procesamiento.
- La distancia óptima para el frenado está en el rango de 30-70cm, porque al seleccionar un rango menor el campo visual de la cámara no cubre el objeto y el infante entraría a la zona de peligro.
- Se concluyó que el prototipo posee autonomía de hasta 2 horas con 6 minutos cumpliendo con el tiempo establecido en los requerimientos.

- El costo final de la implementación del prototipo es de \$257.70 siendo un precio aceptable ya que el dispositivo se puede usar hasta por tres generaciones.

## RECOMENDACIONES

- Integrar más cámaras en el prototipo para tener mayor cobertura de las direcciones a las que puede acceder el niño cuando usa el andador.
- Reducir el tamaño de las imágenes positivas y negativas con el objetivo de disminuir el costo computacional de la Raspberry al momento de entrenar.
- Para obtener mejores resultados del entrenamiento se debe usar mayor número de imágenes negativas que positivas y estas deben ser capturadas en distintos ángulos y diferente tipo de iluminación.
- Usar el software VNC Viewer para acceder de forma remota a la Raspberry a través del internet ya que permite visualizar y controlar su interfaz gráfica, a diferencia del software Putty que trabaja por medio de la terminal del procesador para el ingreso de comandos.
- Usar el identificador de las gradas en otros lugares peligrosos ya que el algoritmo está entrenado para detectar la señal de stop.
- El dispositivo está diseñado para que funcione en ambientes controlados por lo que se recomienda evitar usarlo en lugares abiertos, pues la lluvia u otros factores, podrían afectar su funcionamiento.
- Se recomienda desconectar los elementos electrónicos de las baterías cuando el dispositivo no esté en funcionamiento para mantener el tiempo de autonomía del prototipo.

## BIBLIOGRAFÍA

- ABC COMPONENTS LTD**, *Buzzer*. [en línea]. 2016. Disponible en: [http://www.farnell.com/datasheets/2032394.pdf?\\_ga=2.101101826.2090848696.1560227253-639472497.1559660386](http://www.farnell.com/datasheets/2032394.pdf?_ga=2.101101826.2090848696.1560227253-639472497.1559660386).
- AGUILAR, E.**, *Identificación en tiempo real de personas en posesión de pequeñas armas dentro de un ambiente video-vigilado* [en línea]. (tesis) (pregrado). Universidad de las Fuerzas Armadas. Sangolquí, Ecuador. 2018. [Consulta: 17 abril 2019]. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/15377>.
- AL-NOURI, L. y AL-ISAMI, S.**, *Baby walker injuries* [en línea]. (tesis) (pregrado). University of Baghdad. Iraq. 2006. [Consulta: 14 noviembre 2018]. Disponible en: <http://www.ncbi.nlm.nih.gov/pubmed/16494707>.
- ALTRONICS**, *Raspberry PowerPack V2.0 Lithium Battery*. [en línea]. 2019. Disponible en: <https://altronics.cl/powerpack-v12-respaldo-bateria>.
- ALVAREZ, M.A.**, *Qué es Python*. [en línea]. 2003. Disponible en: <https://desarrolloweb.com/articulos/1325.php>.
- ARDUINO**, *Arduino Mega 2560 Rev3*. [en línea]. 2019. Disponible en: <https://store.arduino.cc/usa/mega-2560-r3>.
- ARDUINO PRODUCTS**, *Arduino Products*. [en línea]. 2019. Disponible en: <https://www.arduino.cc/en/Main/Products>.
- ARÉVALO, V., GONZÁLEZ, J. y AMBROSIO, G.**, *La librería de visión artificial OPENCV*. [en línea]. España: 2016. [Consulta: 21 noviembre 2018]. Disponible en: [https://www.researchgate.net/publication/236668252\\_La\\_Libreria\\_de\\_Vision\\_Artificial\\_OpenCV\\_Aplicacion\\_a\\_la\\_Docencia\\_e\\_Investigacion\\_in\\_spanish](https://www.researchgate.net/publication/236668252_La_Libreria_de_Vision_Artificial_OpenCV_Aplicacion_a_la_Docencia_e_Investigacion_in_spanish).
- AV ELECTRONICS**, *Batería Lipo Turnigy 7.4V 1000mAh 2S 30C | AV Electronics*. [en línea]. 2019. Disponible en: <https://avelectronics.cc/producto/bateria-lipo-turnigy-7-4v-1000mah-2s-30c/>.
- BEDÓN, L.F.**, *Sistema de control para la movilidad y extensión de una silla de ruedas Eléctrica de bipedestación* [en línea]. (tesis) (pregrado). Universidad Técnica de Ambato.

