



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

**“IMPLEMENTACIÓN DE UN JUGUETE INTELIGENTE PARA LA
ENSEÑANZA DE NIÑOS DE 2 A 7 AÑOS, USANDO UNA TÉCNICA
DE SOFT COMPUTING”**

TRABAJO DE TITULACIÓN: DISPOSITIVO TECNOLÓGICO

Para optar al grado académico de:

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

**AUTORES: LISSETTE ESTEFANÍA SALINAS SALINAS
EDISON MARCELO CHAMORRO CARRERA**

TUTOR: ING. PABLO EDUARDO LOZADA YÁNEZ

Riobamba - Ecuador

2017

©2017, Lissette Estefanía Salinas Salinas y Edison Marcelo Chamorro Carrera

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 del Autor.

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

El Tribunal de Trabajo de Titulación certifica que: El trabajo de investigación “IMPLEMENTACIÓN DE UN JUGUETE INTELIGENTE PARA LA ENSEÑANZA DE NIÑOS DE 2 A 7 AÑOS, USANDO UNA TÉCNICA DE SOFT COMPUTING”, de responsabilidad de la señorita Lissette Estefanía Salinas Salinas y el señor Edison Marcelo Chamorro Carrera, ha sido minuciosamente revisado por los Miembros del Tribunal del Trabajo de Titulación, quedando autorizada su presentación.

NOMBRE	FIRMA	FECHA
Ing. Washington Luna E. DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
Ing. Freddy Chávez V. DIRECTOR EIE CONTROL Y REDES INDUSTRIALES
Ing. Pablo Lozada Y. DIRECTOR DE TRABAJO DE TITULACIÓN
Ing. Edwin Altamirano S. MIEMBRO DEL TRIBUNAL

Nosotros, **LISSETTE ESTEFANÍA SALINAS SALINAS** y **EDISON MARCELO CHAMORRO CARRERA**, somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de Titulación, y el patrimonio intelectual de la misma pertenece a la **ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**.

Lissette Estefanía Salinas Salinas

Edison Marcelo Chamorro Carrera

DEDICATORIA

Este logro se lo dedico a toda mi familia, de manera especial a mi madre, ejemplo de valentía y superación, amiga incondicional que siempre estuvo al pendiente de mí. A todos mis amigos, quienes se convirtieron en mi segunda familia y juntos compartimos muchas experiencias y anécdotas que se quedarán grabadas por siempre en mi mente.

LISSETTE

Este triunfo se lo dedico a toda mi familia, quienes han estado conmigo en los buenos y malos momentos. Especialmente a mi madre Lilia Carrera y abuelitos, que me guiaron por el camino del bien, además de su apoyo moral y económico para poder culminar esta meta en mi vida.

EDISON

AGRADECIMIENTO

En primer lugar queremos dar gracias a Dios por la vida, la salud, sabiduría y demás bendiciones, agradecemos a nuestros familiares y amigos incondicionales que nos han apoyado en este largo caminar.

LISSETTE Y EDISON

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS.....	x
ÍNDICE DE FIGURAS.....	xi
ÍNDICE DE GRÁFICOS.....	xiv
ÍNDICE DE ANEXOS.....	xv
ÍNDICE DE ABREVIATURAS.....	xvi
RESUMEN.....	xviii
SUMMARY.....	xix
INTRODUCCIÓN.....	1
CAPÍTULO I	
1. MARCO TEÓRICO REFERENCIAL.....	6
1.1 Juguetrónica.....	6
1.2 Inteligencia Artificial.....	6
1.3 Soft Computing.....	7
1.3.1 Métodos que integran la Soft Computing.....	9
1.3.1.1 Modelo Probabilístico.....	9
1.3.1.2 Lógica Difusa.....	10
1.3.1.3 Redes Neuronales Artificiales.....	11
1.3.1.4 Sistemas evolutivos.....	15
1.4 Algoritmo AdaBoost.....	16
1.4.1 Árboles de decisiones para la clasificación.....	18
1.5 Reconocimiento facial con el método de Viola-Jones.....	19
1.5.1 Características faciales.....	20
1.5.2 Clasificadores cascada Haar.....	20
1.5.2.1 Imagen Integral.....	21
1.5.3 Entrenamiento.....	22
1.6 Visión Artificial.....	23
1.6.1 Adquisición de la imagen.....	23
1.6.2 Digitalización de imágenes.....	24
1.6.2.1 Modelo RGB.....	24
1.6.2.2 Modelo HSV.....	25
1.6.3 Procesamiento de imágenes.....	25
1.6.3.1 Transformaciones morfológicas.....	26
1.6.3.2 Transformaciones geométricas.....	27
1.6.3.3 Operaciones sobre histograma de una imagen.....	27

1.6.4	<i>Detección de contornos</i>	27
1.7	Fundamentos de robótica	28
1.7.1	<i>Inteligencia del Robot</i>	29
1.7.2	<i>Sistema sensorial</i>	29
1.7.3	<i>Actuadores</i>	29
1.7.3.1	<i>Motorreductores</i>	29
1.7.4	<i>Estructura del robot</i>	30
1.7.5	<i>Fuente de Alimentación</i>	30
1.7.5.1	<i>Baterías</i>	31
1.8	Minicomputador Raspberry Pi	31
1.9	Arduino	32
1.10	Software para desarrollo de programas y modelamiento del prototipo	33
1.10.1	<i>Sistema Operativo Raspbian Jessie</i>	33
1.10.2	<i>OpenCv</i>	33
1.10.3	<i>Python</i>	34
1.10.4	<i>SolidWorks</i>	34
1.10.5	<i>Software Arduino</i>	35
CAPÍTULO II		
2.	MARCO METODOLÓGICO	36
2.1	Instalación de programas en Raspberry Pi	36
2.1.1	<i>Instalación del Sistema Operativo Raspbian Jessie</i>	36
2.1.2	<i>Instalación de Python</i>	38
2.1.3	<i>Instalación de OpenCv</i>	38
2.2	Selección de elementos	40
2.2.1	<i>Selección de motores</i>	40
2.2.2	<i>Selección de cámara</i>	40
2.2.3	<i>Selección de sensores</i>	42
2.2.4	<i>Selección de baterías</i>	42
2.2.4.1	<i>Duración de la batería</i>	43
2.3	Diseño e implementación de prototipo del juguete inteligente	44
2.3.1	<i>Diseño del prototipo</i>	44
2.4	Diseño del sistema de visión artificial	47
2.4.1	<i>Sistema de reconocimiento facial</i>	49
2.4.2	<i>Sistema de reconocimiento de colores y figuras geométricas</i>	53
2.5	Sistema para control de movimientos	56
2.5.1	<i>Sistema de locomoción</i>	56
2.5.2	<i>Sistema sensorial</i>	57

2.6	Integración de hardware y software	59
2.7	Interacción lúdica y cognitiva.....	61
2.8	Funcionamiento del juguete inteligente	62
CAPÍTULO III		
3.	MARCO DE ANÁLISIS DE RESULTADOS	64
3.1	Pruebas de funcionalidad.....	64
3.1.1	<i>Evaluación del sistema de reconocimiento facial</i>	64
3.1.2	<i>Pruebas del sistema de reconocimiento de colores y figuras 2D</i>	67
3.1.3	<i>Pruebas del sistema de control de movimientos</i>	68
3.1.3.1	<i>Sistema sensorial.....</i>	68
3.1.3.2	<i>Control de giros</i>	69
3.2	Pruebas de interacción con niños	70
CONCLUSIONES.....		76
RECOMENDACIONES.....		77
BIBLIOGRAFÍA		
ANEXOS		

ÍNDICE DE TABLAS

Tabla 1-2: Principales librerías de OpenCv instaladas.....	39
Tabla 2-2: Tiempos de reacción de cámaras.....	41
Tabla 3-2: Ventajas y desventajas de las baterías lipo.....	42
Tabla 4-2: Ventajas y desventajas de los métodos que integran Soft Computing.....	47
Tabla 5-2: Aplicaciones de los métodos que integran Soft Computing.....	48
Tabla 6-2: Subprogramas y funciones que aportan al sistema de reconocimiento facial.....	50
Tabla 7-2: Composición del sistema para identificar el color y forma.....	54
Tabla 8-2: Descripción de la GUI.....	62
Tabla 1-3: Valores de umbral predictivo del rostro de Leandro.....	64
Tabla 2-3: Valores del umbral predictivo del rostro de Samantha.....	65
Tabla 3-3: Resultados de efectividad de la red neuronal.....	66
Tabla 4-3: Control de velocidad de los motores para giros.....	70
Tabla 5-3: Resultados de aceptación de Pushaq.....	71
Tabla 6-3: Nivel de interés por edades, datos numéricos.....	72
Tabla 7-3: Nivel de interés por edades, datos porcentuales.....	72
Tabla 8-3: Juicio valorativo por parte de docentes y personal de DCE.....	74

ÍNDICE DE FIGURAS

Figura 1-1: Componentes de la Soft Computing.....	8
Figura 2-1: Ejemplo de red bayesiana.....	9
Figura 3-1: Ejemplo de cadena o red de Markov.....	10
Figura 4-1: Fuzzificación.....	11
Figura 5-1: Estructura de una Red Neuronal Artificial.....	12
Figura 6-1: Esquema de un modelo neuronal.....	13
Figura 7-1: Función AND en una red neuronal.....	14
Figura 8-1: Función OR en una red neuronal.....	14
Figura 9-1: Función XOR en una red neuronal.....	14
Figura 10-1: Esquema de aprendizaje perceptrónico.....	14
Figura 11-1: Esquema de Sistemas Evolutivos.....	15
Figura 12-1: Operador cruce en dos puntos.....	15
Figura 13-1: Operador mutación con intercambio de genes.....	16
Figura 14-1: Diagrama de flujo del algoritmo AdaBoost.....	18
Figura 15-1: Estructura de árbol de decisiones tipo CART.....	19
Figura 16-1: Características Haar.....	20
Figura 17-1: Imagen integral.....	21
Figura 18-1: Convolución filtro Haar con imagen integral.....	22
Figura 19-1: Clasificador cascada tipo haar.....	22
Figura 20-1: Esquema general de VA.....	23
Figura 21-1: Imagen Digital.....	24
Figura 22-1: Combinaciones del modelo RGB.....	24
Figura 23-1: Representación del modelo HSV.....	25
Figura 24-1 Aumento de contraste: (a) imagen captada.....	26
Figura 24-1 Aumento de contraste: (b) ecualización de rango de grises.....	26
Figura 25-1: Dilatación de una imagen binaria.....	26
Figura 26-1: Erosión de una imagen binaria.....	26
Figura 27-1: Imagen con histograma.....	27
Figura 28-1: Detección de contorno de una imagen.....	28
Figura 29-1: Componentes de un robot.....	28
Figura 30-1: Principio de funcionamiento de los motorreductores.....	30
Figura 31-1: Minicomputador Raspberry Pi 3 Modelo B.....	32
Figura 32-1: Placa Arduino Mega.....	32

Figura 1-2: Configuración del SO Raspbian Jessie.....	37
Figura 2-2: Escritorio de Raspberry Pi.....	37
Figura 3-2: Ventana Shell y editor de Python.....	38
Figura 4-2: Librerías de OpenCv.....	39
Figura 5-2: Motor eleva vidrios Nemesis.....	40
Figura 6-2: Módulo de cámara Raspberry Pi.....	41
Figura 7-2: Sensor ultrasónico HC-SR04.....	42
Figura 8-2: Batería Lipo Turnigy 3S.....	43
Figura 9-2: Ensamble del robot.....	44
Figura 10-2: Chasis del robot.....	44
Figura 11-2: Cuerpo del robot.....	45
Figura 12-2: Tapa posterior del cuerpo.....	45
Figura 13-2: Cabeza del robot.....	45
Figura 14-2: Impresión 3D de piezas modeladas en Solidworks.....	46
Figura 15-2: Juguete inteligente implementado.....	46
Figura 16-2: Diagrama para el reconocimiento facial.....	49
Figura 17-2: Subprograma de configuración.....	51
Figura 18-2: Clasificador cascada en OpenCV.....	51
Figura 19-2: Conversión de imagen normal a imagen integral.....	51
Figura 20-2: Captura de imágenes positivas para la base de datos.....	52
Figura 21-2: Función para iterar archivos.....	52
Figura 22-2: Código para hallar la media de los rasgos faciales.....	53
Figura 23-2: Cálculo del umbral predictivo.....	53
Figura 24-2: Sistema para reconocimiento de colores y formas.....	54
Figura 25-2: Definición los colores a ser identificados.....	55
Figura 26-2: Eliminación el ruido.....	55
Figura 27-2: Determina el color y figura de la imagen.....	56
Figura 28-2: Locomoción del juguete inteligente.....	57
Figura 29-2: Diagrama de flujo para el control de movimientos.....	58
Figura 30-2: Programa para control de movimientos.....	58
Figura 31-2: Diagrama de conexión en la Raspberry Pi.....	59
Figura 32-2: Circuito del regulador de tensión DC.....	60
Figura 33-2: Diagrama de control de movimientos.....	60
Figura 34-2: Interfaz gráfica de Pushaq.....	61
Figura 35-2: Interfaz gráfica para el Técnico.....	63
Figura 1-3: Fotografías captadas con picam.....	66

Figura 2-3: Filtro en tiempo real para hallar el rango de colores.....	67
Figura 3-3: Distancias sensadas por HC-SR04.....	68
Figura 4-3: Presentación de Pushaq en la escuela “Archipiélago de Galápagos”.....	70
Figura 5-3: Pushaq narrando cuentos infantiles.....	71
Figura 6-3: Niños encantados con Pushaq levantando la mano.....	71
Figura 7-3: Adrián Carrasco saludando a Pushaq.....	75

ÍNDICE DE GRÁFICOS

Gráfico 1-3: Influencia de luminosidad y pose en el reconocimiento facial.....	65
Gráfico 2-3: Comportamiento de los sensores HC-SR04.....	69
Gráfico 3-3: Nivel de interés por edades.....	73
Gráfico 4-3: Nivel de interés global.....	74

ÍNDICE DE ANEXOS

ANEXO A: Manual de minicomputador Raspberry Pi 3 modelo B

ANEXO B: Disposición de pines GPIO en placa Raspberry Pi 3 B+

ANEXO C: Especificaciones técnicas del módulo de la cámara Raspberry Pi

ANEXO D: Datasheet de driver de motor Sabertooth dual 12A

ANEXO E: Datasheet del sensor ultrasónico HC-SR04

ANEXO F: Manual de usuario del juguete inteligente Pushaq

ANEXO G: Manual de técnico del juguete inteligente Pushaq

ÍNDICE DE ABREVIATURAS

2D	Dos dimensiones
CAD	Diseño Asistido por Computadora
CART	Árboles de Clasificación de Regresión
cm	Centímetros
DC	Corriente Continua
E/S	Entradas y Salidas
GPIO	General Purpose Input or Output (Entrada o Salida de Propósito General)
GUI	Interfaz Gráfica de Usuario (Graphical User Interface)
HD	High Definition (Alta Definición)
HSV	Hue, Saturation, Value (Matriz, Saturación, Valor)
Hz	Hercio
IA	Inteligencia Artificial
JI	Juguete Inteligente
KHz	Kilo Hercios
LDA	Análisis Lineal Discriminante
mA	Miliamperios
mAh	Miliamperios por hora
MH	Metaheurística
ms	Milisegundos
mm	Milímetros
PCA	Análisis de Componentes Principales (Principal Component Analysis)
PG	Programación Genética
PLA	Plástico de Ácido Poliláctico
PWM	Pulse Width Modulation (Modulación de ancho de pulso)
RNA	Red Neuronal Artificial

RGB	Red, Green, Blue (Rojo, verde y azul)
RPM	Revoluciones por minuto
SBC	Simple Board Computer (Computador de Placa Simple)
SD	Secure Digital (Formato de tarjeta de memoria)
SO	Sistema Operativo
TIC	Tecnología de la Información y Comunicación
UC	Microcontrolador
USB	Universal Serial Bus (Bus universal en serie)
V	Voltios
VA	Visión Artificial
W	Vatios
XML	Lenguaje de Marca Extensible

RESUMEN

El presente trabajo de titulación tuvo como objetivo implementar un juguete inteligente para la interacción lúdica y cognitiva para niños en etapa pre operacional, creando ambientes favorables para la percepción de colores y figuras geométricas. El dispositivo juguetrónico pone a disposición del usuario una interfaz gráfica con varias opciones en las que se incluyen cuentos, canciones, juegos, galería de imágenes y reconocimiento facial (Face ID) para activar el programa de enseñanza. El sistema de reconocimiento facial se basa en el uso de clasificadores cascada y algoritmo AdaBoost para entrenar una red neuronal, implementada como una adaptación del modelo propuesto por Paul Viola y Michael Jones. Se emplearon técnicas de Visión Artificial para determinar el contorno y color de objetos 2D en espacios controlados y un filtro de tonos en tiempo real para hallar los valores de las bandas espectrales en formato RGB. Toda esta información fue procesada en un minicomputador Raspberry Pi. También se implementó un sistema de locomoción diferencial que le permite al robot desplazarse sobre superficies planas junto a sensores que emiten señales a un procesador arduino para coordinar movimientos y planificar trayectorias. Los resultados obtenidos de la interacción con niños fueron satisfactorios, de una muestra de 192 niños el 98.44 % le agrado. Por otro lado, especialistas en psicología y educación calificaron al robot como un recurso didáctico innovador y que aporta a la educación especial ayudando a niños con deficiencia intelectual. Con la implementación del juguete inteligente se recomienda profundizar en la investigación que permita contribuir al levantamiento de las bases cognitivas.

PALABRAS CLAVE: <TECNOLOGÍA Y CIENCIAS DE LA INGENIERÍA>, <INTELIGENCIA ARTIFICIAL>, <SOFT COMPUTING>, <VISIÓN ARTIFICIAL>, <RED NEURONAL>, <ALGORITMO ADABOOST>, <JUGUETE INTELIGENTE>, <INTERACCIÓN LÚDICA>.

SUMMARY

The present titling work had as objective to implement an intelligent toy for the playful and cognitive interaction for children in pre-operational stage, creating favorable environments for the perception of colors and geometric figures. The playhead device provides users with a graphical interface with several options including stories, songs, games, image gallery and facial recognition (Face ID) to activate the teaching program. The face recognition system is based on the use of cascade classifiers and AdaBoost algorithm to train a neuronal network, implemented as an adaptation of the model proposed by Paul Viola and Michael Jones. Artificial Vision techniques were used to determine the contour and color of 2D objects in controlled spaces and a real-time tone filter to find the values of the spectral bands in RGB format. All this information was processed on a Raspberry Pi minicomputer. Also, a differential locomotion system was implemented that allows the robot to move on flat surfaces together with sensors that emit signals to an Arduino processor to coordinate movements and to plan trajectories. The results obtained from the interaction with children were satisfactory, out of a sample of 192 children, 98.44% liked it. On the other hand, specialists in psychology and education qualified the robot as an innovative didactic resource and that contributes to special education helping children with intellectual deficiency. With the implementation of the intelligent toy it is recommended to deepen the research that allows to contribute to the raising of the cognitive bases.

KEYWORDS: <TECHNOLOGY AND SCIENCE OF THE ENGINEERING>, <ARTIFICIAL INTELLIGENCE>, <SOFT COMPUTING>, <ARTIFICIAL VISION>, <NEURONAL NETWORK>, <ADABOOST ALGORITHM>, <INTELLIGENT TOY>, <PLAYFUL INTERACTION>.

INTRODUCCIÓN

Uno de los avances tecnológicos de mayor impacto es la imitación del ser humano tanto en su funcionamiento motriz como intelectual, para lo cual se ha combinado inteligencia artificial con robótica. Hoy en día a los robots se les asigna diversos usos en varias áreas, como en: la industria, educación, hogar, salud, exploración, etc.; tal es el caso de los autómatas asistenciales que realizan tareas domésticas y sirven como medio de entretenimiento. (Rentería & Rivas, 2000, p.2-5)

Según Olga Ballesteros (2011, pp.20-21) los sistemas robóticos aplicados a la educación inicial contribuyen al aprendizaje inductivo, desarrollo sensorial y psicomotriz, además de generar entornos para la adquisición de habilidades, competencias y destrezas. Aplicar este tipo de tecnología en la enseñanza de los más pequeños del hogar resulta sencillo, pues los niños se adaptan fácilmente a los cambios.