Ambato, Ecuador. 2017. Disponible en: <http://repo.uta.edu.ec/handle/123456789/26094>.

**CÁRDENAS, G., GÁLVEZ, F., RUANO, K. y LAGOS, D.G.**, Efectos del uso del caminador en el desarrollo de los bebés. *Boletín* [en línea], vol. 4, no. 1. 2017. Disponible en: <http://ojseditorialumariana.com/index.php/BoletinInformativoCEI/article/view/1276/1217>.

**CAYAMBE, I. y CASTAÑEDA, J.**, *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. [Consulta: 22 mayo 2019]. Disponible en: <http://dspace.esPOCH.edu.ec/bitstream/123456789/6036/1/108T0171.pdf>.

**CHALLENGER, I., DÍAZ, Y. y BECERRA, R.**, Lenguaje de Programación Python. *Ciencias Holguín* [en línea], vol. XX. 2014. Disponible en: <http://www.ciencias.holguin.cu/index.php/cienciasholguin/article/view/826/887>.

**CORONA, L., ABARCA, G. y MARES, J.**, *Sensores y actuadores*. [en línea]. 1st. Mexico: Grupo Editorial Patria. 2014. [Consulta: 12 marzo 2019]. ISBN 9786074389364. Disponible en: [https://books.google.es/books?hl=es&lr=&id=wMm3BgAAQBAJ&oi=fnd&pg=PP1&dq=sensores+de+proximidad&ots=6N6lhw6\\_Xt&sig=uNMG\\_7JZEVOdm1a37bD6kFwRCtM#v=onepage&q=sensores de proximidad&f=false](https://books.google.es/books?hl=es&lr=&id=wMm3BgAAQBAJ&oi=fnd&pg=PP1&dq=sensores+de+proximidad&ots=6N6lhw6_Xt&sig=uNMG_7JZEVOdm1a37bD6kFwRCtM#v=onepage&q=sensores de proximidad&f=false).

**COSTA, A. y FERNÁNDEZ, J.**, *La imagen digital* [en línea]. España: s.n. 2005. [Consulta: 8 marzo 2019]. Disponible en: [http://www.revistadeortodoncia.com/files/2005\\_35\\_3\\_255-266.pdf](http://www.revistadeortodoncia.com/files/2005_35_3_255-266.pdf).

**DE LOS COBOS, S., GODDARD, J., PÉREZ, B. y GUTIÉRREZ, M.**, *Máquinas de Vector de Soporte para Clasificación* [en línea]. Mexico: Memorias XIV Foro Nacional de Estadística. 2000. [Consulta: 26 abril 2019]. ISBN 970-13-3051-X. Disponible en: <https://books.google.com.ec/books?hl=es&lr=&id=93fdDAAAQBAJ&oi=fnd&pg=PA25&dq=vsm+maquinas+de+vector+de+soporte&ots=hnYGY7SnK0&sig=8rVKWHG4TpWCAwPIVXPrSEulfG8#v=onepage&q=vsm maquinas de vector de soporte&f=false>.

**ELEC FREAKS**, Ultrasonic Ranging Module HC-SR04. [en línea]. S.l.: 2019. Disponible en: <https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf>.