El aporte más reciente para la construcción de dispositivos juguetrónicos es la soft computing, una derivación de la Inteligencia Artificial (IA). Engloba una serie de técnicas computacionales para dar solución a problemas confusos, incompletos, difíciles de resolver con métodos convencionales de hard computing (Verdegay, 2005, p.30). En el artículo “*Detecting pedestrians using patterns of motion and appearance*” los autores proponen nuevos métodos para el reconocimiento de objetos empleando soft computing. (Viola & Jones, 2003)

En el presente trabajo se implementará un juguete inteligente didáctico para enseñar colores y figuras a niños de 2 a 7 años, mediante la interacción lúdica. Los conceptos básicos requeridos para la implementación son: electrónica, robótica, informática, soft computing, ingeniería del producto, sistemas embebidos, inteligencia y visión artificial

El prototipo resultado del presente trabajo de investigación se lo ha llamado robot Pushaq, mismo que puede enseñar, jugar, almacenar fotografías, recitar cuentos y reproducir canciones para que los niños se diviertan, creando ambientes favorables para generar conocimientos, a la vez servirá como robot psicopedagógico. Incluye un innovador sistema de reconocimiento facial (Face ID).

Pushaq se comunica por medio de sensores, pantallas, cámaras y altavoces, estos elementos emiten señales que son captadas del entorno al minicomputador para procesar dicha información y luego ser transmitidos a los dispositivos de salida o actuadores permitiendo al robot interactuar con el infante.

ANTECEDENTES

Actualmente vivimos en la era de la tecnología, inmersa en un mundo de equipos inteligentes que apuestan hacer la vida más placentera. Este gran avance de la humanidad promueve el adelanto e innovación. Introducir tecnología en niños a temprana edad es factible e importante para su formación. Es fundamental el desarrollo de dispositivos o aplicaciones que ayuden a la enseñanza de los infantes tanto en hogares como en centros educativos. (Bravo & Forero, 2012)

Una de las primeras tecnologías aplicadas al proceso de enseñanza fue la televisión, mediante programas educativos se impartía conocimiento de forma audiovisual diferente a la tradicional; luego aparecieron juguetes didácticos para aprender colores, figuras, sonidos, etc. (Gómez, 2012, p.3). Tiempo después con el avance de la informática y electrónica se crean un sin número de aplicaciones y dispositivos para la enseñanza y estimulación temprana.

Existen sistemas sofisticados que combinan software y hardware para la escolarización infantil, uno de estos es Robi un juguete de Fisher Price (2017) que reproduce más de 40 canciones, aprendizajes y bailes, activándole desde su barriga o pie. ENGKEY es un robot que enseña inglés a niños en las escuelas, diseñado por la empresa ABC.

Teebot (El Comercio, 2015, p.5) es un robot educativo ecuatoriano, brinda la oportunidad a niños entre 4 y 12 años aprender las bases de programación y electrónica, otros trabajos con similares características creados en Escuela Superior Politécnica de Chimborazo son el pingüino robot asistencial (Castañeda & Cayambe, 2016) para niños mediante visión artificial que juega a la pelota y un sistema educativo robótico controlado por sensor Kinect, para el seguimiento en tres dimensiones.

En el mercado encontramos varios robots educativos comerciales como: Dash&Dot, Lego Mindstorms, Zowi, Bee-Bot, Albert, Moway Scratch, etc. Además hoy en día a los robots humanoides se les programa para dictar clases en las escuelas de Japón, esto se evidencia en el artículo Exploring the Possibility of using Humanoid Robots as Instructional Tools for Teaching a Second Language in Primary School. (Chin-Wei & al, 2010)

FORMULACIÓN DEL PROBLEMA

¿Cómo un juguete inteligente usando un sistema de visión artificial, procesado con una técnica de Soft Computing podrá interactuar de forma lúdica y cognitiva con niños de 2 a 7 años?

SISTEMATIZACIÓN DEL PROBLEMA

1. ¿Cómo seleccionar una técnica de Soft Computing para el procesamiento inteligente de imágenes?
2. ¿Cómo diseñar un sistema de visión artificial para el tratamiento de imágenes de una escena captada?
3. ¿Cuáles son las características que se deben tomar en cuenta a la hora de diseñar y construir el juguete?
4. ¿Qué mecanismos se emplearán para la integración del sistema de visión artificial al prototipo?
5. ¿En base a qué parámetros se va a evaluar los resultados de la implementación del juguete inteligente?

JUSTIFICACIÓN

Justificación teórica

La robótica ha revolucionado el mundo, es común ver a robots realizando tareas cotidianas. Existen varios tipos como: médicos, asistenciales, terapéuticos, zoomórficos, humanoides, inmersos en todas las áreas. Cada vez estas máquinas se vuelven más inteligentes, capaces de tomar sus propias decisiones ante eventuales situaciones reales, gracias a las técnicas sofisticadas de inteligencia artificial y soft computing.

Una de las concepciones del siglo XXI es la construcción de nuevos modelos educativos, en la que exige la incorporación de medios tecnológicos. Creando un escenario accesible, complejo, global, flexible y cambiante; trascendental para formar el pensamiento crítico y creativo. La tecnología se convierte en un recurso didáctico dentro del proceso de aprendizaje-enseñanza, favoreciendo especialmente a los niveles básico y sin duda los más importantes ya que a temprana edad se forjan las bases cognitivas. (Gómez, 2012, pp.7-8)

Jean Piaget (1967, p.178) afirma que los niños aprenden a través de la exploración y manipulación, recomienda aplicar juegos simbólicos en etapa pre-operacional (inicia cuando el niño empieza hablar, generalmente a los 2 años y se extiende a los 7 años); este tipo de juego ayuda a retener información de imágenes pasadas en el cerebro y asociar a la realidad. El juego en general es fundamental para el desarrollo del conocimiento humano.

Para el diseño y construcción de juguetes robóticos se aplican conceptos y principios básicos de electrónica, mecánica, informática e inteligencia artificial, ya que sin el aporte de estas ciencias sería imposible crear un prototipo capaz de movilizarse, interpretar la información del mundo e interactuar con personas.

Justificación aplicativa

Se desarrollará un juguete inteligente con los siguientes componentes:

- Sistema de inteligencia artificial en Raspberry Pi
- Sistema de control de movimientos en microcontrolador arduino
- Sistema de visión artificial
- Cámaras y sensores
- Pantalla digital y altavoces

Este juguete funcionará de la siguiente manera: cada vez que se presione el botón de encendido aparecerá un video de presentación, posee una GUI con varias opciones. La parte más importante es el de aprendizaje, se desbloquea con el rostro (Face ID). El robot estará dotado de sensores de distancia para evitar la colisión con otros objetos e incluso con el infante, precautelando la integridad física.

La cara es la contraseña, activa la cámara para detectar los objetos 2D cercanos. Extraída la información necesaria se tendrán las salidas del sistema, de forma audiovisual a través de una pantalla y altavoces.

El robot va a decir con voz la forma y color del cuerpo que tiene en sus manos al mismo tiempo va a mostrar un gráfico animado en una pantalla digital. De esta manera se pretende que los niños aprendan a identificar colores y figuras, además estimular el desarrollo sensorial, intelectual, de recreación y disciplina

Gracias a su potente algoritmo para reconocimiento de patrones, su fácil uso e interacción amigable, también servirá como elemento didáctico en la educación especial con mejores oportunidades en la enseñanza a niños con necesidades especiales, al mismo tiempo que puede ser útil como robot social terapéutico infantil y asistencial. Para mayor información consultar en el Anexo F.

OBJETIVOS

Objetivo general

- Implementar un juguete inteligente, mediante la utilización de Soft Computing, para la interacción de forma lúdica y cognitiva con niños de 2 a 7 años de edad, lo que desarrollará la adecuada percepción de formas y colores en el infante.

Objetivos específicos

- Analizar las diferentes técnicas de Soft Computing para el procesamiento inteligente de imágenes.
- Diseñar el sistema de visión artificial empleando técnicas de Soft Computing que realizará el tratamiento de imágenes de la escena captada durante la interacción con el infante.
- Diseñar e implementar el prototipo de juguete adecuado, mediante la utilización de materiales no tóxicos, garantizando la seguridad e integridad del pequeño.
- Integrar el software de visión artificial al prototipo construido, utilizando mecanismos electrónicos y tecnológicos adecuados, en la creación del juguete inteligente.
- Evaluar el funcionamiento y resultados del juguete inteligente en base al alcance y beneficios obtenidos con la implementación del mismo.

CAPÍTULO I

1. MARCO TEÓRICO REFERENCIAL

1.1 Juguetrónica

El termino juguetrónica hace referencia a un mecanismo psicopedagógico, donde un robot a más de ser un juguete se transforma en un recurso educativo (Tripero, 2017, pp.1-2). A partir de lo lúdico se desarrollan habilidades y competencias.

La juguetrónica tiene por objetivo crear contextos basados en principios de psicología cognitiva y del constructivismo, dejando en el pasado la enseñanza tradicional y enfocándose en nuevos métodos de enseñanza-aprendizaje. (Tripero, 2017, p.2)

Dentro del campo educativo es considerando un elemento no solo innovador sino enriquecedor, incrementa considerablemente el interés en los aprendices para adquirir mayor conocimiento. (Bravo & Forero, 2012; Ballesteros, 2011)

1.2 Inteligencia Artificial

De acuerdo con Nilsson (2001, pp.1-2), la IA (Inteligencia Artificial) es la ciencia que estudia el comportamiento inteligente en los ordenadores. Los cuatro pilares fundamentales, son: búsqueda de la mejor respuesta dentro de un conjunto de estados, algoritmos genéticos, redes neuronales artificiales y razonamiento análogo al pensamiento abstracto humano.

El concepto de IA está lleno de controversia, varios expertos niegan la posibilidad de que una máquina reemplace al ser humano, puesto su incapacidad para producir emociones y sentimientos, otros afirman que ese es un futuro cercano. (Diez, et al, 2001)

Gracias a la IA se han desarrollado sistemas expertos que imitan la capacidad mental, proporcionando resultados en corto tiempo y con mejores juicios. Una de las áreas de mayor aplicación es la medicina, ya que el acierto de diagnósticos presenta una alta tasa de confiabilidad, otras áreas beneficiadas son la: milicia, aeronáutica, ingeniería, economía, automotriz, biométrica, bioinformática, etc. (Nilsson, 2001, p.3)

1.3 Soft Computing

Zadeh introdujo el concepto de conjunto fuzzy en los años 60, apareciendo con ello nuevos métodos y modelos de programación para hallar una solución óptima a problemas correctamente planteados, precedentemente no resolubles con métodos convencionales. (Verdegay, 2005, pp.33)

El tema de Soft Computing era un tabú, no existía una definición clara hasta que L.A. Zadeh (1994) realizó la primera aproximación, englobando el uso de herramientas, metodologías y estrategias para solventar inconvenientes. Delimitando en los siguientes términos:

Soft computing no es un cuerpo homogéneo de conceptos y técnica. Más bien es una mezcla de distintos métodos que de una u otra forma cooperan desde sus fundamentos. En ese sentido, el principal objetivo es aprovechar la tolerancia que conlleva a la imprecisión e incertidumbre, para conseguir manejabilidad, robustez y soluciones a bajo costo. Los principales ingredientes de la soft computing son la Lógica Difusa, la Neuro-computación y el Razonamiento probabilístico, incluyendo en este último a los algoritmos genéticos, las redes de creencia, los sistemas caóticos y algunas partes de la teoría de aprendizaje. En esa asociación de la lógica difusa, neurocomputación y razonamiento probabilístico, la lógica difusa se ocupa principalmente de la imprecisión y el razonamiento aproximado; la neurocomputación del aprendizaje, y el razonamiento probabilístico de la incertidumbre y la propagación de creencias. (Zadeh, 1994, pp. 48-56)

Sin embargo, la definición anterior no dio los frutos deseados, razón por la cual Li redefinió así:

Cualquier proceso de computación que expresamente incluya imprecisión en los cálculos en uno o más niveles, y que permita cambiar la granularidad del problema o suavizar los objetivos de optimización en cualquier etapa, se define como perteneciente al campo de la Soft Computing. (Verdegay, 2005, pp. 45)

Con esta última definición, queda claro que esta rama de la IA fue concebida para manejar acontecimientos reales, tal y como lo tratan los humanos, empleando la inteligencia y raciocinio. Al resolver un problema incomprensible, aplicando una o más técnicas diferente a los métodos tradicionales como la visión computacional, se está aplicando soft computing.

Una de las concepciones de la Soft computing es satisfacer antes que optimizar, es decir, dar prioridad a una solución que satisfaga los requerimientos del usuario, sin importar si dicha respuesta no es la mejor dentro de un conjunto de soluciones. (Verdegay, 2005, pp.29-47)

En el **Figura 1-1** se observa la composición fundamental de la Soft computing, en el nivel de razonamiento aproximado están los métodos probabilísticos y lógica difusa. Por otro lado están los métodos de optimización, con los algoritmos evolutivos y redes neuronales. La fusión de varios métodos genera sistemas híbridos.

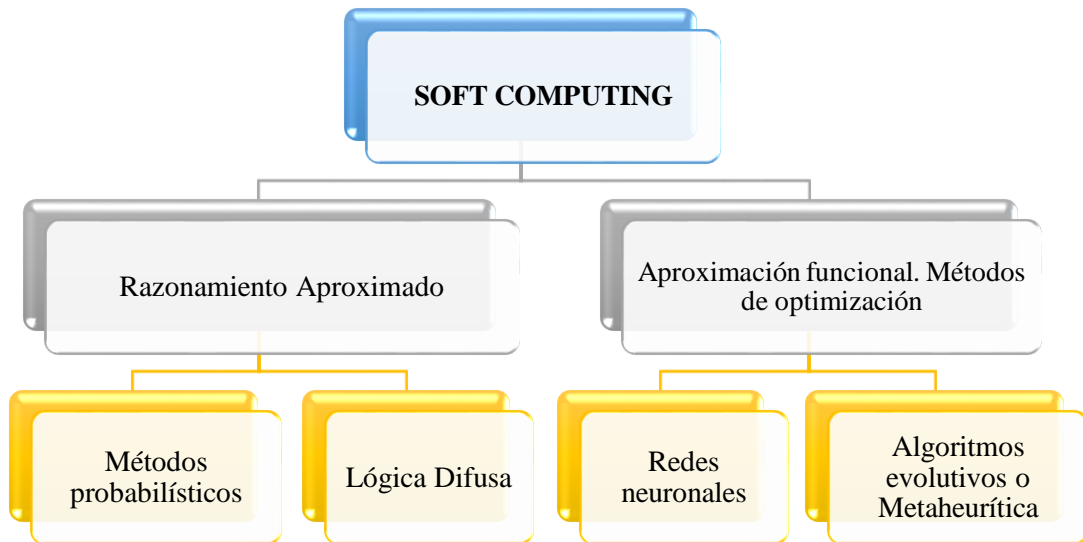


Figura 1-1: Componentes de la Soft Computing

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Un algoritmo MH (metaheurístico) encuentra soluciones a partir de datos imperfectos con menor esfuerzo computacional, es lo que le diferencia de los algoritmos de optimización, iterativos y hasta de modelo heurístico (González, 2008, pp. 30-32). La desventaja es el crecimiento exponencial, a medida que el problema crece. Existen cuatro tipos de MH que sobresalen:

- MH (1) evolutiva: selecciona la respuesta óptima, dentro de un conjunto de posibles soluciones, basado en la evolución natural.
- MH (2) de relajación: basada en los redondeos, delimita algún elemento característico.
- MH (3) de búsqueda de entornos: recoge espacios de soluciones iterativamente, parecido a los evolutivos pero emplean varios métodos de búsqueda.
- MH (4) de apoyo: es intermedia entre los anteriores, realiza valoraciones numéricas

1.3.1 Métodos que integran la Soft Computing

1.3.1.1 Modelo Probabilístico

El modelo probabilístico trabaja con base a criterios de probabilidad y estadística. Consiste en la observación indirecta de datos y la predicción, toma como referencia información pasada. Este modelo está acotado por reglas matemáticas y teoría de conjuntos, los modelos que destacan son las Redes Bayesianas y la cadena de Markov. (González, 2008, pp.33-34)

La red bayesiana es una representación gráfica, compuestas de grafos dirigidos (nodos, arcos) y una serie de probabilidades condicionales, considerando la teoría de utilidad. Cada nodo representa una variable aleatoria, unidos entre sí con arcos. Lo que permite realizar algoritmos computaciones de inferencia y aprendizaje. (González, 2008, p.34)

Este modelo estima variables desconocidas en base a variables conocidas, los arcos crean dependencias e independencias. Las relaciones de independencia probabilística simplifican la representación de conocimiento y razonamiento, como se puede ver en la **Figura 2-1**.

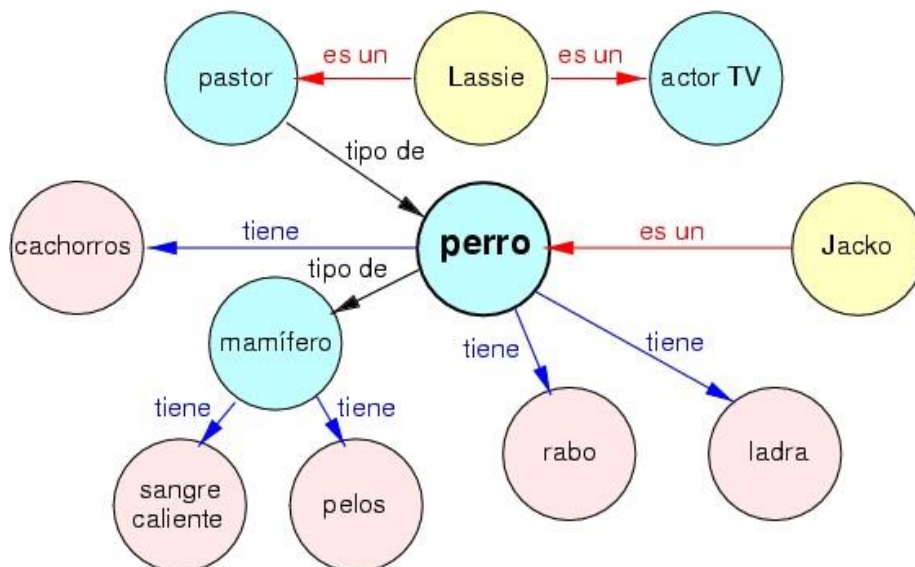


Figura 2-1: Ejemplo de red bayesiana

Fuente: <https://jpordonez.wordpress.com/2008/08/>

Las redes o cadenas de Markov son grafos no dirigidos, cada nodo representa un estado y están unidos por dependencias probabilísticas simétricas. Un modelo de Markov es un proceso estocástico, es decir, la probabilidad de que ocurra algún acontecimiento depende únicamente del evento inmediatamente anterior. (González, 2008, p.36)

Como se observa en la **Figura 3-1**, para predecir el estado 2 debería conocer primero el estado 1, diferente a como pasa en las redes bayesianas donde basta tener valores pasados sin importar la secuencia.

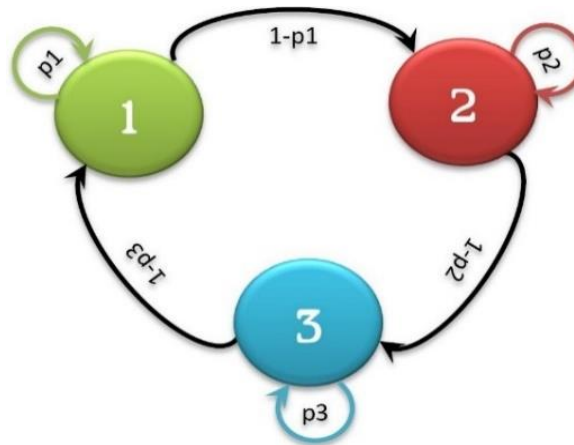


Figura 3-1: Ejemplo de cadena o red de Markov

Fuente: <http://bit.ly/2hPS4UD>

1.3.1.2 Lógica Difusa

La lógica fuzzy, borrosa o difusa es una aproximación al razonamiento introducida por Zadeh. Esta técnica permite trabajar con información parcial, poco entendible, inconclusa, con alto grado de imprecisión. (D'Negri & De Vito, 2006, pp. 126-136)

A diferencia de la lógica clásica en la que solo son admisibles dos opciones, la lógica difusa tolera valores dentro del intervalo $\{0, 1\}$, es decir, existen más de dos alternativas (Verdegay, 2005, p.2). Por ejemplo en el caso de las lavadoras, la ropa puede estar: sucia, no muy sucia, media limpia y limpia; los estados del lavado de ropa denotan el grado de pertenencia ($M_1, M_2, M_3, \dots, M_n$).

Para resolver problemas con lógica difusa primero se fuzzifica estableciendo grados de pertenencia (como en la **Figura 4-1**). Luego se desfuzzifica, sumando los productos de la fuzzificación por el singlenton correspondiente y dividido para la suma de las fuzzificaciones. La expresión matemática es la siguiente.

$$Y_2 = \frac{M_1 * C_1 + M_2 * C_2}{M_1 + M_2} \quad \text{Ecuación 1.}$$

Las dos valiosas técnicas de la lógica borrosa son: Mandami que no requiere de modelos matemáticos, como se aprecia en la **Figura 4-1**. Y Takagi Sugeno Kang que emplea herramientas matemáticas e interpreta operadores lógicos como: AND, OR, NOT. (D'Negri & De Vito, 2006, p.127)

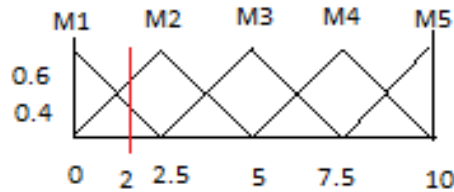


Figura 4-1: Fuzzificación

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Mandami fue el primero en crear un sistema de control para una máquina de vapor empleando lógica difusa. Años más tarde en Japón, Michio Sugeno con su grupo de trabajo diseñaron el metro de Sendai, usando un sistema de control difuso. A partir de estos dos grandes acontecimientos la lógica difusa es aceptada a nivel mundial y se usa en estudios conjuntamente con redes neuronales y algoritmos genéticos. (D'Negri & De Vito, 2006, pp.127-129)

1.3.1.3 Redes Neuronales Artificiales

Una RNA (Red Neuronal Artificial) es un conjunto de neuronas que modelan las propiedades más relevantes de un sistema nervioso biológico, con la capacidad de aprender a través del entrenamiento, basado en la minimización del error. Cabe resaltar que cada neurona trabaja de forma independiente. (Nilsson, 2001; González, 2008)

Las RNAs se caracterizan por adquirir conocimiento con la experiencia, se adaptan fácilmente a los cambios, poseen tolerancia a fallos, establecen relaciones no lineales entre los datos y el procesamiento es paralelo. Un ejemplo práctico de una red neuronal es el reconocimiento facial a partir de una fotografía, aun estando girada identifica a la persona que tiene dicho rostro en cuestión de segundos. (Nilsson, 2001, pp.23-25)

Izaurieta y Saavedra (2000, pp.1-2) determinan en base a un estudio neurológico que el procesamiento de cada neurona es lento, pero la cantidad de estas compensa su eficiencia, es lo que nos diferencia de los animales. El cerebro humano tiene cerca de 10^{10} neuronas con aproximadamente 10^{13} conexiones y un alto índice de eficiencia energética.

Nilsson (2001) clasifica a las redes neuronales de acuerdo a la dirección de propagación: hacia adelante (feedforward) y hacia atrás (feedback). Mientras Juan González (2008) subclasifica según la topología, de la siguiente manera:

- a) Por el número de capas
 - Redes neuronales monocapas: únicamente tienen la capa de entrada y de salida.
 - Redes neuronales multicapa: disponen de capas intermedias denominadas capas ocultas.
- b) Por el tipo de conexión
 - Redes neuronales no recurrentes: la propagación de las señales se produce en un solo sentido.
 - Redes neuronales recurrentes: la propagación de las señales son retroalimentadas.
- c) Por el grado de conexión
 - Redes neuronales totalmente conectadas: las neuronas de diferentes capas se encuentran totalmente conectadas.
 - Redes neuronales parcialmente conectadas: las neuronas de diferentes capas no están totalmente conectadas.

En la **Figura 5-1** se observa la estructura de una red neuronal multicapa no recurrente y totalmente conectadas, donde la neurona de interés es la salida, las neuronas de la capa de entrada contienen valores numéricos de alguna señal.

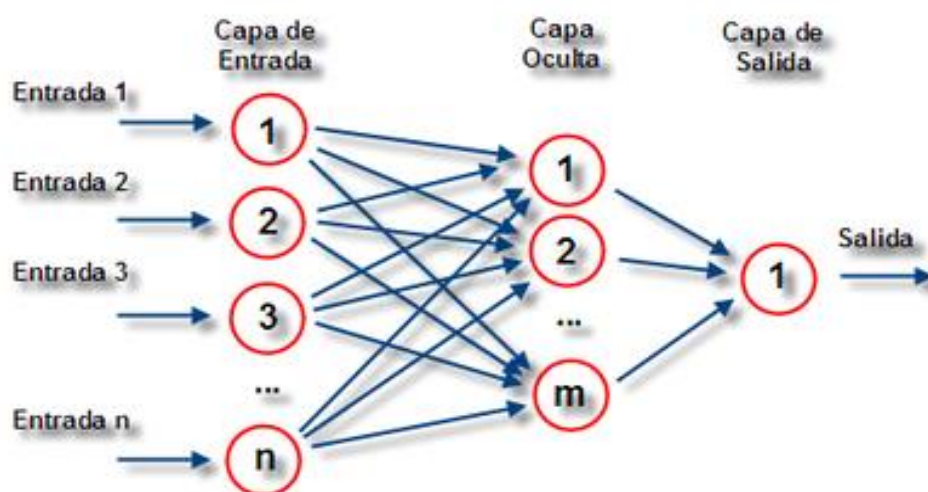


Figura 5-1: Estructura de una Red Neuronal Artificial

Fuente: <http://juanitajuanabernal.blogspot.com/p/propiedades.html>

Una RNA está compuesta por cuatro elementos básicos; entre ellos los pesos o sinapsis que determinan el comportamiento de la neurona, un sumador de sinapsis, una función de activación no lineal que limita la amplitud y el umbral exterior para activar la neurona. (Serrano, et al, 2009, p.15)

La **Figura 6-1** ilustra el modelo estructural de una red neuronal. Donde U_n resulta la sumatoria del producto de las entradas y los pesos (w_n), mientras que la salida corresponde a la diferencia de U_n con el umbral y todo esto multiplicado por la función no lineal (ρ).

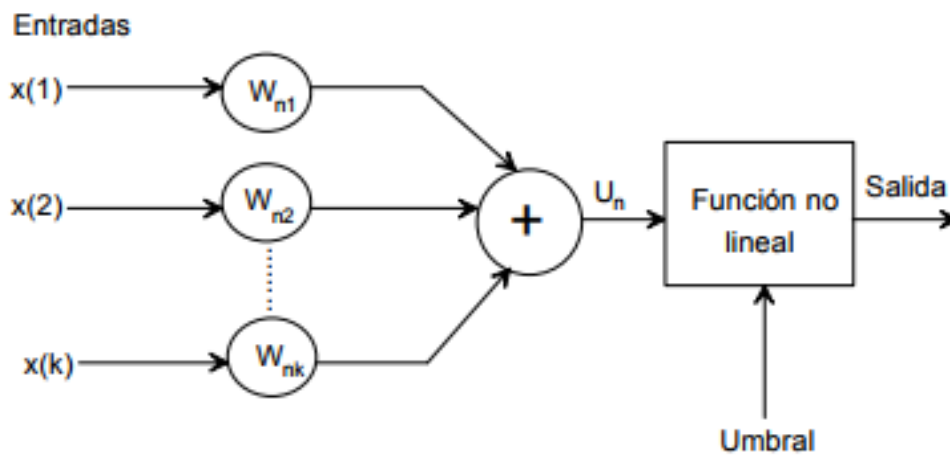


Figura 6-1: Esquema de un modelo neuronal

Fuente: Serrano, Soria, et al., 2009 (Redes Neuronales artificiales)

La Ecuación 2 expresa la suma de los productos de las entradas por los pesos y la Ecuación 3 determina la salida de la red neuronal.

$$U_n = \sum_{i=1}^k W_{ni} * X(i) \quad \text{Ecuación 2.}$$

$$out = \rho(U_n - umbral) \quad \text{Ecuación 3.}$$

McCulloch y Pitts crearon una neurona en un dispositivo binario, demostrando que la red neuronal puede resolver cualquier problema computable (Soria, & Martín, 2009, pp.8-10). Algunos de los operadores lógicos sintetizados en las RNA son: AND, OR, XOR. Las **Figuras 7-1, 8-1 y 9-1** esquematizan cada uno de los operadores mencionados.

El aprendizaje de una RNA puede ser supervisado y no supervisado. Para el entrenamiento supervisado primero se introduce al tema, repite, evalúa y corrige las sinapsis para minimizar el error con algún algoritmo. El entrenamiento se realiza varias veces hasta que la red responda correctamente. (Izaurieta & Saavedra, 2000, p.7)

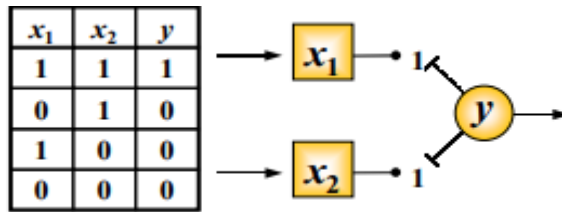


Figura 7-1: Función AND en una red neuronal

Fuente: Izaurieta & Saavedra, 2000 (Redes neuronales artificiales)

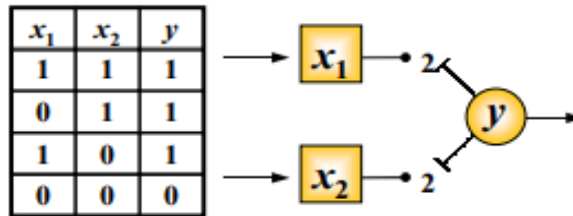


Figura 8-1: Función OR en una red neuronal

Fuente: Izaurieta & Saavedra, 2000 (Redes neuronales artificiales)

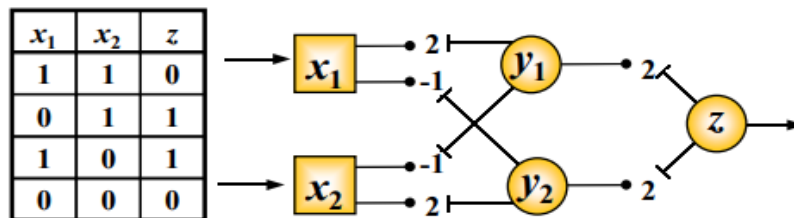


Figura 9-1: Función XOR en una red neuronal

Fuente: Izaurieta & Saavedra, 2000 (Redes neuronales artificiales)

En el entrenamiento no supervisado la red aprende por sí sola, empleando métodos de clasificación de entradas. El aprendizaje perceptrónico se aplica a redes monocapas para el reconocimiento de patrones, únicamente cuando los patrones resultan linealmente separables (Izaurieta & Saavedra, 2000, p.7). El funcionamiento de un perceptrón se muestra gráficamente en la **Figura 10-1**.

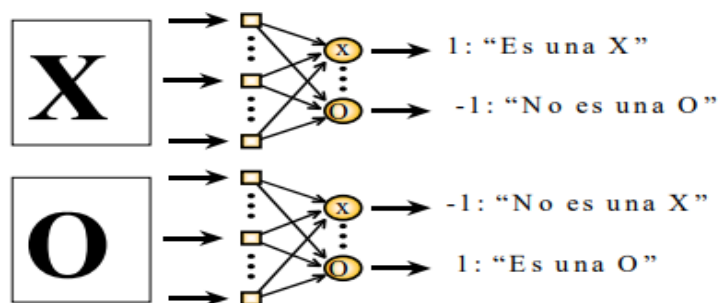


Figura 10-1: Esquema de aprendizaje perceptrónico

Fuente: Izaurieta & Saavedra, 2000 (Redes neuronales artificiales)

1.3.1.4 Sistemas evolutivos

Los sistemas evolutivos están inspirados en la naturaleza, es un conjunto de individuos unidos por operadores genéticos. Inicialmente el sistema biológico consiste en una pequeña población formada por funciones, constantes y entradas. A medida que evolucionan aprenden e incrementan la población, se detiene cuando se halle la mejor solución. (Pajares & et, 2003)

El algoritmo genético se fundamenta en la supervivencia del más fuerte. Escoge solo aquellos seres enérgicos para reproducir y formar nuevos organismos, usa operadores de cruce y mutación. Entretanto los cromosomas débiles son descartados. (Algoritmos genéticos, 2016)

La **Figura 11-1** muestra el esquema de un algoritmo genético. Las fases intermedias se repiten hasta que los organismos se adaptan completamente al medio y se elija al mejor individuo.

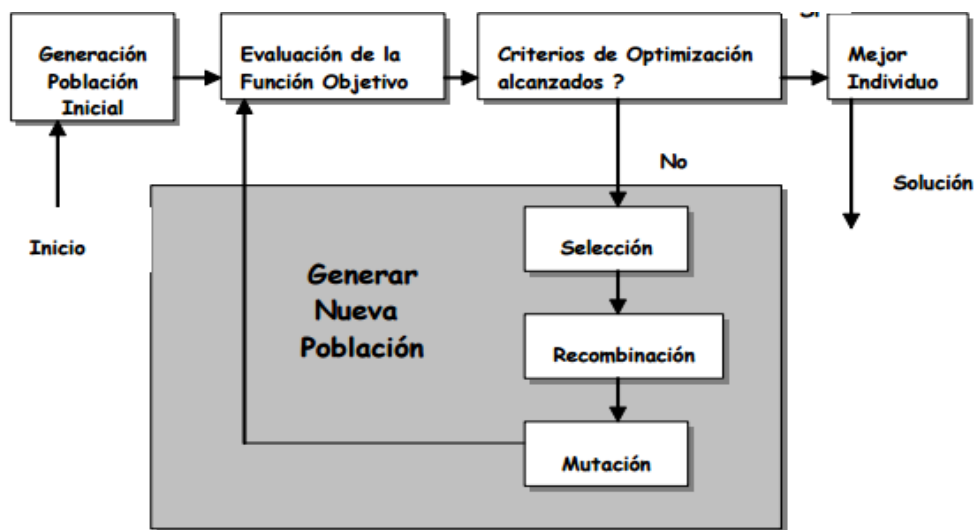


Figura 11-1: Esquema de Sistemas Evolutivos

Fuente: <http://www.it.uc3m.es/~jvillena/irc/practicas/estudios/aeag>

En el cruce se seleccionan dos padres aleatoriamente para producir dos cromosomas, con una probabilidad de 0.5 a 1; de no ser así simplemente se duplican los padres (Pajares & et, 2003). Existen cruces en un punto, dos puntos, cruce uniforme y aritmético. La **Figura 12-1** ilustra el cruce en dos puntos.



Figura 12-1: Operador cruce en dos puntos

Fuente: Algoritmos genéticos (<http://bit.ly/2xl9Pgt>)

El operador mutación se aplica a hijos de forma individual, selecciona genes al azar e invierte la posición. Se produce inversión, cambio o modificación de genes (Pajares & et, 2003). La **Figura 13-1** muestra un ejemplo de mutación con inversión de genes.

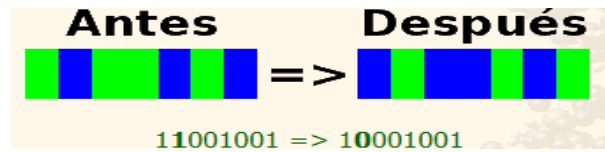


Figura 13-1: Operador mutación con intercambio de genes

Fuente: Algoritmos genéticos (<http://bit.ly/2xl9Pgt>)

Después de realizar el cruce y la mutación, se acepta o se rechaza a los nuevos hijos, para lo cual se aplica métodos de aceptación total, mejora o torneo. (Pajares & et, 2003)

1.4 Algoritmo AdaBoost

AdaBoost (Adaptative Boosting) es una adaptación de boosting creado en 1996 por Freund y Schapire. Ambos algoritmos entrenan una serie clasificadores débiles para convertirlos en fuertes, la diferencia radica en el entrenamiento iterativo de AdaBoost. (Morales, 2015, pp.21; Montejo-Ráez & Ur, 2006)

Boosting surge a partir de una historia de apuesta de caballos, donde un hacendado cansado de perder asigna valores iguales a todos sus amigos para que apuesten por él. A medida que pasaban las apuestas reasigna dichos valores, incrementaba a los que ganaban y reducía a los que perdían, asegurando su ganancia. (Paz, 2009, pp.4-5)

El algoritmo Adaboost consta de los siguientes pasos:

1. Definir las entradas, por lo general suelen ser un conjunto de datos con sus respectivas etiquetas para determinar el tipo.

$$(x_1, y_1), \dots, (x_n, y_n)$$

Donde:

n = número de datos

$x_i \in X$

$y_i \in Y = \{1, -1\}$, Asigna 1 cuando pertenece a la clase X y 0 cuando no pertenece.

2. Determina el número de iteraciones (pasos) e inicializa la distribución de pesos (iguales)

$$\text{Para } t = 1, \dots, T \quad \text{Ecuación 3.}$$

$$D_i = \frac{1}{n} \text{ para } i = 1, \dots, n \quad \text{Ecuación 4.}$$

3. Entrena el clasificador débil (*weak learner*) empleando la distribución anterior

4. Obtiene una hipótesis débil

$$h_t: X \rightarrow \{-1, +1\} \quad \text{Ecuación 5.}$$

5. Calcula el error más bajo

$$\epsilon_t = \Pr_{i \sim D_t}[h_t(x_i) \neq y_i] \quad \text{Ecuación 6.}$$

6. Calcula el peso del clasificador débil actual

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \quad \text{Ecuación 7.}$$

7. Actualiza la distribución D_t para cada uno de los datos, según los errores

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} X = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \quad \text{Ecuación 8.}$$

Donde:

Z_t es un factor de normalización

$$X = \begin{cases} e^{-\alpha_t} & \text{si } h_t = y_i \\ e^{\alpha_t} & \text{si } h_t \neq y_i \end{cases} \quad \text{Ecuación 9.}$$

8. Define la hipótesis final (clasificador fuerte), a partir de la suma de todas las hipótesis débiles.

$$H(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad \text{Ecuación 10.}$$

El algoritmo es sensible a datos ruidosos y atípicos, con mejor tiempo de ejecución. Se considera una herramienta poderosa para las redes neuronales. Un buen entrenador contiene una cantidad considerable de datos, un bajo error de entrenamiento y es fácil de implementar. (Morales Sánchez, 2015, p.8)

Adaboost minusvalora el error mas no alcanza tasas de detección a costa de falsos positivos, de esa tarea está encargado el umbral perceptrónico. Un umbral alto no garantiza una buena detección, mientras que un umbral bajo promete una alta tasa de detección. (Paz, 2009, pp.27)

La **Figura 14-1** muestra el diagrama de flujo del algoritmo AdaBoost.

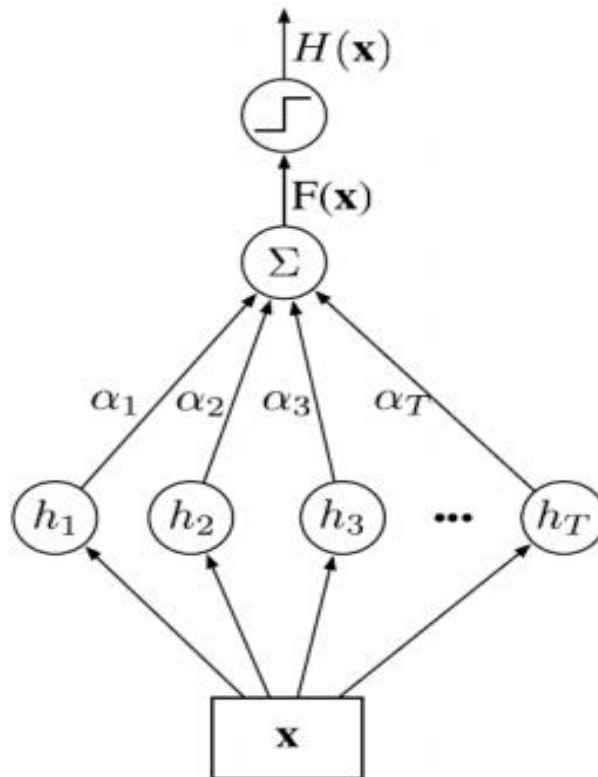


Figura 14-1: Diagrama de flujo del algoritmo AdaBoost

Fuente: Morales, 2015 (Diagrama de Flujo funcionamiento AdaBoost)

1.4.1 Árboles de decisiones para la clasificación

Los árboles de decisiones tipo CART (Classification and Regression Trees) son de gran subvención para el algoritmo Adaboost, su estructura permite buscar resultados en base a una secuencia de reglas, similar a la condición “if”. (Morales, 2015, pp.30)

La **Figura 15-1** muestra un árbol de decisiones, cada vez que se ingrese un dato a la raíz su atributo es evaluado y se define el camino a seguir ya sea verdadero o falso. Repitiendo secuencialmente este proceso hasta llegar a una hoja en la que se pueda predecir el valor de Y (-1, +1).