**ELECTRÓNICOS CALDAS MG90S**, *Micro servomotor MG90S*. [en línea]. 2019. Disponible en: [https://www.electronicoscaldas.com/motores-y-servos/451-micro-servo-motor-mg90s.html?search\\_query=MG+90S&results=15](https://www.electronicoscaldas.com/motores-y-servos/451-micro-servo-motor-mg90s.html?search_query=MG+90S&results=15).

**ELECTRÓNICOS CALDAS MG995**, *Servomotor MG995*. [en línea]. 2019. Disponible en: <https://www.electronicoscaldas.com/motores-y-servos/608-servo-motor-mg995.html>.

- ELECTRÓNICOS CALDAS MG996R**, *Servomotor MG996R*. [en línea]. 2019. Disponible en: [https://www.electronicoscaldas.com/motores-y-servos/546-servo-motor-mg996r.html?search\\_query=mg996r&results=1](https://www.electronicoscaldas.com/motores-y-servos/546-servo-motor-mg996r.html?search_query=mg996r&results=1).
- ELTECHS**, *The ultimate Guide to Raspbian and other Raspberry Pi Software*. [en línea]. 2017. Disponible en: <https://eltechs.com/raspbian-and-other-raspberry-pi-software/>.
- ENRÍQUEZ, R.**, Guía de Usuario de Arduino. [en línea]. Córdoba, España: 2009. [Consulta: 29 mayo 2019]. Disponible en: [http://electroship.com/documentos/Arduino\\_user\\_manual\\_es.pdf](http://electroship.com/documentos/Arduino_user_manual_es.pdf).
- EUROPEAN COMMISSION**, *La Comisión Europea adopta una norma de la UE para evitar que los andadores causen accidentes*. [en línea]. 2009. [Consulta: 8 enero 2019]. Disponible en: [http://europa.eu/rapid/press-release\\_IP-09-45\\_es.htm?locale=en](http://europa.eu/rapid/press-release_IP-09-45_es.htm?locale=en).
- FERNÁNDEZ, J. y DÍAZ, F.**, Manual de prácticas de CAD utilizando el programa SolidWorks 2014. [en línea]. México: 2015. [Consulta: 10 marzo 2019]. Disponible en: [http://olimpia.cuautitlan2.unam.mx/pagina\\_ingenieria/mecanica/mat/mat\\_mec/m9/MANUAL\\_DE\\_PRACTICAS\\_DE\\_SOLIDWORKS.pdf](http://olimpia.cuautitlan2.unam.mx/pagina_ingenieria/mecanica/mat/mat_mec/m9/MANUAL_DE_PRACTICAS_DE_SOLIDWORKS.pdf).
- FONSECA, J.**, Batteries And Lithium-Ion. *Journal Boliviano de Ciencias* [en línea], vol. 8, pp. 60. 2011. Disponible en: <https://es.scribd.com/document/104016334/Celdas-Baterias-y-Pilas-de-Litio>.
- FU, K., GONZÁLEZ, R. y LEE, C.**, *Robótica Control, Detección, Visión e Inteligencia*. McGraw-Hil. México: s.n. 1989. ISBN 0-07-022625-3.
- GARCÍA, L. y CHÁVEZ, F.**, *Prototipo de Reconocimiento de Imágenes para la Recomendación de Productos para Supermercados* [en línea]. (tesis) (pregrado). Instituto Politécnico Nacional. México. 2016. [Consulta: 22 mayo 2019]. Disponible en: [https://tesis.ipn.mx/xmlui/bitstream/handle/123456789/21396/Prototipo de reconocimiento de imágenes para la recomendación de productos para supermercados sMart.pdf?sequence=5&isAllowed=y](https://tesis.ipn.mx/xmlui/bitstream/handle/123456789/21396/Prototipo%20de%20reconocimiento%20de%20im%C3%A1genes%20para%20la%20recomendaci%C3%B3n%20de%20productos%20para%20supermercados%20sMart.pdf?sequence=5&isAllowed=y).
- GONZÁLEZ, A.**, *Clever Data ¿Qué es Machine Learning?* [en línea]. 2014. Disponible en: <https://cleverdata.io/que-es-machine-learning-big-data/>.
- GUAMUSHIG, J.L. y JEREZ, M.A.**, *Diseño y construcción de un andador inteligente para el desplazamiento autónomo de los adultos mayores con visión reducida y problemas de movilidad del hogar de vida «Luis Maldonado Tamayo»; mediante la investigación de técnicas de visión artificial* [en línea]. (tesis) (pregrado). Universidad de las Fuerzas Armadas ESPE. Latacunga, Ecuador. 2018. Disponible en:

<http://repositorio.espe.edu.ec/handle/21000/14030>.

**HERRERA, J.M. y SÁNCHEZ, L.M.,** *Numpy: Computación científica con Python para módulos de evaluación continua en asignaturas de ciencias aplicadas.* [en línea]. 2013. Disponible en: <https://es.scribd.com/document/396274000/4-1-Numpy-Computacion-Cientifica-Con-Python-Para-Modulos-de-Evaluacion-Continua-en-Asignaturas-de-Ciencias-Aplicadas>.

**HIDALGO, L.A.,** *Inteligencia Artificial y Sistemas Expertos* [en línea]. España: Servicio de Publicaciones Universidad de Córdoba. 1996. ISBN 9788478013463. Disponible en: <https://es.scribd.com/document/393467719/Luis-Amador-Inteligencia-artificial-1996-1-2-pdf>.

**KARA, S., ESRA, A., GÜMÜS, D. y CATAL, F.,** Actitud de los pediatras sobre el uso de andadores para los bebés. *Archivos argentinos de pediatría* [en línea], vol. 111, no. 6, pp. 528-536. 2013. ISSN 03250075. DOI 10.5546/aap.2013.528. Disponible en: <http://www.sap.org.ar/docs/publicaciones/archivosarg/2013/v111n6a11.pdf>.

**LA SERNA, N. y ROMÁN, U.,** Revista de investigación de sistemas e informática. Técnicas de Segmentación en Procesamiento Digital de Imágenes. *Revista de investigación de Sistemas e Informática* [en línea]. Lima, Perú: 2014. Disponible en: [http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/risi/2009\\_n2/v6n2/a02v6n2.pdf](http://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/risi/2009_n2/v6n2/a02v6n2.pdf).

**LOGITECH C170,** *Webcam C170.* [en línea]. 2011. Disponible en: [https://support.logitech.com/es\\_es/product/webcam-c170/specs](https://support.logitech.com/es_es/product/webcam-c170/specs).

**LOJANO, I.A. y SACTA, M.A.,** *Prevalencia de las alteraciones del desarrollo motor en niños y niñas de 0 a 1 año de edad de consulta externa del Centro de Salud No 2 de la ciudad de Cuenca, agosto 2009-enero 2010: relación de las alteraciones del desarrollo motor con las prácticas de* [en línea]. (tesis) (pregrado). Universidad de Cuenca. Cuenca, Ecuador. 2010. [Consulta: 15 noviembre 2018]. Disponible en: <http://dspace.ucuenca.edu.ec/handle/123456789/3888>.

**MANOSALVAS, C.,** *Diseño de un sistema embebido para el control de ingreso y salida de vehículos a través de internet, en el acceso principal de la ESPOCH.* [en línea]. (tesis) (pregrado). Escuela Superior Politécnica de Chimborazo. Riobamba, Ecuador. 2017. Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/7621>.