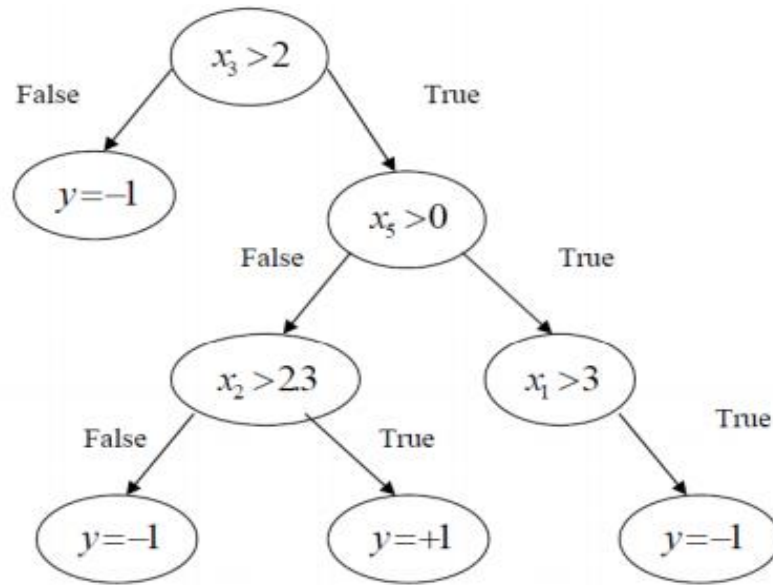


Figura 15-1: Estructura del árbol de decisiones tipo CART

Fuente: Morales Sánchez, 2015 (Ejemplo del árbol CART)

1.5 Reconocimiento facial con el método de Viola-Jones

El reconocimiento facial es una actividad innata del ser humano, hasta un recién nacido puede recordar el rostro de su madre sin mayor esfuerzo. Esta tarea la ejecuta una parte específica del cerebro, compuesta por células nerviosas especializadas. Pese a ser una acción completamente sencilla para el hombre, resulta sumamente complejo implementar en un computador. (Moreno Díaz, 2004, p.1)

En el artículo “*Robust real-time face detection*” Paul Viola y Michael Jones (2004) proponen un método efectivo para la detección facial, basados en el aprendizaje automático a partir de clasificadores sencillos que entrenan varias imágenes positivas y negativas. Incluyen un algoritmo adaboost y clasificadores cascada para el entrenamiento, además usan filtros haar para establecer algún fragmento de cara.

Para el reconocimiento se requiere de una imagen o secuencia de videos, la misma que se convierte a formato digital en escala de grises, para extraer los rasgos faciales y relacionarla con una base de datos previamente establecida. Determinando si existe alguna cara y a quien le pertenece. (Viola & Jones, 2003, p.1)

1.5.1 Características faciales

La percepción fisonómica es el resultado de un estudio completo de los atributos locales, como nariz, boca, mentón, borde la cara, mejillas, ojos, barba y cabello, que caracterizan el rostro. Moreno (2004, pp. 2-3) hace referencia a dos procedimientos tradicionales para la extracción de características faciales por medio de un computador: holísticos y geométricos.

La primera forma emplea modelos de comparación de subespacios faciales, como el Análisis de Componentes Principales (PCA) a partir de facciones significativas o eigenfaces y el Análisis Lineal Discriminante (LDA) con fisherfaces. Entretanto la segunda técnica compara las peculiaridades geométricas del semblante. (Moreno, 2004, pp. 15-30)

Eigenfaces es un método estadístico, incorpora una red bayesiana para la clasificación. Cada que toma una imagen la transforma en un vector unitario, halla la media y proyecta el vector resultante en un subespacio de menor dimensión. Dicha proyección se relaciona con la proyección de la base de datos para determinar alguna similitud. (OpenCV, 2017; Moreno, 2004)

1.5.2 Clasificadores cascada Haar

Los clasificadores segmentan la imagen y los filtros haar determinan si un fragmento corresponde a una cara. Si se consideraran todas las posibles posiciones en una subventana de 24x24 pixeles, se tendría alrededor de 45396 características, lo cual significa un alto costo computacional. Como solución es este problema se hace uso de la imagen integral. (Planells, 2009; Rama & Tarrés, 2007)

La Ecuación 11 calcula el número de características en una ventana.

$$\text{número características} = \sum_{1 \leq i \leq N} w_i \cdot \text{Sum_rectángulo}(r_i) \quad \text{Ecuación 11.}$$

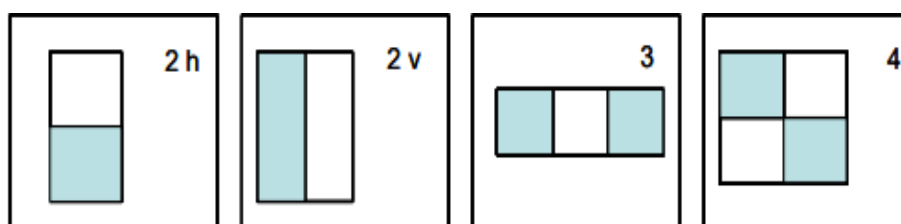


Figura 16-1: Características Haar

Fuente: Adaboost con aplicación a la detección de cara, 2009

Viola y Jones (2004, p.139) limitaron las operaciones sobre la imagen, permitiendo solo 3 tipos de características: de borde (2 rectángulos), de línea (3 rectángulos) y en forma de X (4 rectángulos), tal y como se presenta en la **Figura 16-1**. Al igual que las características, los pesos pueden tomar un solo valor ya sea +1 ó -1.

1.5.2.1 Imagen Integral

La imagen integral extrae las características de forma rápida en base a la propia imagen acumulativa, sin considerar la intensidad. (Guevara, et al., 2008). Una imagen integral se observa en la **Figura 17-1**.



Figura 17-1: Imagen integral

Fuente: Detección de rostros con clasificadores cascada, 2008

El punto (x, y) de la **Figura 17-1** contiene la suma de todos los pixeles de la parte superior izquierda, donde $ii(x, y)$ es la imagen integral en el punto (x, y) empezando por el origen y la imagen original es $i(x, y)$ (Viola & Jones, 2004, p.139). La Ecuación 12 es la expresión matemática que se emplea para hallar la imagen integral y las Ecuaciones 13 y 14 corresponden a la recursividad.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad \text{Ecuación 12.}$$

$$s(x, y) = s(x, y - 1) + i(x, y) \quad \text{Ecuación 13.}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad \text{Ecuación 14.}$$

Con las expresiones anteriores se calcula la suma de pixeles de un rectángulo cualquiera, con 4 accesos de memoria. Por ejemplo para calcular el valor del rectángulo A de la **Figura 18-1**, se suma A con D y se resta C y B.

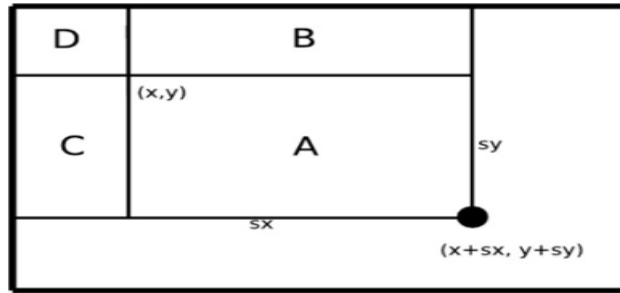


Figura 18-1: Convolución Filtro Haar con imagen integral

Fuente: Planells, 2009 (Implementación de algoritmo Viola-Jones)

1.5.3 Entrenamiento

Para construir un clasificador fuerte (reconocimiento facial) se entrena cada clasificador cascada con el algoritmo Adaboost, seleccionando un subconjunto de funciones. Inicialmente se le asigna pesos iguales a todas las imágenes, estos valores se modifican en cada iteración del entrenamiento volviendo a calcular la tasa de error, se repite el procedimiento anterior hasta alcanzar la tasa de error mínima. (Viola & Jones, 2004, p.144)

Las tres medidas fundamentales del reconocimiento facial son las siguientes:

- Tasa de detección: porcentaje de caras correctamente detectadas
- Tasa de falsos positivos: regiones marcadas incorrectamente
- Tasa de falsos negativos: número de caras detectadas correctamente

La **figura 19-1** muestra un ejemplo de clasificador cascada. De la imagen ingresada se evalúa cada etapa, si en C1 falla se rechaza y no se consideran las características restantes, caso contrario se analiza la etapa C2 de características y continúa el proceso. La ventana que supere todas las etapas es aquella que contiene una cara.

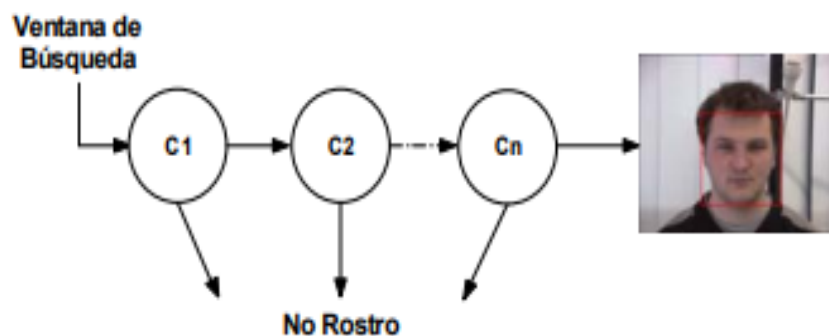


Figura 19-1: Clasificador cascada tipo haar

Fuente: Guevara, et al, 2008 (Detección de rostros con clasificades cascada)

Para el entrenamiento se acepta únicamente zonas con alta probabilidad de contener caras y se desechan aquellos pixeles con información vaga que no aportan a la detección, disminuyendo tiempo y recursos. (Planells, 2009, p.7)

1.6 Visión Artificial

VA (Visión Artificial) es un proceso computacional, dotado de elementos capaces de ver el entorno que le rodea. Dichos sistemas reconocen colores, interpretan fotografías, localizan objetos, identifican figuras, controlan otros sistemas; todo esto mediante el procesamiento de imágenes. (Nilsson, 2001, pp.2-5)

La **Figura 20-1** muestra un esquema de VA. Primero se captura la imagen, se le da el tratamiento necesario aplicando visión y se extrae sus características, tales como tamaño, forma, color, textura, composición, etc.



Figura 20-1: Esquema general de VA

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Un sistema de VA está formado por un conjunto de etapas en forma secuencial, estas etapas son:

- a. **Sensorial:** consiste en la captura de la imagen en una escena
- b. **Pre procesamiento:** en esta etapa se realiza la transformación y filtrado de la imagen
- c. **Segmentación:** es el aislamiento de los objetos de la imagen
- d. **Parametrización:** se extraen las características de la imagen
- e. **Clasificación:** consiste prácticamente en el reconocimiento
- f. **Resultados:** interpreta la escena

1.6.1 Adquisición de la imagen

A través de sensores o cámaras se extrae la imagen de una escena en forma de matriz. Un sistema básico de adquisición de imágenes contiene un lente y un detector, el detector es un sensor de imagen conocido como CCD (García, 2008). El juguete inteligente usará una cámara para adquirir las imágenes del medio exterior.

1.6.2 Digitalización de imágenes

Las imágenes se digitalizan en una matriz de dimensiones $n \times m$, un computador procesa esta matriz para tener una representación icónica del entorno. La imagen digital no es más que una función de intensidad bidimensional, tiene dos propiedades; la resolución, ligada a la cantidad de píxeles y la definición, corresponde a la nitidez. (Pajares & et, 2003)

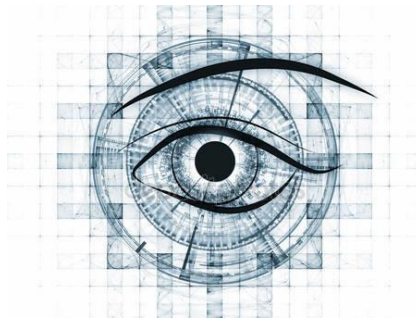


Figura 21-1: Imagen Digital

Fuente: <http://bit.ly/2yJKQEY>

Para representar los espacios de una imagen digital se emplea código de colores. Los sistemas sobresalientes son el modelo RGB y HSV, donde el pixel indica el nivel de intensidad. Cabe resaltar que el color es un potente descriptor, junto con el brillo de la matriz y la saturación. (Pajares & et, 2003)

1.6.2.1 Modelo RGB

RGB (Rojo, verde y azul) es la composición de colores primarios. En las imágenes a blanco y negro, el rango de colores varía de 0 a 255. El 0 representa el negro absoluto, el 255 corresponde al blanco absoluto, existen 254 tonalidades más distintas en escala de grises. (Castañeda & Cayambe, 2016, p.8)



Figura 22-1: Combinaciones del modelo RGB

Fuente: <http://www.davidmiguel.com/arduino/dia-2-led-rgb/>

Para las imágenes a color se asigna 3 bandas espectrales. Cada combinación representa un color: negro (0,0,0); blanco (255,255,255); verde (0,255,0); rojo (255,0,0) y azul (0,0,255). En la **Figura 22-1** se ilustra las combinaciones del modelo RGB.

1.6.2.2 Modelo HSV

El espacio de color HSV se basa en tres propiedades, matriz o tonalidad, saturación y brillo o luminosidad. Este modelo mantiene cierta similitud respecto al espectro de colores visibles, a diferencia del modelo RGB, como se evidencia en la **Figura 23-1**.

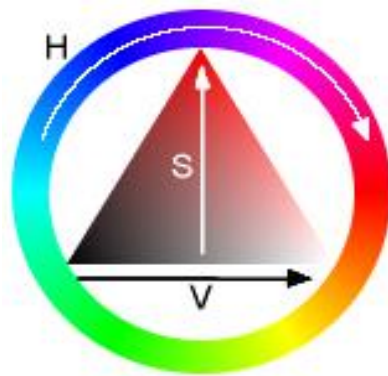


Figura 23-1: Representación del modelo HSV

Fuente: Los colores de la informática, 2011

Cada ángulo de la matriz representa un color, que va desde 0° hasta 360°. La saturación de color viene dada en porcentaje de 0% hasta el 100% (0-1), en caso de ser cero se tiene una tonalidad blanca. El brillo o luminosidad se mide en porcentajes, siendo el 0% negro y 100% el blanco. (Pajares & et, 2003)

1.6.3 Procesamiento de imágenes

Con el procesamiento se mejora la calidad de la imagen original, gracias al tratamiento digital como en la **Figura 24-1**. Una imagen es un conjunto de punto dentro de un plano (x, y), cada punto representa un pixel. Otras acciones del procesamiento de imágenes son la detección de bordes y la búsqueda de regiones. (García, 2008)

Previo al procesamiento de imágenes, se realiza el pre procesado, para eliminar el ruido. En el pre proceso intervienen algunas operaciones, tales como: suma, resta, multiplicación, división, inversión, umbralización, binarización, logaritmos, exponenciales y aplicación de operadores lógicos. (García, 2008)

La **Figura 24-1** (a) muestra la imagen original tomada con una cámara y la figura (b) muestra la imagen clara, aplicando ecualización en escalada de grises.



Figura 24-1: Aumento de contraste: (a) imagen captada, (b) ecualización en escala de grises

Fuente: Procesamiento de imágenes y visión (Sucar & Gómez)

1.6.3.1 Transformaciones morfológicas

Las transformaciones morfológicas son operaciones que se aplica a imágenes binarias, para modificar su geometría manteniendo su esencia. El objetivo de estas transformaciones es reconstruir imágenes nítidas libre de perturbaciones, facilitando así el análisis y extracción de características. Las operaciones básicas son:

- Dilatación: Aumenta el contorno de objetos. Como se ilustra en la **Figura 25-1**.
- Erosión: Reduce el contorno de objetos, esta operación no es inversa a la dilatación. Como se ilustra en la **Figura 26-1**.
- Apertura: Realiza primero erosión y luego dilatación.
- Cierre: Realiza primero la dilatación y luego la erosión, es una operación inversa a la apertura.



Figura 25-1: Dilatación de una imagen binaria

Fuente: Visión artificial y procesamiento digital de imágenes usando Matlab, 2008



Figura 26-1: Erosión de una imagen binaria

Fuente: Visión artificial y procesamiento digital de imágenes usando Matlab, 2008

1.6.3.2 Transformaciones geométricas

Las transformaciones geométricas modifican las relaciones espaciales entre píxeles, ya sea de forma individual o en vecindad. Las transformaciones individuales alteran píxel a píxel en una escala global, emplean operadores de identidad, inverso y de umbral. Mientras que las transformaciones por vecindad realizan operaciones de filtro pasa alto y pasa bajo sobre un conjunto de puntos o píxeles. (Pajares & et, 2003, pp. 40-50)

1.6.3.3 Operaciones sobre histograma de una imagen

El histograma es una representación gráfica de la distribución de píxeles en una imagen. La forma de representación es en barras verticales, como en la **Figura 27-1**.

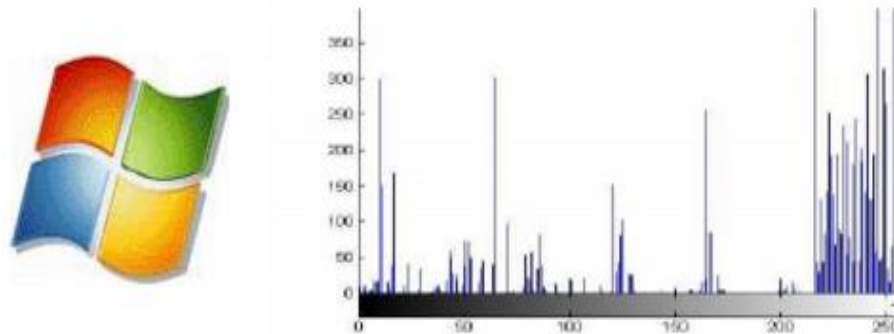


Figura 27-1: Imagen con histograma

Fuente: Visión artificial y procesamiento digital de imágenes usando Matlab, 2008

Las operaciones en los histogramas son:

- Aumento y reducción de contraste
- Ecuilización de histogramas
- Ajuste de intensidad

1.6.4 Detección de contornos

La detección de contornos es un paso para el reconocimiento de patrones, se caracteriza por el cambio brusco en la intensidad lumínica que presentan los bordes. Asumiendo que cada patrón pertenece a una categorización. (García, 2008)

Para detectar el contorno de una imagen se aplican filtros pasa alto especiales, para reducir el ruido y seleccionar solo los píxeles con mayor intensidad, dando realce al borde y facilitar su detección.

En la **Figura 28-1** se visualiza un ejemplo de extracción de contorno en monedas.

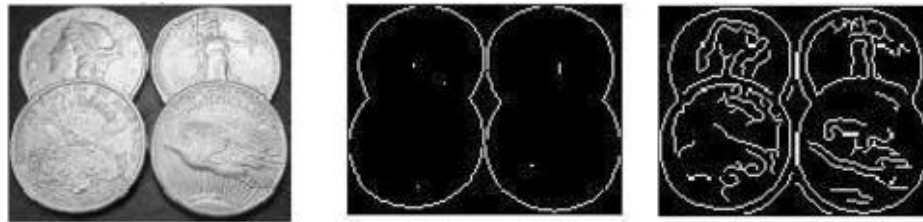


Figura 28-1: Detección de contorno de una imagen

Fuente: Visión artificial y procesamiento digital de imágenes usando Matlab, 2008

1.7 Fundamentos de robótica

Un robot es una máquina reprogramable de uso general, cuenta con sensores externos encargados de examinar el entorno que les rodea, además poseen un procesador de algoritmos relacionados a su sistema de control y sensorial. (K. Fu, 1989, p.1)

El término robótica aparece por primera vez en una obra de teatro en 1921, donde los protagonistas eran máquinas en forma de humanos. Los robots surgieron a partir de la Segunda Guerra Mundial con Alan Turing (K. Fu, 1989, pp.1-2). Sin duda las películas de ciencia ficción han dado vida a varios autómatas y motivado para el diseño de nuevos robots.

En el **Figura 29-1** se indica las principales partes que constituyen un robot.

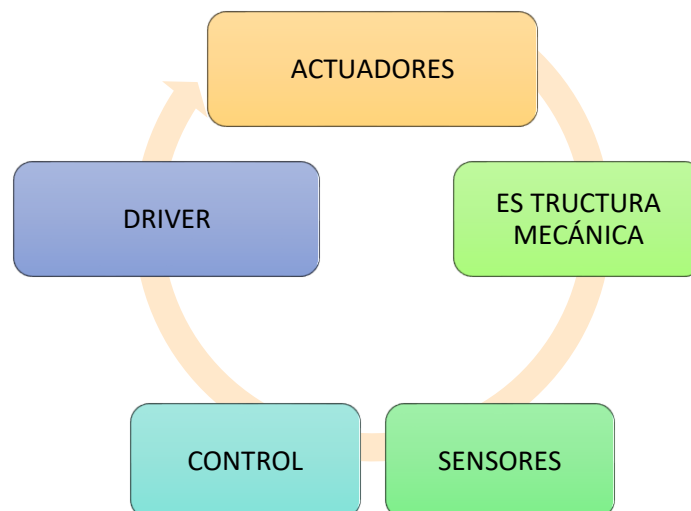


Figura 29-1: Componentes de un robot

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Uno de los tipos de robot más usados son los móviles, caracterizados por tener patas, ruedas, llantas o cualquier sistema de locomoción que le permitan movilizarse libremente sobre una superficie.

1.7.1 Inteligencia del Robot

La inteligencia en un robot comprende la planificación y control de acciones, mediante un algoritmo. La capacidad para ejecutarlo eficientemente y en el menor tiempo posible, depende del hardware (unidad de procesamiento y control) y del software (sistema experto, técnicas aplicables de IA) que se utilice. (K. Fu, 1989)

1.7.2 Sistema sensorial

Los sensores hacen las veces de los sentidos, existen sensores internos y externos. Los externos permiten relacionarse con su entorno de manera flexible, tal que pueda operar los programas de enseñanza, detecta variables como distancia, contacto, luminosidad, alcance. Los internos tratan con variables como la posición, un claro ejemplo es el sensor giroscópico. (K. Fu, 1989)

1.7.3 Actuadores

El sistema de accionamiento es un conjunto de dispositivos que generan movimiento de acuerdo a las órdenes emitidas por el sistema de control. Los actuadores para la locomoción de los robots son motores eléctricos, debido a sus ventajas de control, sencillez, precisión, alta fiabilidad y bajo consumo de potencia. (Rentería & Rivas, 2000)

1.7.3.1 Motorreductores

Son motores eléctricos que transforman energía eléctrica en mecánica, cotizados para robótica, debido a la adecuada reducción de velocidad realizando movimientos seguros y eficientes. Un reductor de velocidad brinda una serie de beneficios, como regularidad perfecta tanto en velocidad como en potencia transmitida, bajo peso y tamaño reducido. (Alzate & Lotero, 2008, pp.82-84)

La **Figura 30-1** presenta el principio de funcionamiento de los motores reductores con dos esferas de diferente diámetro. Al ejercer una fuerza en la rueda A producirá un pequeño movimiento en la rueda B, es decir, la rueda B completará una vuelta cuando la esfera de menor dimensión haya dado varias vueltas.

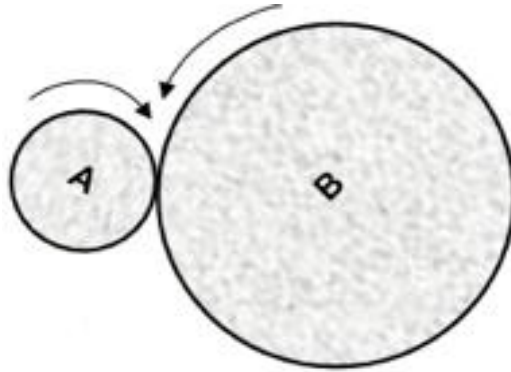


Figura 30-1: Principio de funcionamiento de motorreductores

Fuente: Principio de funcionamiento de motorreductores (Alzate & Lotero, 2008)

Con este sistema no solo disminuye la velocidad de giro del motor, si no también aumenta el denominado par o torque. La relación inversa que tiene la potencia con la velocidad final del giro equivale al torque del mismo, cuando más alto sea la velocidad de giro menor será el torque, este tipo de motores resulta conveniente usar cuando predomine la fuerza ante la velocidad. (Alzate & Lotero, 2008, pp.82-84)

1.7.4 Estructura del robot

La estructura es el bastidor del robot, compuesta por piezas resistentes que dan soporte y protegen a los elementos internos, tales como, cámaras, sensores, motores, baterías, llantas. De esta forma se garantiza la seguridad del usuario y del propio robot. A más de proporcionar apariencia o personalidad. (Castañeda & Cayambe, 2016)

Para realizar el boceto del robot se emplea programas especializados en el modelamiento de planos estructurales. Generalmente los prototipos juguetrónicos usan materiales con textura resistente, liviana, aislante, sobretodo libre de sustancias químicas tóxicas que pongan en riesgo la salud del niño.

1.7.5 Fuente de Alimentación

Todo artefacto requiere electricidad para funcionar, los componentes electrónicos necesitan ser excitados por una intensidad de corriente y tensión para activarse. En los robots se seleccionan aquellos entes que proveen energía garantizando la autonomía, más aún en el caso de los robots móviles que no pueden desplazarse con libertad conectados a un toma corrientes u otro medio de alimentación estático. (Rentería & Rivas, 2000)

1.7.5.1 Baterías

Una batería es un dispositivo capaz de almacenar energía eléctrica por medio de productos químicos, existen recargables y desechables. A la hora de seleccionar una batería hay que considerar el uso y requerimientos, para el caso de un robot móvil es preferible una pequeña, liviana, con alta tasa de descarga, que provea la cantidad necesaria de corriente y voltaje. (Castañeda & Cayambe, 2016)

1.8 Minicomputador Raspberry Pi

Raspberry Pi es una mini computadora, reducida a una sola placa diminuta lo suficientemente potente. La idea de esta placa fue concebida por un grupo de aprendices de la Universidad de Cambridge, con el objetivo de enseñar programación a niños. Este dispositivo es compatible con diferentes SO, pese a tener su propia plataforma de código abierto Raspbian (González, et al, 2014)

Raspberry pi usa software libre, es de bajo costo y fácil de usar; todas estas ventajas han hecho de Raspberry el sistema embebido más cotizado en el mercado. La placa funciona con una fuente de alimentación de 2.5A a 5V en DC.

Para el desarrollo del presente trabajo se utilizará la placa Raspberry Pi 3 modelo B, una de las más modernas, cuyas características se detallan a continuación:

- Procesador Quad Core a una velocidad de 1.2 GHz
- Memoria RAM de 1 GB
- 4 puertos USB
- 40 pines GPIO
- Puerto full HMI y Ethernet
- Conectividad Bluetooth 4.1 y Wi-Fi 802.11n
- VideoCore IV núcleo de gráficos 3D
- Tarjeta Micro SD de 32 Bit para cargar el SO
- Interfaz de cámara (CSI) y pantalla (DSI)

En la **Figura 31-1** la composición de la placa Raspberry Pi 3 B+.

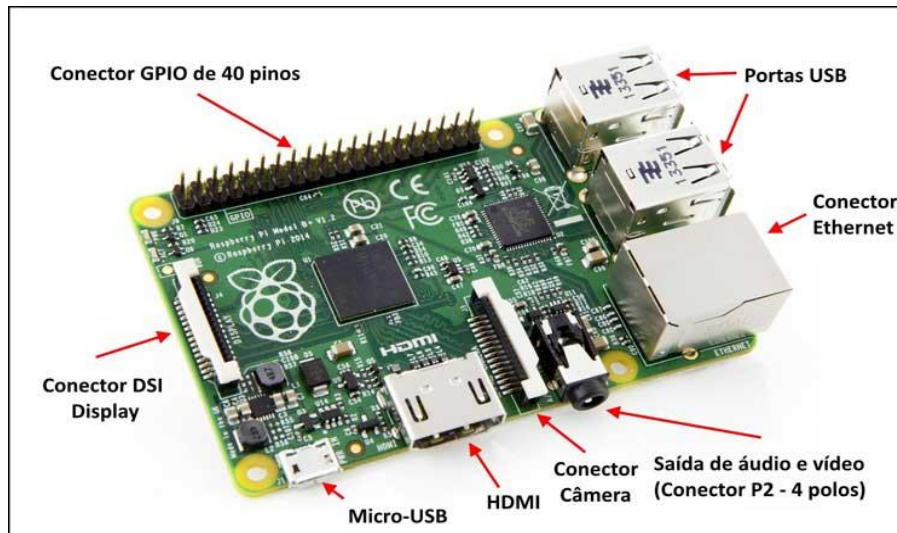


Figura 31-1: Minicomputador Raspberry Pi 3 Modelo B

Fuente: <http://mikroe.es/category/electronica/raspberry-pi/>

1.9 Arduino

Arduino es una plataforma electrónica de código abierto, nació de la idea de crear una placa de circuito impreso de bajo costo. Años más tarde se sumaron a este proyecto varios aficionados, programadores, estudiantes y catedráticos con el fin de mejorar las utilidades de la placa, para adaptarse a las nuevas necesidades y desafíos. (Arduino-Home, 2017)

Posee un microcontrolador AVR, puertos digitales y analógicos de E/S, algunos pines son exclusivos para salidas PWM, el número de puertos depende de la versión de Arduino. Esta placa es compatible con todo los SO, se le programa desde el computador en el software propio y se carga el programa en la placa usando comunicación serial. (Arduino-Home, 2017)

La **Figura 32-1** muestra la placa Arduino Mega.



Figura 32-1: Placa Arduino Mega

Fuente: <https://www.arduino.cc/en/Main/arduinoBoardMega>

Arduino Mega es una placa con las mejores funcionalidades, diseñado para proyectos de mayor complejidad. Tiene un microcontrolador ATmega2560, cuenta con 54 pines digitales (39 de E/S, 15 salida PWM), 16 entradas analógicas y un oscilador de cristal de 16 MHz. Funciona con una tensión de 5 VDC. (Arduino-Home, 2017)

1.10 Software para desarrollo de programas y modelamiento del prototipo

1.10.1 Sistema Operativo Raspbian Jessie

Raspbian es el SO propio de la plataforma Raspberry Pi, es una distribución de Linux, por lo tanto, es de código abierto, basado principalmente en Debian. Para arrancar el Raspberry se necesita tener grabado el SO en la tarjeta micro SD en formato imagen. (Harrington, 2015)

Existen varias versiones de Raspbian como, Jessie, Lite y Wheezy. La más actual es Raspbian Jessie 4.4 lanzado en el 2017. En el SO también viene preinstalados programas educativos. (Harrington, 2015)

1.10.2 OpenCv

OpenCv acrónimo de Open Source Computer Vision (Visión computarizada abierta), es una biblioteca gratuita de visión artificial. Contiene aproximadamente 500 funciones para procesamiento de imágenes en tiempo real e incluye patentes de algoritmos implementados computacionalmente; para el reconocimiento de patrones, calibración de cámaras, visión estero y robótica. (Arévalo, et al, 2002; OpenCV, 2017)

Actualmente OpenCv (2017) cuenta con una interfaz de programación en lenguaje C, Python, Java y Android, el programador dispone de una estructura bastante sencilla y eficiente, resulta útil para problemas de aprendizaje automático. OpenCv tiene una estructura modular, integrada principalmente por:

- **core:** módulo que contiene funciones básicas para procesamiento de imágenes
- **highgui:** módulo para interfaz
- **imgproc:** módulo con algoritmos de procesamiento de imágenes
- **video:** módulo para análisis de videos
- **objdetect:** módulo para detección y reconocimiento de objetos

A partir de la versión 2.4 aparece la librería FaceRecognizer para facilitar el reconocimiento facial, los algoritmos disponibles son: eigenfaces, fisherfaces e histogramas de patrones binarios locales. Al estar bajo una licencia BSD es adoptada en todo el mundo y se puede solicitar ayuda a la comunidad de usuarios para aclarar inquietudes. (OpenCV, 2017)

1.10.3 Python

Python es un lenguaje de programación abierto, compatible con Raspbian y otros SO disponibles en el mercado. Este programa se caracteriza por tener un código sencillo, donde no es necesario declarar el tipo de variables que se va usar, ni preocuparse de los puntos y comas a final de cada línea. (González, et. al, 2014)

Es un lenguaje de programación orientado a objetos, más que comandos o sentencias es una escritura en inglés. Para ejecutar o correr el programa, se interpreta en vez de compilar, por tal razón se le conoce como lenguaje interpretado. Además, incorpora ciertos paquetes de ayuda para visión artificial. (González, et. al, 2014)

1.10.4 SolidWorks

SolidWork es una aplicación CAD (Diseño asistido por computadora), permite modelar piezas en 3D de manera precisa. Permite realizar análisis mecánico, simulación, ensamblaje de un conjunto de piezas. Es una herramienta potente para el diseño mecánico de productos mediante un ordenador. (Gómez Gonzáles, 2014, pp.7-50)

Al inicializar el programa, se presentan 3 opciones: pieza, ensamblaje y dibujo. En la primera opción permite diseñar partes individuales. En ensamblaje se unen las piezas diseñadas, integrando las partes para formar un todo sólido y rígido. En dibujo se realizan planos en 2D. (Castañeda & Cayambe, 2016)

Solidwork presenta una interfaz gráfica sencilla y guiada. Incorpora librerías o Toolbox que contienen tornillos, levas, rueda dentada, tuercas, sistema de engranes, entre otros sistemas mecánicos pre diseñados que sirven de ayuda al usuario. (Gómez Gonzáles, 2014, p.8)

1.10.5 Software Arduino

Arduino es un software libre, propio de las placas Arduino. Contiene herramientas para reconocimiento de dispositivos, además de una biblioteca C de alta calidad, para compilar los programas. Fácil de usar, incluye programas de ayuda o muestra que sirven de guía para los usuarios. Este software es compatible con todos los SO, tiene varias versiones, por tratarse de código libre se actualiza constantemente. (Arduino-Home, 2017)

CAPÍTULO II

2. MARCO METODOLÓGICO

En el presente capítulo se detalla la metodología aplicada en el diseño e implementación del juguete inteligente.

2.1 Instalación de programas en Raspberry Pi

Antes de ejecutar cualquier tarea en la Raspberry Pi se debe instalar el software base. Por tanto el primer paso es la instalación del SO, posterior a esto se cargan todos los paquetes y librerías necesarias. Es sustancial que todos los programas estén correctamente instalados y configurados, para evitar futuros errores.

2.1.1 Instalación del Sistema Operativo Raspbian Jessie

Para instalar el SO se guarda Raspbian Jessie como imagen ISO en la tarjeta Micro SD, la misma que se inserta en la Raspberry Pi, de modo que al energizar la placa arranca el SO, tal y como sucede en los ordenadores convencionales. Los pasos a seguir para la instalación se detallan a continuación:

- a. Descargar Raspbian Jessie de la página <https://www.raspberrypi.org/downloads/raspbian/>
- b. Formatear la tarjeta Micro SD, usando el programa SDFormater
- c. Cargar el SO en la tarjeta Micro SD, con el programa Win32 Disk Imager

La secuencia para la configuración del sistema operativo Raspbian Jessie:

1. Actualizar y limpiar el sistema, para lo cual se digitará en el LXTerminal los comandos:
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
2. Finalizado el ítem anterior se procede a modificar parámetros de la Raspberry, ingresando el comando `sudo raspi-config`, aparecerá una pantalla como indica la **Figura 1-2**.
3. En las dos primeras opciones se conserva el nombre y contraseña que vienen definidas como pi y raspberry respectivamente.

4. En Boot Option no se realizan cambios.
5. Dentro de Localisation Option se modificó la zona horaria y se activó el Wi-fi de la Raspberry Pi.
6. En Interfacing options se habilitaron: el conector CSI para el módulo de cámara raspberry, el puerto serial y los puertos GPIO.
7. No se realizaron cambios dentro de Overclock.
8. En Advanced se expandió la memoria para usar todo el espacio de almacenamiento.
9. Las dos últimas opciones no se realizaron cambios.
10. Para configurar la red IP fija, se usa el comando `sudo nano /etc/network/interfaces`.
11. Finalmente se reinicia la Raspberry para que se apliquen los cambios realizados.

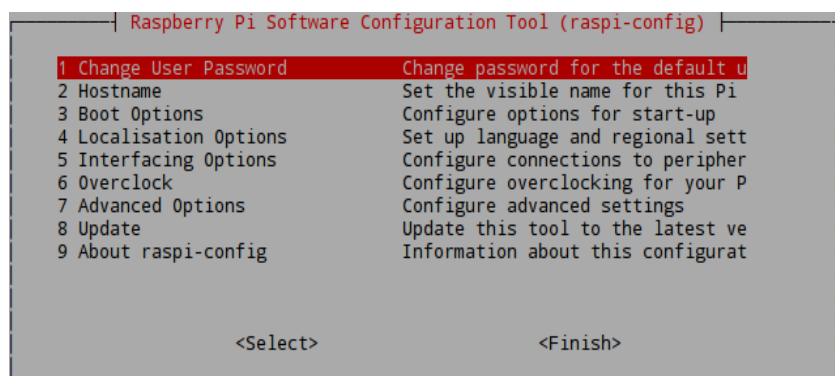


Figura 1-2: Configuración del SO Raspbian Jessie

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Una vez configurada la placa en el menú Raspi-Config, se procede a instalar el resto de aplicaciones y librerías requeridas, desde el LXTerminal. La **Figura 2-2** muestra el escritorio de la Raspberry Pi.