**MEJÍA, J.,** *Procesamiento Digital de Imágenes.* [en línea]. México: 2004. Disponible en: [http://laurence.com.ar/artes/comun/Apuntes procesamiento digital de imagenes.pdf](http://laurence.com.ar/artes/comun/Apuntes%20procesamiento%20digital%20de%20imagenes.pdf).

**MÍNGUEZ, J.,** *Silla de ruedas guiada por el cerebro y un andador inteligente para personas*

- dependientes. [en línea]. 2009. Disponible en:*  
<https://www.europapress.es/epsocial/responsables/noticia-presentan-silla-ruedas-guiada-cerebro-andador-inteligente-personas-dependientes-20090522150921.html>.
- NILSSON, N.**, *Inteligencia Artificial: Una nueva síntesis*. España: s.n. 2001. ISBN 84-481-2824-9.
- OMS y UNICEF**, Informe mundial sobre prevención de las lesiones en los niños. *Organización Mundial de la Salud [en línea]*, pp. 48. 2012. Disponible en:  
[https://apps.who.int/iris/bitstream/handle/10665/77762/WHO\\_NMH\\_VIP08.01\\_spa.pdf;jsessionid=D24483517229A42B57EDFC37E4FF708B?sequence=1](https://apps.who.int/iris/bitstream/handle/10665/77762/WHO_NMH_VIP08.01_spa.pdf;jsessionid=D24483517229A42B57EDFC37E4FF708B?sequence=1).
- ORTOPEDIA**, *Andadores. [en línea]. [sin fecha]. Disponible en:*  
<https://ortopedia1.com/andadores>.
- ORTUÑO, J.J.**, *Detección e identificación visual de caracteres en productos industriales [en línea]. (tesis) (pregrado)*. Universidad de Alicante. Alicante, España. 2016. Disponible en:  
<http://hdl.handle.net/10045/57289>.
- PAJARES, G. y DE LA CRUZ, J.**, *Visión por computador Imágenes digitales y aplicaciones*. Madrid, España: RA-MA. 2002. ISBN 84-7897-472-5.
- PAREDES, E.K. y TOAPANTA, J.E.**, *Diseño e implementación de un bipedestador automático para la movilidad de adultos mayores [en línea]. (tesis) (pregrado)*. Universidad de las Fuerzas Armadas ESPE. Latacunga, Ecuador. 2018. Disponible en:  
<http://repositorio.espe.edu.ec/handle/21000/13995>.
- PÉREZ, J. y CUEVAS, J.**, *Andadores: Familia y Salud. [en línea]. 2012. Disponible en:*  
<http://www.familiaysalud.es/podemos-prevenir/prevencion-de-accidentes/como-prevenir/andadores>.
- RASPBERRY**, *Puertos GPIO. [en línea]. 2016. [Consulta: 22 mayo 2019]. Disponible en:*  
<https://prototipopi.wordpress.com/2016/05/02/puertos-gpio/>.
- RASPBERRY PI**, *Products Archive - Raspberry Pi. [en línea]. 2019. Disponible en:*  
<https://www.raspberrypi.org/products/>.
- SANHUEZA, O.**, *Análisis y Detección de Características Faciales Usando Aprendizaje Estadístico [en línea]. (tesis) (pregrado)*. Universidad de Chile. Chile. 2008. [Consulta: 24 abril 2019]. Disponible en: <http://repositorio.uchile.cl/handle/2250/103144>.
- SOCIEDAD AMERICANA DE PEDIATRÍA**, *Los andadores para bebés: una elección peligrosa. [en línea]. 2018. Disponible en:*

<https://www.healthychildren.org/Spanish/safety-prevention/at-home/Paginas/Baby-Walkers-A-Dangerous-Choice.aspx>.

**SOLANO, G.,** *Diseño y construcción de un dispositivo portátil para la identificación de billetes orientado a personas con discapacidad visual mediante el uso de visión artificial* [en línea]. (tesis) (pregrado). Escuela Politécnica de Chimborazo. Riobamba, Ecuador. 2018. [Consulta: 18 enero 2019]. Disponible en: <http://repositorio.uchile.cl/handle/2250/103144>.