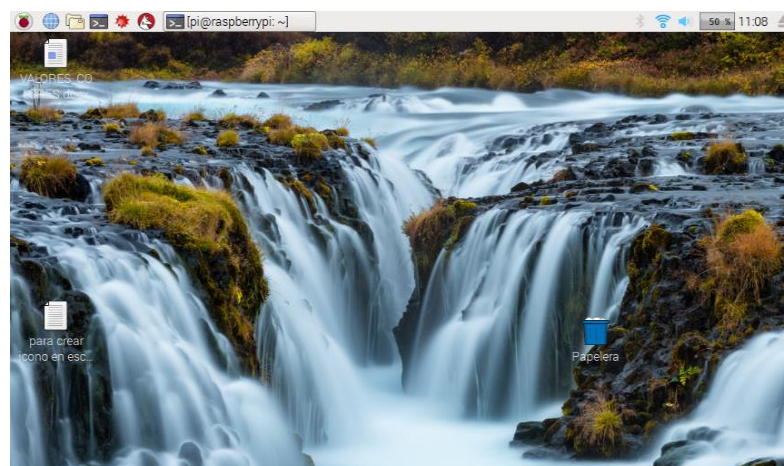


Figura 2-2: Escritorio de Raspberry Pi

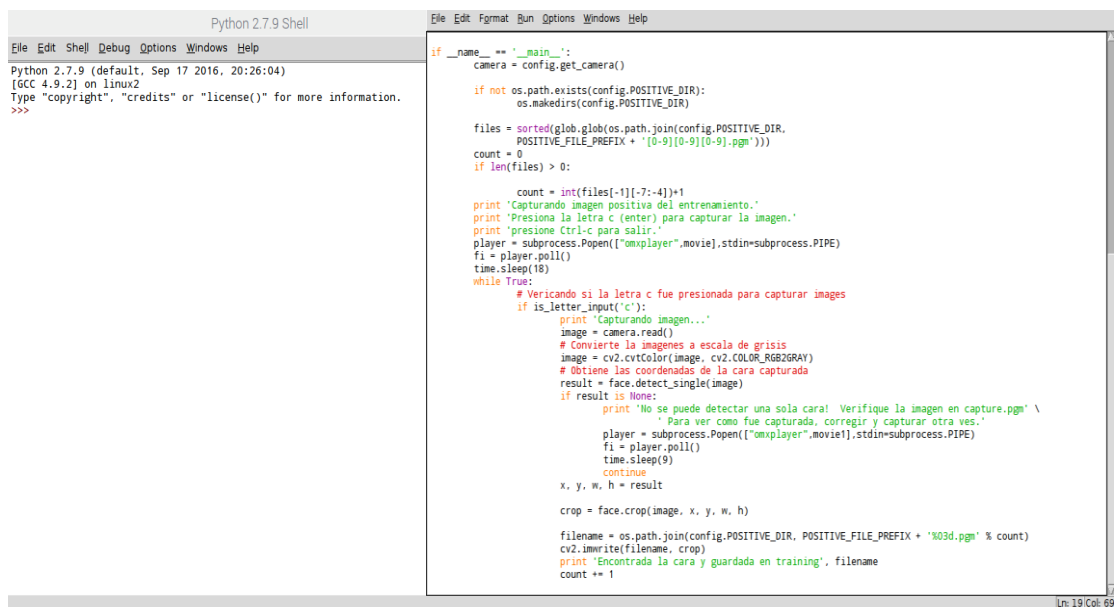
Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.1.2 Instalación de Python

Precedentemente se actualiza el sistema para instalar Python 2.7, en el LXTerminal se digitaliza las siguientes líneas:

```
sudo apt-get update
sudo apt-get install python-pip python3-pip
```

La **Figura 3-2** muestra de ventana del Shell que es el intérprete de comandos y el editor la ventana en la cual el usuario crea su programa.



```
Python 2.7.9 Shell
File Edit Shell Debug Options Windows Help
Python 2.7.9 (default, Sep 17 2016, 20:26:04)
[GCC 4.9.2] on linux2
Type "copyright", "credits" or "license()" for more information.
>>>

if __name__ == '__main__':
    camera = config.get_camera()
    if not os.path.exists(config.POSITIVE_DIR):
        os.makedirs(config.POSITIVE_DIR)

    files = sorted(glob(os.path.join(config.POSITIVE_DIR,
        POSITIVE_FILE_PREFIX + '[0-9][0-9][0-9].pgm')))
    count = 0
    if len(files) > 0:
        count = int(files[-1][-7:-4])+1
    print 'Capturando imagen positiva del entrenamiento.'
    print 'Presiona la letra c (enter) para capturar la imagen.'
    print 'Presione Ctrl-c para salir.'
    player = subprocess.Popen(["omplayer",movie],stdin=subprocess.PIPE)
    fi = player.poll()
    time.sleep(18)
    while True:
        # Verificando si la letra c fue presionada para capturar imagenes
        if is_letter_input('c'):
            print 'capturando imagen...'
            image = camera.read()
            # Convierte la imagenes a escala de gris
            image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
            # Obtiene las coordenadas de la cara capturada
            result = face.detect_single(image)
            if result is None:
                print 'No se puede detectar una sola cara! Verifique la imagen en capture.pgm \
                    ' Para ver como fue capturada, corregir y capturar otra vez.'
                player = subprocess.Popen(["omplayer",movie],stdin=subprocess.PIPE)
                fi = player.poll()
                time.sleep(9)
                continue
            x, y, w, h = result
            crop = face.crop(image, x, y, w, h)

            filename = os.path.join(config.POSITIVE_DIR, POSITIVE_FILE_PREFIX + '%03d.pgm' % count)
            cv2.imwrite(filename, crop)
            print 'Encontrada la cara y guardada en training', filename
            count += 1
```

Figura 3-2: Ventana shell y editor de Python

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.1.3 Instalación de OpenCv

Para instalar el paquete de visión artificial OpenCv se debe actualizar y limpiar el sistema, luego descargar el software desde internet y se procede a instalar en la Raspberry Pi de la siguiente manera:

```
sudo apt-get update
sudo apt-get install python-opencv libopencv-dev
```

Para verificar los paquetes y librerías instaladas sobre la placa, como se observa en **Figura 4-2** se emplea el comando `dpkg --get-selections`.

```
libopencv-dev install
libopencv-features2d-dev:armhf install
libopencv-features2d2.4:armhf install
libopencv-flann-dev:armhf install
libopencv-flann2.4:armhf install
libopencv-gpu-dev:armhf install
libopencv-gpu2.4:armhf install
libopencv-highgui-dev:armhf install
libopencv-highgui2.4:armhf install
libopencv-imgproc-dev:armhf install
libopencv-imgproc2.4:armhf install
libopencv-legacy-dev:armhf install
libopencv-legacy2.4:armhf install
libopencv-ml-dev:armhf install
libopencv-ml2.4:armhf install
libopencv-objdetect-dev:armhf install
libopencv-objdetect2.4:armhf install
libopencv-ocl-dev:armhf install
libopencv-ocl2.4:armhf install
libopencv-photo-dev:armhf install
libopencv-photo2.4:armhf install
libopencv-stitching-dev:armhf install
libopencv-stitching2.4:armhf install
libopencv-superres-dev:armhf install
libopencv-superres2.4:armhf install
libopencv-ts-dev:armhf install
```

Figura 4-2: Librerías de OpenCv

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Algunas de las librerías y dependencias de OpenCv instaladas en la Raspberry Pi 3 se evidencian en la tabla consecuyente.

Tabla 1-2: Principales librerías de OpenCv instaladas

PRINCIPALES LIBRERÍAS INSTALADAS				
libvavcinfo0	libpreccpp0	libv4lconvert0	Libtext	libudev0
libvlccore8	Libopencv	libv8-3.14.5	Libpaper	libudisks2-0
libvision-0.1-1	Libwayland	libopenexr-dev	libopencv2.4	libupnp6
libvision-0.4-0	libvte9	libhyphen0	libparams-utils	liborcus-0.8-0
libvnccliente0	libtcl8.6	libimlib2	libopencv2.4-jni	libusb -1.0-0
libwebcam0	libthai-date	libtiff5-dev	liborbit-0-2-0	libutempter0
libsidplay2	libtimedate-per1	Libwinpr	libodfgen-0.1-1	libopencv2.4-java

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.2 Selección de elementos

Los elementos fueron seleccionados en base varios criterios como: compatibilidad entre elementos, tiempos de respuesta, funcionalidad, tipo de licencia y costo económico. Tanto la placa Raspberry Pi 3 B+ como el arduino son de código abierto, así como la mayoría de componentes.

2.2.1 Selección de motores

Para el movimiento del juguete inteligente se eligió los motores eleva vidrios eléctricos. Este actuador es universal, funcionan a 12 VDC y soportan una corriente máxima de 7A, tiene incorporada la caja reductora. Ideal para la locomoción debido a que su torque es mayor a la velocidad, es decir, presenta mayor fuerza y menor rapidez de desplazamiento.

La **Figura 5-2** ilustra el motor encargado de la movilidad del robot, cuya velocidad angular es de 2750 rpm, valor dentro del rango admisible en robótica (1000 a 3000 rpm).



Figura 5-2: Motor eleva vidrios Nemesys

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.2.2 Selección de cámara

La cámara es uno de los componentes de suma significancia, ya que es la encargada de capturar imágenes o grabar videos del medio exterior, para ser procesadas con métodos computacionales y extraer las características necesarias para la toma de decisiones.

La interacción del robot con el niño debe ser en tiempo real, para evitar que el impúber pierda el interés y no dar cumplimiento a los objetivos. Por dicha razón se realizó una prueba en la placa Raspberry para estimar el tiempo de respuesta de tres cámaras idóneas.

Tabla 2-2: Tiempo de reacción de las cámaras

Tiempo respuesta Imagen capturada	SENSOR KINECT	CÁMARA USB LOGITECH C920	MÓDULO DE CÁMARA PI
Imagen 1	3221 ms	2023 ms	28 ms
Imagen 2	3044 ms	1947 ms	32 ms
Imagen 3	3290 ms	2020 ms	29 ms
Imagen 4	3033 ms	1970 ms	27 ms
Imagen 5	3380 ms	1950 ms	28 ms
Imagen 6	3450 ms	1929 ms	31 ms
Imagen 7	3058 ms	1980 ms	27 ms
Imagen 8	3250 ms	2014 ms	27 ms
Imagen 9	3003 ms	1930 ms	27 ms
Imagen 10	3410 ms	1986 ms	29 ms

Realizado por: Salinas Lissette, Chamorro Edison, 2017

En base a los datos de la **Tabla 2-2** la cámara pi tiene los mejores tiempos de respuesta, frente al sensor Kinect y la cámara Logitech C920, la razón se debe a que es un módulo externo propio de Raspberry. Por tanto la cámara pi integrará el juguete inteligente.

El módulo de cámara Raspberry Pi funciona igual a un sensor omnidireccional, crea efectos especiales mediante bibliotecas compatibles con la cámara. Posee un sensor IMX219 de 8 megapíxeles, apto para filmar videos y captar imágenes. Para su conexión usa 15 pines para conectarse a la placa con cinta flat por medio de puerto DSI. (Raspberry, 2017).

En la **Figura 6-2** se presenta el módulo de cámara Raspberry Pi, cuenta con una resolución de 2592 x 1994 píxeles. (Raspberry, 2017)



Figura 6-2: Módulo de cámara Raspberry Pi

Realizado por: Salinas Lissette, Chamorro Marcelo, 2017

2.2.3 Selección de sensores

El sistema sensorial estará compuesto por sensores ultrasónicos HC-SR04; debido a su bajo costo, fácil uso, compatibilidad con arduino, amplio rango de detección y ausencia de inconvenientes debido a los niveles de luz, como suele suceder con los infrarrojos. (Super Robótica, 2017)

HC-SR04 detecta la presencia de algún objeto y determina la distancia a la que se encuentra. Tiene un emisor y receptor, funciona emitiendo un sonido y calcula el tiempo que tarda en regresar el eco. En la **Figura 7-2** se visualiza el sensor HC-SR04. Su ángulo de medición es de 30°, trabaja a 40 KHz y detecta objetos hasta 4m de distancia. (Super Robótica, 2017)



Figura 7-2: Sensor ultrasónico HC-SRB04

Realizado por: Salinas Lissette, Chamorro Marcelo, 2017

2.2.4 Selección de baterías

Se ha escogido las baterías compuestas por litio y polímero para alimentar a los elementos electrónicos. Debido a sus singulares propiedades, cada día se expanden en la industria; es frecuente ver este tipo de baterías en celulares, computadoras, tablets y demás dispositivos portátiles. En la **Tabla 3-2** se describen las ventajas y desventajas de estas baterías.

Tabla 3-2: Ventajas y desventajas de las baterías lipo.

VENTAJAS	DESVENTAJAS
Mayor densidad de energía	Delicadas
Livianas	Costosas
Alto nivel de carga	Inflamable
Mayor durabilidad	---
Resiste pérdidas de cargas	---
Tamaño reducido	---
Recargables	---

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Fuente: Baterías lipo, ventajas y desventajas, 2015

Analizando brevemente las ventajas frente a las desventajas es indudable que la batería Lipo es la mejor opción, pese a ser inflamable como todas las baterías. La solución a este inconveniente es la precaución y el adecuado uso de la misma, por tal razón las baterías deben estar protegidas para evitar el contacto directo o indirecto con los infantes.

Las baterías como se aprecian en la **Figura 8-2** están compuestas por Litio y Polímero, son recargables, ligeras y están formadas por celdas (S) conectadas en serie. Cada celda contiene 3.7 voltios.



Figura 8-2: Batería Lipo Turnigy 3S

Realizado por: Salinas, Lissette; Edison, Chamorro, 2017

2.2.4.1 Duración de la batería

En el juguete inteligente se usan dos baterías, la una para alimentar la placa Raspberry Pi y la otra conectada al sistema de locomoción. Para estimar el tiempo de duración de carga se reemplazan valores en la Ecuación 15, considerando el 25% de reserva por cualquier circunstancia.

$$tiempo\ uso = \frac{potencia\ hora\ de\ la\ batería}{potencia\ consumida\ de\ dispositivos} - 25\% \quad \text{Ecuación 15.}$$

La batería conectada el minicomputador es de 3S (11.1V) con una corriente de 220mh, equivale a una potencia de 24.42 Wh. Mientras la placa presenta un consumo máximo de 1.8W.

$$t_1 = \frac{24.42\ Wh}{1.8\ W} - 25\% = 10\ horas$$

La batería que suministra energía al sistema de locomoción tiene una potencia de 44.4 Wh (3S y 400mh) y los actuadores presenta un consumo máximo de 33.3W.

$$t_1 = \frac{44.4\ Wh}{33.4\ W} - 25\% = 1\ hora$$

2.3 Diseño e implementación de prototipo del juguete inteligente

2.3.1 Diseño del prototipo

Las características a considerar antes de construir el prototipo son: apariencia, dimensiones, color, peso y el material. Por tratarse de un robot didáctico infantil debe ser atractivo, colorido, liviano, fácil de transportar y resistente a golpes.

El diseño del robot es totalmente inédito, el modelamiento de las piezas se realizó en SolidWorks. La estructura consta de tres partes fundamentales: chasis, cuerpo y cabeza. En la parte posterior del cuerpo y la cabeza van tapas, los espacios destinados a las tapas servirá para ingresar los componentes electrónicos.

El prototipo tiene forma de un muñeco extraterrestre, con dimensiones de 50x30cm, 17cm de profundidad y 3mm de espesor. La **Figura 9-2** indica la estructura del juguete inteligente.

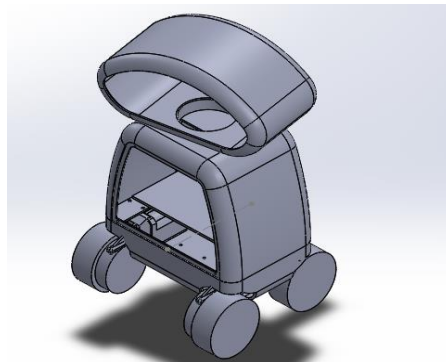


Figura 9-2: Ensamblaje del robot

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El chasis de la **Figura 10-2** sujeta a los motores, fija al driver y arduino para control de movimientos del sistema de locomoción, sostiene al resto de la estructura.

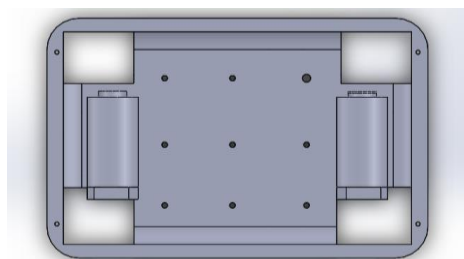


Figura 10-2: Chasis del robot

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El cuerpo es la pieza central, representa al tronco del cuerpo humano, como se observa en la **Figura 11-2**. La parte interna de esta pieza alberga el cerebro del robot (Raspberry), la pantalla digital, parlante, baterías lipos y reguladores de tensión.



Figura 11-2: Cuerpo del robot

Realizado por: Salinas Lissette, Chamorro Edison, 2017

La **Figura 12-2** ilustra la tapa posterior del cuerpo.

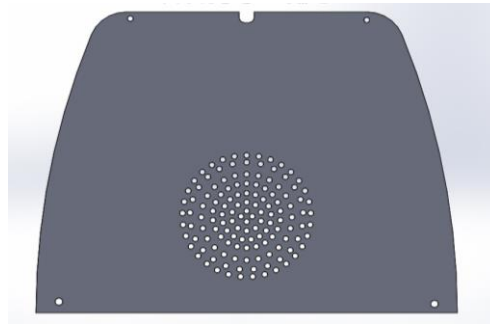


Figura 12-2: Tapa posterior del cuerpo

Realizado por: Salinas Lissette, Chamorro Edison, 2017

La cabeza tienen una forma excepcional como se muestra en la **Figura 13-2**, en el interior se situará el módulo de la cámara. En la vista frontal se colocará acrílico con agujeros dándole el aspecto de ojos y led que forman la boca, nariz y mejillas. La parte posterior será cubierta con una tapa impresa en 3D.

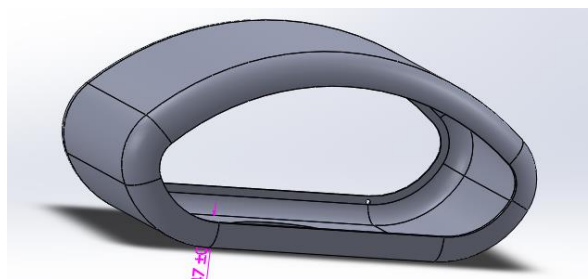


Figura 13-2: Cabeza del robot

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.3.2 Implementación de prototipo

Teniendo listo el diseño se procede a la impresión 3D. El material usado para la impresión es PLA, un producto natural e inocuo para la salud, hecho a base de maíz, no imite ningún tipo de gas tóxico lo cual le hace amigable con el medio ambiente.

En la **Figura 14-2** se observa la impresión en 3D de la cabeza.

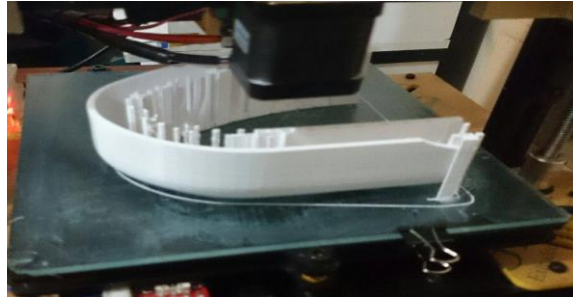


Figura 14-2: Impresión 3D de piezas modeladas

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Concluida la impresión se ensambla las piezas, para lo cual se unió con pegamento. Fijas las piezas se pintaron tal que se vea colorido y capte la atención del niño, luego se colocaron todos los componentes. La **Figura 15-2** muestra el prototipo terminado.



Figura 15-2: Juguete Inteligente

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El tipo de pintura usado fue aerosol biodegradable, compuesto a base de aceites minerales y arcillas, los pigmentos son minerales libres de plomo y cadmio, lo cual no causa daño al medio ambiente ni perjudica la salud del infante.

2.4 Diseño del sistema de visión artificial

Antes de diseñar el sistema de visión artificial para el procesamiento inteligente de imágenes, se debe seleccionar la técnica que cumpla los requerimientos necesarios. Para lo cual se analiza los pros y contras de cada una de las técnicas que integran la soft computing.

Tabla 4-2: Ventajas y desventajas de los métodos que integral soft computing.

VENTAJAS Y DESVENTAJAS DE COMPONENTES DE SOFT COMPUTING		
MÉTODO	VENTAJA	DESVENTAJA
Método Probabilístico	Trabaja con datos perdidos	Alto tiempo de evaluación
	Representa información gráficamente	Requiere gran cantidad de datos
	Realizar inferencias bidireccionales	Poco flexibilidad
	Optimiza recursos	
	Usa operadores probabilísticos	
	Reduce el sobre ajuste de los datos	
Lógica Difusa	Resuelve problemas no lineales	Dificulta la interpretación de valores difusos
	No necesita modelo matemático	Elevada cantidad de cálculos
	Tolera perturbaciones	Dificultad para modelar un sistema difuso
	Estable y preciso	Poca flexibilidad
	Alta velocidad de respuesta	
	Trabaja con información incompleta	
Redes Neuronales	Buen aprendizaje	No es óptimo para ejecutar varias tareas
	Auto organización	Tiempo de aprendizaje elevado
	Tolerancia a fallos	No permite interpretaciones
	Flexibilidad	Gran cantidad de datos para el entrenamiento
	Trabaja en Tiempo real	Alto consumo de CPU
	No necesita conocer cálculos matemáticos	
Algoritmos Genéticos	No necesita conocer a fondo el problema	Tarda mucho en converger
	Opera simultáneamente con varias soluciones	Converge prematuramente
	Fácil de ejecutar	Alto consumo de CPU
	Usa operadores probabilísticos	
	Eficaz para problemas de optimización	
	Reduce máximos locales	

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Fuente: De los conjuntos fuzzy a la soft computing, 2005

Las técnicas de soft computing se caracterizan por tratar datos imprecisos y consumir gran cantidad de CPU. El modelo probabilístico optimiza recursos pero requiere tiempo para la evaluación. La lógica difusa resuelve procesos no lineales y complejos pero con demasiados cálculos. Las redes neuronales son excelentes para el aprendizaje, pese al tiempo para entrenar. Los algoritmos genéticos reducen los máximos locales, aunque tarde mucho en converger.

El tiempo de respuesta debe ser el más bajo posible, de ser mejor en tiempo real. No obstante cada método aporta de diferente manera, un mismo modelo puede ser óptimo para ciertas aplicaciones y pésimo para otros, sin olvidar que es mejor satisfacer antes que optimizar.

Tabla 5-2: Aplicaciones de los métodos que integran la Soft Computing

ÁREA DE APLICACIÓN DE LA SOFT COMPUTING				
Área	Modelo Probabilístico	Redes Neuronales	Lógica Difusa	Algoritmos Genéticos
Medicina	Monitoreo de pacientes	Diagnóstico tipo de cáncer	Diagnóstico médico	Predicción de estructura ARN
Biométrica y Bioinformática	Reguladores de genes y estructura de proteínas	Identificación de voz, imágenes y huellas	Identificación de escritura	Construcción de árboles biogenéticos
Economía	Analiza bolsa de valores y patrones de compra	Automatiza procesos financieros	Ninguna	Automatiza procesos financieros
Informática	Motores de búsqueda de Google y diagnóstica fallos de computador	Spam de correo electrónico, reconocimiento de patrones	Base de datos difusa, compresión de datos y seguridad informática	Diseño de redes computacionales
Industria	Automatización de procesos industriales	Automatización de procesos industriales	Control de cámaras, aire acondicionado, microondas, trenes y elevadoras	Optimización Layout y automatización de procesos industriales
Robótica	Procesamiento de imágenes	Reconocimiento de patrones	Procesamiento de imágenes	Aprendizaje de comportamiento de robot
Automotriz	Ninguna	Control de tráfico	Sistema de frenos, transmisión y ahorro de combustible	Conducción automática
Otros	Juegos de azar y meteorología	Juegos y química cuántica	Juegos electrodoméstico	Juegos, ganadería y agricultura

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Fuente: De los conjuntos fuzzy a la soft computing, 2005

Los modelos de la soft computing están inmersos en todas las áreas, desempeñando diferente labor. Las hibridaciones de dos o más técnicas dan como resultado sistemas expertos con mejores resultados y mayor cobertura, siendo los algoritmos para juegos los más significativos.

El procesamiento inteligente de imágenes se divide en dos etapas, la primera el reconocimiento facial y la segunda el reconocimiento de color y figura geométrica de cuerpos 2D. En base a los contenidos de la **Tabla 4-2** y la **Tabla 5-2**, el método adecuado para el reconocimiento facial son las redes neuronales, debido a su corto tiempo de ejecución y excelente desempeño en el campo de biométrica.

Para la etapa de reconocimiento de colores y figuras geométricas se ha optado resolver por visión artificial, dado que si se elige algún método robusto agotaría los recursos de hardware, provocando lentitud en los procesos.

2.4.1 Sistema de reconocimiento facial

El sistema de reconocimiento facial es una adaptación del modelo propuesto por Viola y Jones (2014), en el cual se emplea una red neuronal, entrenada con el algoritmo Adaboost y clasificadores cascada tipo haar. Dicha estudio se detalla en la sección **1.5** del Capítulo I.

En el diagrama de bloques de la **Figura 16-2**, la cámara pi toma una fotografía en escala de grises al niño que se encuentre frente al lente, luego transforma la toma a una imagen integral. En la tercera fase se extraen los rasgos de la cara usando el algoritmo eigenfaces y filtros haar. Finalmente se entrenan los clasificadores cascada con el algoritmo adaboost.

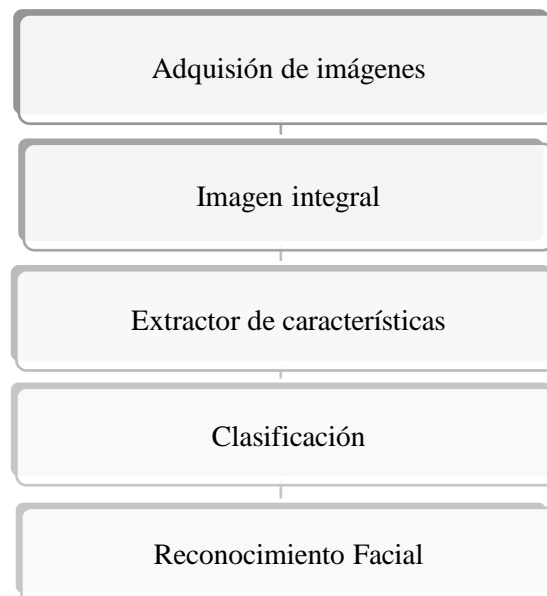


Figura 16-2: Diagrama para el reconocimiento facial

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Antes de empezar a programar se debe verificar que están correctamente instaladas las funciones y dependencias requeridas, al igual que en todos los lenguajes de programación se empieza declarando las librerías.

La **Tabla 6-2** describe la estructura del sistema de reconocimiento de rostros implementada, formado por subprogramas y funciones, cada uno es responsable de ejecutar una tarea específica. En conjunto permiten el funcionamiento global.

Tabla 6-2: Subprogramas y funciones que aportan al sistema de reconocimiento facial

RECONOCIMIENTO FACIAL		
Subprograma	Descripción	Aporte
Config	Función de configuración parámetros importantes	<ul style="list-style-type: none"> • Habilita puerto GPIO • Configura las imágenes • Configura el clasificador cascada • Configura la cámara • Direcciona archivos • Crea las etiquetas
Face	Funciones de ayuda	<ul style="list-style-type: none"> • Crea la imagen integral • Verifica características del clasificador cascada
cargar_datos	Función para optimizar tiempo	<ul style="list-style-type: none"> • Carga los datos del entrenamiento
capture-positives	Captura imágenes y genera base de datos	<ul style="list-style-type: none"> • Captura las imágenes con rostro • Crea base de datos de imágenes positivas
Train	Entrenamiento de la Red Neuronal	<ul style="list-style-type: none"> • Entrena la red neuronal con el algoritmo Adaboost • Extraer las características faciales con algoritmo Eigenfaces
reconocimiento	Reconocimiento de individuos Identificados	<ul style="list-style-type: none"> • Captura la imagen • Extrae características • Reconoce rostros

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El subprograma *config.py* alberga las configuraciones principales. En el cual se habilitan los puertos GPIO, se configura la cámara, se crean las etiquetas para las imágenes que intervienen en el entrenamiento, así como las dimensiones de las mismas y se define el umbral perceptónico, indispensable para la identificación de individuos.

También se direcciona el archivo *train.xml* que se genera al momento de ejecutar el subprograma *train.py*. El código usado se exhibe en la **Figura 17-2**, en las dos últimas líneas se activa la cámara pi. Para definir una función se emplea la instrucción *def nombre_función ()*:

```
# Etiquetas de las imagenes capturadas para realizar el entrenamiento
POSITIVE_LABEL = 1
NEGATIVE_LABEL = 2
LEO_LABEL = 3
SAMMY_LABEL = 4

# Se define el ancho y alto de la cara, estos valores no se deben cambiar
FACE_WIDTH = 92
FACE_HEIGHT = 112

# Configuración de clasificador cascada para el reconocimiento facial
HAAR_FACES = 'haarcascade_frontalface_alt.xml'
HAAR_SCALE_FACTOR = 1.3
HAAR_MIN_NEIGHBORS = 4
HAAR_MIN_SIZE = (30, 30)

# Nombre para almacenar la ultima imagen capturada
DEBUG_IMAGE = 'capture.pgm'

def get_camera():
    # Configuración del modulo de cámara raspberry para capturar imágenes
    import picam
    return picam.OpenCVCapture()
```

Figura 17-2: Subprograma de configuración

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El subprograma *face.py* ayuda al reconocimiento facial, contiene la función del clasificador cascada y la imagen integral. Una de las ventajas de OpenCv es la inclusión de clasificadores pre-entrenados, como haar cascade y eyes cascade, los cuales se descargan como archivo.xml.

```
import cv2
import config

haar_faces = cv2.CascadeClassifier(config.HAAR_FACES)
```

Figura 18-2: Calificador cascada en OpenCv

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Luego se carga la imagen de entrada en escala de grises, si se encuentran caras la función devuelve un valor positivo, caso contrario retorna un índice negado. La conversión de las tomas a imagen integral se muestra en la **Figura 19-2**

```
def crop(image, x, y, w, h):
    # para la imagen integral
    crop_height = int((config.FACE_HEIGHT / float(config.FACE_WIDTH)) * w)
    midy = y + h/2
    y1 = max(0, midy-crop_height/2)
    y2 = min(image.shape[0]-1, midy+crop_height/2)
    return image[y1:y2, x:x+w]
```

Figura 19-2: Conversión de imagen normal a imagen integral

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Antes de iniciar el juego, el robot debe identificar al niño. Se crea la base de datos con información de los participantes. La cantidad de datos es controlada por el técnico, cada que se presione la tecla C se almacena una imagen únicamente del rostro, si la captura no presenta cara se rechaza y emite un mensaje de negación.

La **Figura 20-2** muestra el subprograma que captura imágenes positivas para el entrenamiento del juguete inteligente.

```
if is_letter_input('c'):
    print 'Capturando imagen...'
    image = camera.read()
    # Convierte la imagenes a escala de grisis
    image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    # Obtiene las coordenadas de la cara capturada
    result = face.detect_single(image)
    if result is None:
        print 'No se puede detectar una sola cara! Verifique la imagen en capture.pgm'
        print ' Para ver como fue capturada, corregir y capturar otra ves.'
        player = subprocess.Popen(["omxplayer",movie1],stdin=subprocess.PIPE)
        fi = player.poll()
        time.sleep(9)
        continue
    x, y, w, h = result

    crop = face.crop(image, x, y, w, h)
```

Figura 20-2: Captura de imágenes positivas para la base de datos

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El entrenamiento consiste en la clasificación de imágenes positivas y negativas, empleando el modelo iterativo de Bossting. Una de las bondades de OpenCV es la inclusión de toolbox, como eigenfaces para extraer los rasgos faciales y clasificadores cascada.

La **Figura 21-2** muestra la función para iterar los archivos de la base de datos.

```
def walk_files(directory, match='*'):
    """Funcion para iterar archivos de una determinada direccion
    """
    for root, dirs, files in os.walk(directory):
        for filename in fnmatch.filter(files, match):
            yield os.path.join(root, filename)

def prepare_image(filename):
    return face.resize(cv2.imread(filename, cv2.IMREAD_GRAYSCALE))
```

Figura 21-2: Función para iterar archivos

Realizado por: Salinas Lissette, Chamorro Edison, 2017

La **Figura 22-2** exhibe las instrucciones para hallar la media de las facciones y en base a ello determinar si pertenece el rostro a un sujeto conocido.

```
# calcula la media de la cara, para el modelo de reconocimiento de cara
mean = model.getMat("mean").reshape(faces[0].shape)
cv2.imwrite(MEAN_FILE, normalize(mean, 0, 255, dtype=np.uint8))
eigenvectors = model.getMat("eigenvectors")
pos_eigenvector = eigenvectors[:,0].reshape(faces[0].shape)
cv2.imwrite(POSITIVE_EIGENFACE_FILE, normalize(pos_eigenvector, 0, 255, dtype=np.uint8))
neg_eigenvector = eigenvectors[:,1].reshape(faces[0].shape)
```

Figura 22-2: Código para hallar la media de los rasgos faciales

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El reconocimiento comprende la identificación de los niños, se realiza una comparación de la media de las características con el umbral perceptrónico definido en el subprograma de configuración. El proceso se realiza con una imagen que contenga cara, de no ser así se emite un mensaje indicando que se debe fijar la vista a la cámara o asegurarse que sea un humano, como medida de seguridad no son admisibles retratos o fotografías.

```
crop = face.resize(face.crop(image, x, y, w, h))
# Pruebe la cara contra el modelo predictivo.
label, confidence = model.predict(crop) # uso del modelo predictivo

print 'Predicted {0} face with confidence {1} (lower is more confident)'.format(
    'POSITIVE' if label == config.POSITIVE_LABEL else 'NEGATIVE',
    confidence)
```

Figura 23-2: Cálculo del umbral predictivo

Realizado por: Salinas Lissette, Chamorro Edison, 2017

De ser menor el umbral predictivo al umbral perceptrónico (`positive_threshold` en `config.py`) el robot dirá el nombre del niño o niña, e inmediatamente se enlaza con el sistema para el reconocimiento de figuras y colores. Debido al margen de error producido por la luminosidad y el tipo de pose, se admiten varios intentos seguidos, si en todos resulta negativo no avanza a la etapa subsiguiente.

2.4.2 *Sistema de reconocimiento de colores y figuras geométricas*

El software para la identificación de colores y figuras geométricas se aplica en ambientes controlados, después de haber identificado el rostro del niño. La cámara pi graba video y preprocesa la imagen, aplicando filtros gaussianos para eliminar el ruido y así poder detectar el contorno del objeto, seguidamente se extrae la información para clasificar y finalmente imprimir el color y figura del cuerpo en estudio.

El diagrama de bloques del sistema para la identificación de colores y figuras geométricas se aprecia en la **Figura 24-2**.

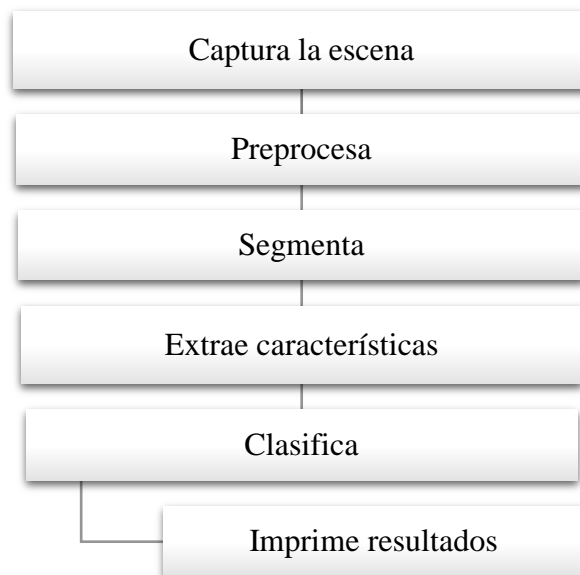


Figura 24-2: Sistema para reconocimiento de colores y formas

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Al igual que el reconocimiento fácil este sistema está compuesto por varios subprogramas, como se indica en la **Tabla 7-2**.

Tabla 7-2: Composición del sistema para identificar el color y forma

SISTEMA DE RECONOCIMIENTO DE COLORES Y FIGURAS GEOMÉTRICAS		
Subprograma	Descripción	Detalle
filtro_tiempo_real	Halla rangos de colores	<ul style="list-style-type: none"> Halla el rango de los colores
colorlaber	Función color	<ul style="list-style-type: none"> Ayuda a la detección de color
shapedetector	Función figuras	<ul style="list-style-type: none"> Ayuda a detectar la figura del contorno
figuras	Determinar el color y figura geométrica de los objetos	<ul style="list-style-type: none"> Captura las imágenes Aplica Filtros Extracción de características Determina el color y forma

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El programa está diseñado para reconocer colores primarios y secundarios en diferentes tonos, desechando aquello que está fuera de rango de tonalidad, de esta manera se enfoca en el borde para determinar la forma del cuerpo.

Para reconocer los colores se siguen los siguientes pasos:

- ✓ Captura la imagen
- ✓ Convierte la imagen a formato RGB
- ✓ Busca los objetos de los colores deseados
- ✓ Elimina el ruido
- ✓ Muestra en pantalla la imagen

Lo primero es definir los rangos de colores que desean ser identificados, como se observa en la **Figura 25-2**.

```
##Rojo
lower_red = np.array([164, 119, 144])
upper_red = np.array([179, 145, 171])
#Naranja
lower_orange = np.array([169, 110, 140])
upper_orange = np.array([175, 130, 175])
#Azul
lower_blue = np.array([110, 102, 112])
upper_blue = np.array([127, 147, 145])
#Verde
lower_green = np.array([112, 44, 100])
upper_green = np.array([125, 66, 135])
#Amarillo
lower_yellow = np.array([18, 60, 106])
upper_yellow = np.array([31, 85, 135])
###Purpura
lower_purple = np.array([ 0, 46, 110])
upper_purple = np.array([ 150, 70, 130])
```

Figura 25-2: Definición de colores a ser identificados

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Definidos los niveles de colores se crean máscaras para encontrar el contorno. La **Figura 26-2** calcula el momento y las áreas de las zonas marcadas, conservando únicamente la de mayor superficie, es decir, elimina el ruido.

```
cnts = cv2.findContours(mask.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
cnts = cnts[0] if imutils.is_cv2() else cnts[1]
### Buscamos en todos los contornos
for c in cnts:
# Calculamos el centro del contorno, entonces se detecta la forma usando
# solamente el contorno
    M = cv2.moments(c)
    if (M["m00"]>area_mayor):
        indice_mayor = i
        lower_mask2 = lower
        upper_mask2 = upper
        area_mayor=M["m00"]
```

Figura 26-2: Elimina el ruido

Realizado por: Salinas Lissette, Chamorro Edison, 2017

En la etapa pre-operacional se estimula la percepción de colores y formas. Como inducción a la educación primaria los niños deben reconocer figuras geométricas como: cuadrado, triángulo, círculo, rectángulo y distinguir colores básicos (primarios y secundarios), además asociarlos con lo que los rodea. Tal como lo afirman psicólogos (Bargh, 2012), educadores, Ángel Pérez Gómez (2012) y el Currículo de Educación Básica de Ecuador (2016).

El programa principal anuncia las funciones *colorlaber* y *shapedetector*, encargados de estipular el color y figura geométrica respectivamente. Compara con el objeto en análisis, decide cuál de los descriptores le corresponde e imprime en pantalla, como se muestra en la **Figura 27-2**.

```
if (forma=="Rectangulo"):
    print 'RECTANGULO'
elif (forma=="Triangulo"):
    print 'TRIANGULO'
elif (forma=="Cuadrado"):
    print 'CUADRADO'
elif (forma=="Circulo"):
    print 'CIRCULO'
if (indice_color==0):
    print 'AMARILLO'
elif (indice_color==1):
    print 'AZUL'
elif (indice_color==2):
    print 'MORADO'
elif (indice_color==3):
    print 'NARANJA'
```

Figura 27-2: Determina el color y figura de la imagen

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.5 Sistema para control de movimientos

Los movimientos son controlados por sensores ultrasónicos. La planificación de trayectoria está sujeto al tipo de ruedas, la cantidad y la dirección del eje de rotación. (Rentería & Rivas, 2000)

2.5.1 Sistema de locomoción

El juguete inteligente tiene un sistema de locomoción diferencial, compuesto de cuatro ruedas no directrices, como se observa en la **Figura 28-2**. Las llantas posteriores se ajustan a los motores que reposan en la base del chasis, mientras las llantas del frente están unidas por un eje metálico.

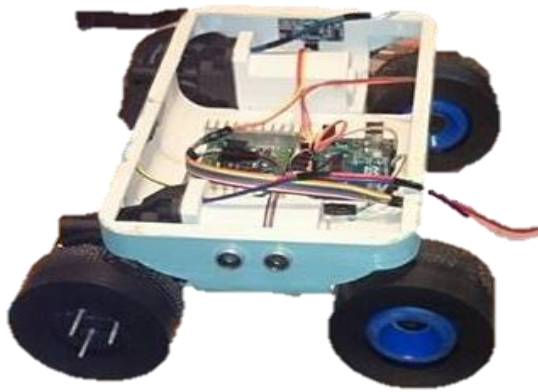


Figura 28-2: Locomoción del juguete inteligente

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Este diseño garantiza estabilidad, tracción a las ruedas delanteras para ejercer movimiento. Tolerancia el desplazamiento sobre superficies planas y rectas, no puede escalar ni realizar movimientos extravagantes. Avanza y retrocede con facilidad, para los giros se controla la velocidad de los motores.

2.5.2 *Sistema sensorial*

El sistema sensorial está compuesto de seis sensores ultrasónicos, distribuidos en el contorno del prototipo para evitar colisiones, si es prudente esquiva los obstáculos y continúa o simplemente se detiene a cierta distancia. Los sensores 1 y 2 anversos son los de mayor prioridad. Los sensores 3 y 4 están en los costados, los sensores 5 y 6 se localizan en las esquinas delanteras.

El robot debe posicionarse siempre a 50 cm del niño, ya que a esa longitud la cámara captura imágenes sin problema y científicamente es recomendable conservar cierta distancia con las máquinas por cuestión de seguridad.

Mientras la variable a (switch) esté encendida, $S1$ y $S2$ (sensor 1, 2) entran en funcionamiento, cuando detecten algo en el rango de 45 a 50 cm se detiene, caso contrario averigua si es mayor o menor, para avanzar o retroceder según sea el caso. Los ultrasónicos laterales permiten el giro hacia la derecha o izquierda y paran cuando un objeto se encuentra a menos de 15 cm.