**SOMOLINOS, J.A.,** *Avances en Robótica y Visión en Computador* [en línea]. 1st. Cuenca, España: s.n. 2002. ISBN 84-8427-199-4. Disponible en: <https://books.google.com.ec/books?id=V-eXwzEIngQC&printsec=frontcover&dq=somolinos+jose+andres++avances+en+robotica&hl=es-419&sa=X&ved=0ahUKEwi45cTM5-HiAhUEuVkkHeJDB58Q6AEIKDAA#v=onepage&q=somolinos+jose+andres+avances+en+robotica&f=false>.

**SUCAR, L.E. y GÓMEZ, G.,** *Procesamiento de Imágenes y Visión Computacional*. [en línea]. México: 2002. Disponible en: <https://ccc.inaoep.mx/~esucar/Clases-van/van1-introduccion.pdf>.

**TESSEL,** *Tessel 2 features*. [en línea]. 2018. Disponible en: <https://tessel.io/>.

**TIPPENS, P.,** *Física I, Conceptos y aplicaciones*. Bogotá: s.n. 2009. ISBN 978-958-41-0391-8.

**TOBAR, A.,** *Reconocimiento de señales de tráfico mediante Raspberry* [en línea]. (tesis) (pregrado). Universidad de Burgos. Burgos, España. 2017. [Consulta: 26 abril 2019]. Disponible en: <http://hdl.handle.net/10259/4639>.

**TRIVIÑO, X., BEDREGAL, P., AZÓCAR, M., VALENZUELA, P. y GONZÁLEZ, C.,** *Uso de andador en lactantes TT - Infant walkers. Rev Chil Pediatr* [en línea], 1997. Disponible en: <https://scielo.conicyt.cl/pdf/rcp/v68n6/art02.pdf>.

**UNE-EN 1273,** *Artículos de puericultura. Andadores. Requisitos de seguridad y métodos de ensayo*. España. 2005.

**UNIÓN EUROPEA,** *Visión Artificial*. [en línea]. España: 2012. [Consulta: 8 marzo 2019]. Disponible en: <http://visionartificial.fpcat.cat/wp-content/uploads/Conocimientos.pdf>.

**UTRERA, R.,** *Uso de algoritmos de aprendizaje automático aplicados a bases de datos genéticos*. [en línea]. España: 2017. Disponible en: <http://openaccess.uoc.edu/webapps/o2/bitstream/10609/65426/6/rgagoTFM0617memoria.pdf>.

**VALERO, J., BONILLA, Y., SANDOBAL, C. y DUQUE, C.,** Control de una silla r botica a trav s de comandos de voz. [en l nea]. Venezuela: 2012. Disponible en: [https://www.researchgate.net/publication/312531914\\_CONTROL\\_DE\\_UNA\\_SILLA\\_ROBOTICA\\_A\\_TRAVES\\_DE\\_COMANDOS\\_DE\\_VOZ](https://www.researchgate.net/publication/312531914_CONTROL_DE_UNA_SILLA_ROBOTICA_A_TRAVES_DE_COMANDOS_DE_VOZ).

**VELASTEGUI, K.J.,** *El uso del andador en el desarrollo de las destrezas motoras en ni os de 10 a 12 meses* [en l nea]. (tesis) (pregrado). Universidad T cnica de Ambato. Ambato, Ecuador. 2018. Disponible en: <http://repositorio.uta.edu.ec/jspui/handle/123456789/27732>.

**V LEZ, J., MORENO, A., S NCHEZ,  . y S NCHEZ, J.,** *Visi n por Computador* [en l nea]. 2nd. Madrid, Espa a: Dykinson. 2003. ISBN 9788497720694. Disponible en: <https://www.casadellibro.com/libro-vision-por-computador/9788497720694/898645>.