En la **Figura 29-2** se visualiza el diagrama de flujo del sistema de control de movimientos.

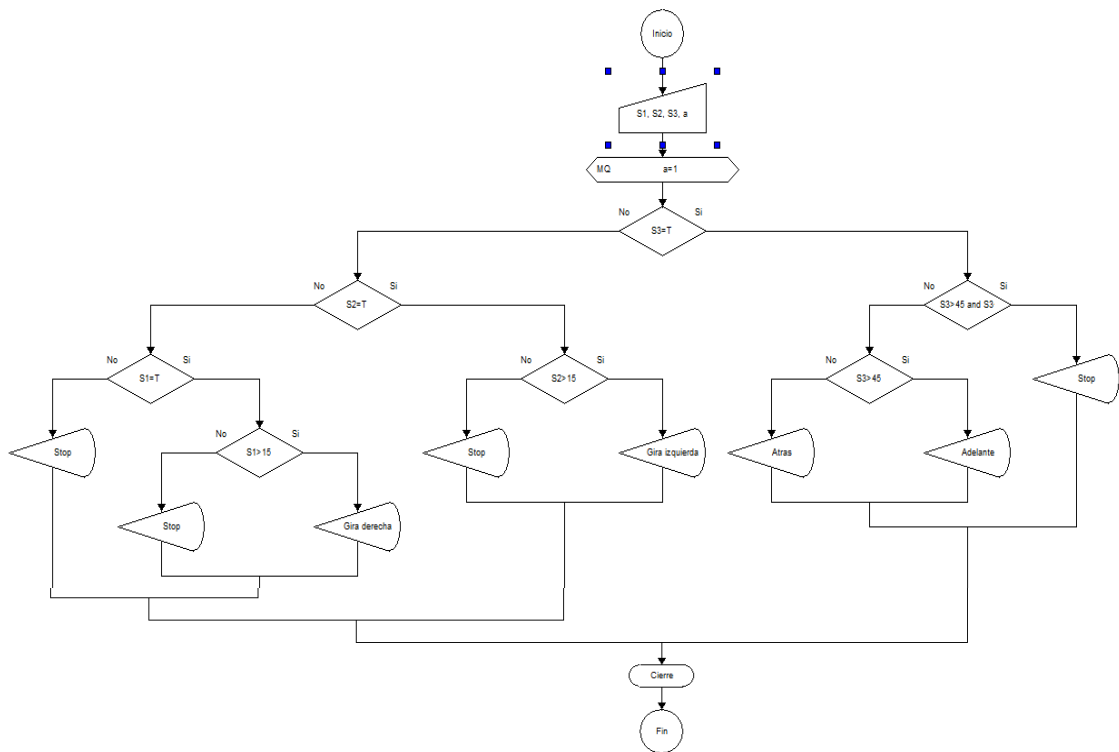


Figura 29-2: Diagrama de flujo para el control de movimientos

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El software arduino contiene las librerías *Sabertooth* y *Ultrasonic*. La primera permite conectarse con el driver, este dispositivo invierte la polarizar de la fuente de alimentación para que el motor DC gire. La segunda librería linealiza y lee los sensores HC-SR04.

La **Figura 30-2** muestra las líneas de comando para llamar a las librerías del driver y sensores.

```
#include <SabertoothSimplified.h>
#include <Sabertooth.h> // libreria del driver de motor
#include <Ultrasonic.h> //libreria del ultrasonico

Ultrasonic ssup(10,13); // (Trig PIN 10,Echo PIN 13) sensor frente superior
Ultrasonic sfre(6,7); // (Trig PIN 6,Echo PIN 7) sensor frente inferior
Ultrasonic sder(3,2); // (Trig PIN 3,Echo PIN 2) sensor derecho
Ultrasonic sizq(5,4); // (Trig PIN 5,Echo PIN 4) sensor izquierd
Ultrasonic sfreder(9,8); //(Trig PIN 9,Echo PIN 8) sensor frente-derecho
Ultrasonic sfreizq(11,12); // (Trig PIN 11,Echo PIN 12) sensor frente-izquierdo
SabertoothSimplified ST; // We'll name the Sabertooth object ST.
int s7 = A5;

void setup()
{
  SabertoothTXPinSerial.begin(9600); // This is the baud rate you chose with the DIP
  Serial.begin(9600);
}
```

Figura 30-2: Programa para control de movimientos

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.6 Integración de hardware y software

La electrónica es la ciencia aplicada de la tecnología moderna. A través de la circuitería eléctrica se integra hardware, software y control de servomecanismos. Los elementos electrónicos establecen interconexión.

Todo dispositivo electrónico debe estar acoplado al espacio designado en la estructura del prototipo, avalando la autonomía e interacción con el infante.

La **Figura 31-2** indica el diagrama de conexión de la Raspberry con la pantalla touch LCD, la cámara, el parlante y la fuente de alimentación proveniente del regulador de tensión.

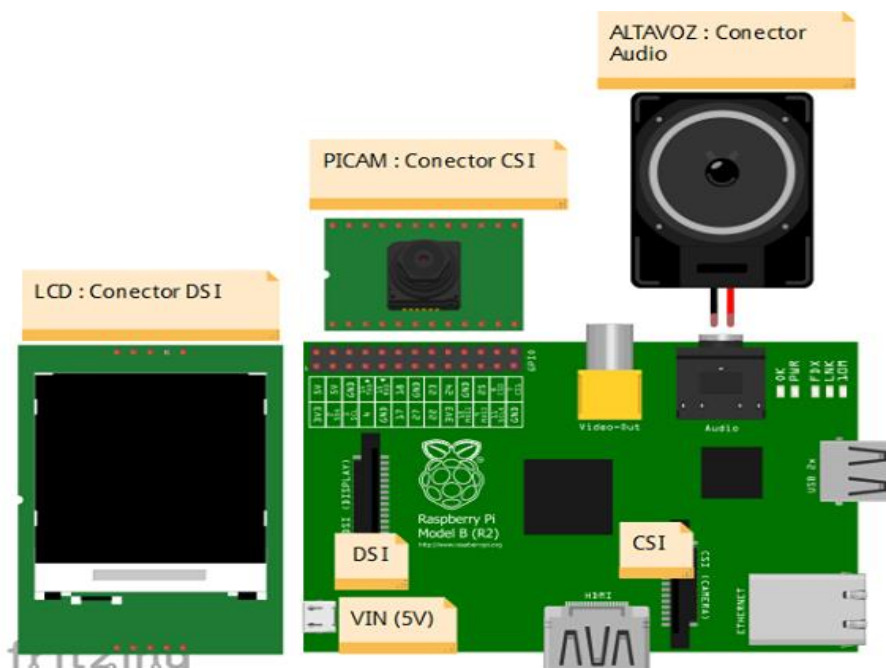


Figura 31-2: Diagrama de conexión en la Raspberry Pi

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Tres swiches dotan de independencia a cada proceso. Uno energiza la raspberry, conectado a al regulador. El otro activa los movimientos y el tercero enciende los led del semblante e infrarrojos de la cámara.

El diagrama del regulador de tensión DC se observa en la **Figura 32-2**. Este circuito se comporta como una fuente reductora, el voltaje de entrada oscila desde 5V hasta 35V. Manteniendo niveles de tensión contante a la salida, dicho valor es regulable. Opera a muy altas frecuencias, dentro del rango de los KHz.

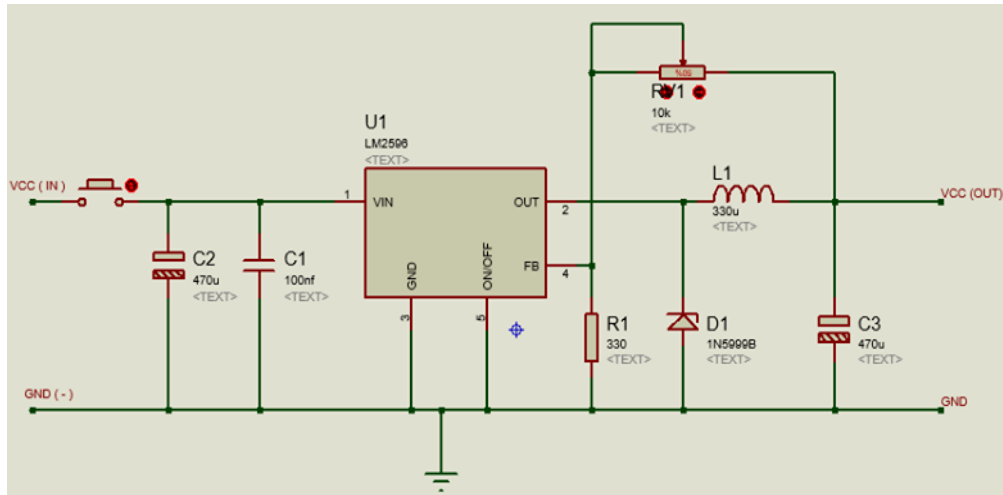


Figura 32-1: Circuito regulador de tensión DC

Realizado por: Salinas Lissette, Chamorro Edison, 2017

La **Figura 33-2** esquematiza la conexión del sistema para el control de movimientos. Los sensores emiten señales al microprocesador, este a su vez se conecta por los pines TX, VCC y GND al driver sabertooth, para coordinar el desplazamiento y proteger a los motores de sobre cargas.

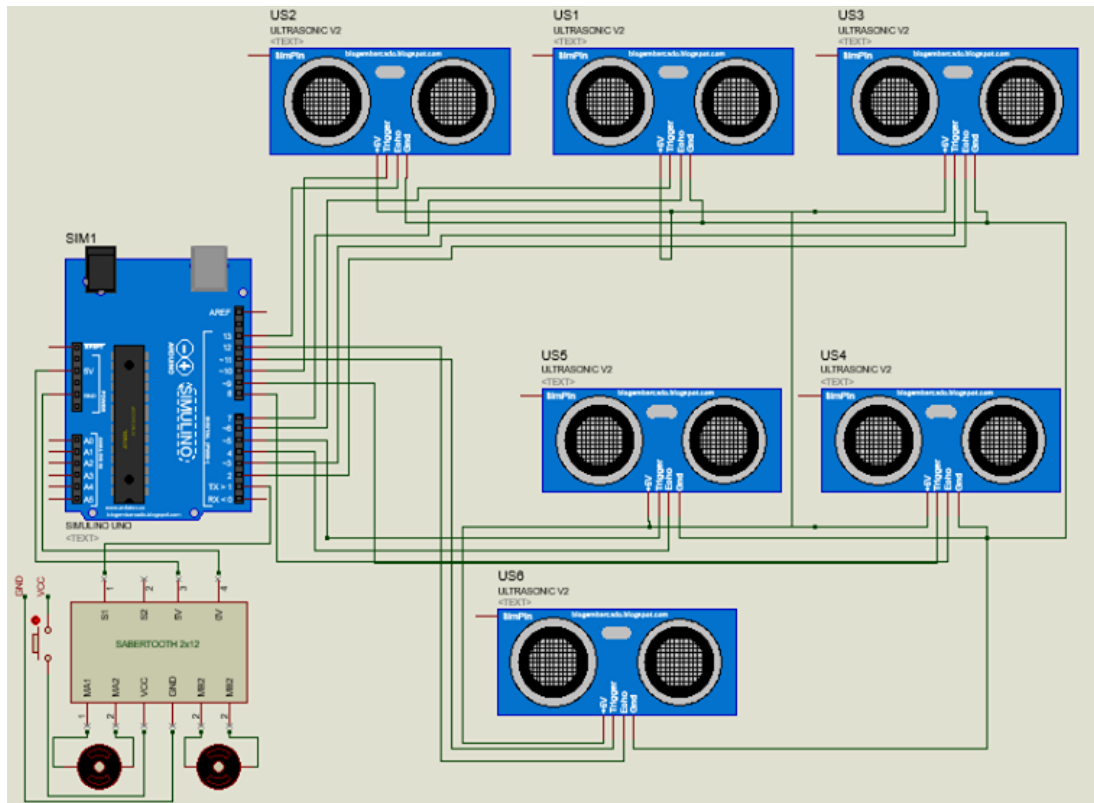


Figura 33-2: Diagrama de control de movimientos

Realizado por: Salinas Lissette, Chamorro Edison, 2017

2.7 Interacción lúdica y cognitiva

El juguete inteligente fue bautizado como Pushaq, término proveniente de vocablo Kichwa cuyo significado es guía, dirigente o amigo, debido a la tarea que realiza. El prototipo interactúa de forma lúdica y cognitiva con niños en etapa pre-operacional a través de una interfaz gráfica.

La GUI pone a disposición del pequeño cinco alternativas para divertirse y aprender, cada una de ellas se elige presionando sobre el ícono correspondiente. Pushaq enseña los colores y figuras, narra cuentos, reproduce canciones, juega, guarda fotografías de los mejores momentos y lo más novedoso es su sistema para reconocimiento facial.

La **Figura 34-2** muestra la interfaz gráfica de Pushaq para la interacción con los niños. Al seleccionar la opción *Aprendiendo con Pushaq*, se captura una imagen para el reconocimiento facial, seguidamente le enseñará la forma y color que tenga cada uno de los objetos que presencie el robot.

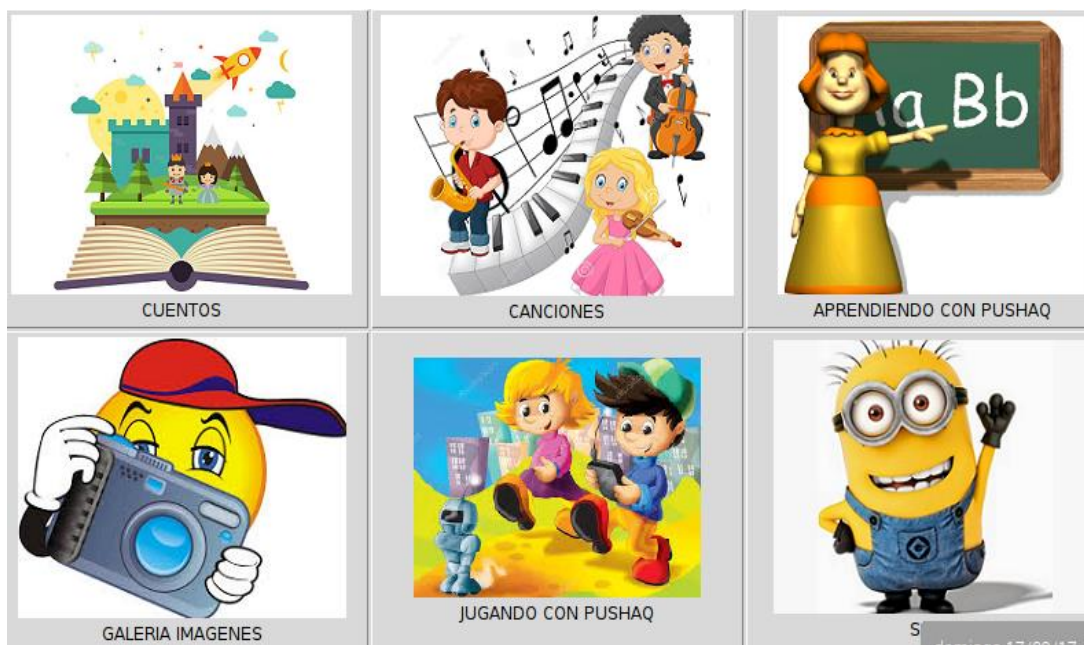


Figura 34-2: Interfaz gráfica de Pushaq

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Pushaq a más de ser un juguete educativo pedagógico cumple una función importante dentro de la educación especial, al ser un recurso moderno genera motivación intelectual. Olga Ballesteros en su tesis doctoral propone modificar los mecanismos de enseñanza inicial, aumentando el uso de recursos lúdicos. (Ballesteros, 2011, p.21)

2.8 Funcionamiento del juguete inteligente

Pushaq dispone de dos botones de encendido, el uno energiza la raspberry y el otro activa los motores. La ejecución inicializa con un video de presentación, luego aparece la interfaz gráfica. Al presionar sobre los íconos se ejecuta una actividad diferente, cada opción de la GUI se detallada en la **Tabla 2-8**.

Tabla 8-2: Descripción de la GUI

ÍCONO	NOMBRE	DESCRIPCIÓN
	Cuentos	Reproduce cuentos infantiles pre grabados
	Canciones	Reproduce canciones infantiles pre grabadas
	Aprendiendo con Pushaq	Identifica el rostro para desbloquear el reconocimiento de figuras y colores de los objetos que tome el niño, a la vez Pushaq reproducirá un audio con los descriptores a fin que el niño aprenda. Si a la tercera vez no se identifica el rostro volverá a la pantalla principal.
	Galería	Almacena y visualiza fotografías
	Jugando con Pushaq	El juego decorando la casa de Pushaq, consisten en ir colocando las figuras geométricas sobre la estructura de casa, el robot asistirá al niño, diciéndolo el nombre de la pieza que debe colocar, en su pechito mostrará cómo debe hacerlo.
	Salir	Al presionar esta opción la raspberry se apagará dando por terminado el proceso.

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Todos los videos y mensajes son pregrabados. El robot tiene la voz infantil, el audio se grabó y modificó en la aplicación móvil *Change Voice*. La entonación blanca es motivo para despertar la curiosidad del niño y generar un vínculo estrecho.

El software puede ser modificado por un técnico, ya sea para añadir rostros a la base de datos, entrenar la red neuronal, cambiar los rangos de colores o hallar los mismos. Conectando la placa Pi a un ordenador que tenga instalado VNC Server, por el puerto 2 con la dirección 169.254.46.1.31:2 y contraseña “raspberrry”. Obteniendo conexión remota al escritorio.

El escritorio de la Raspberry alberga la carpeta GUI, al correr el archivo *GUI.py* se desplegará una interfaz como el de la **Figura 35-2**.

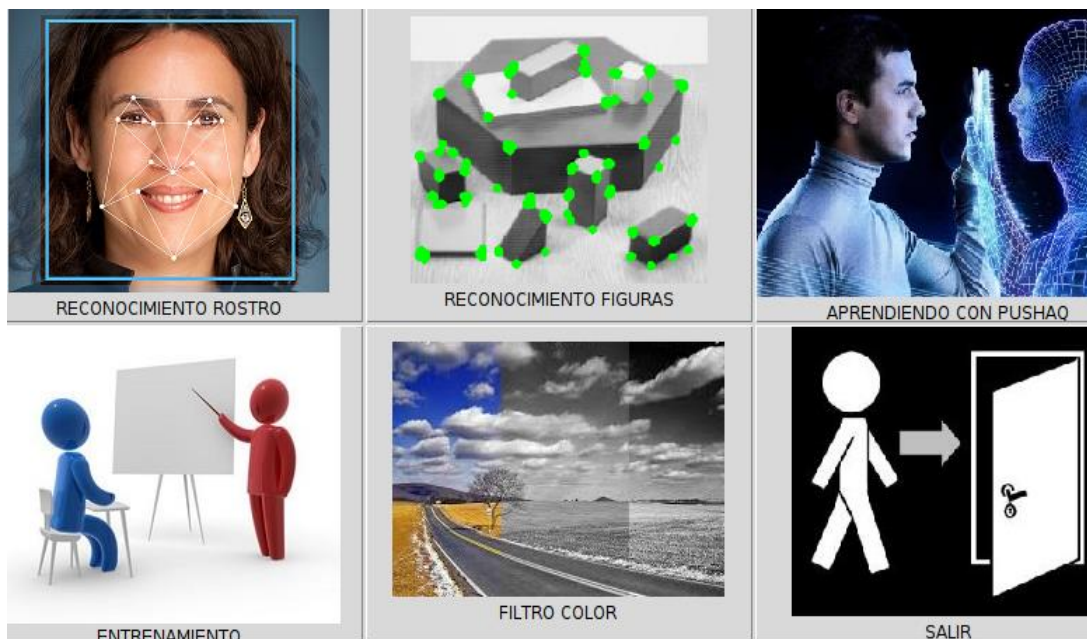


Figura 35-2: Interfaz gráfica para el Técnico

Realizado por: Salinas Lissette, Chamorro Edison, 2017

En los anexos F y G se encuentra el manual de usuario y del técnico de PUSAHQ el juguete inteligente, detallado más a fondo su funcionamiento.

CAPÍTULO III

3. MARCO DE ANÁLISIS DE RESULTADOS

En este capítulo se analizaron los resultados obtenidos de la implementación del juguete inteligente, para lo cual se realizaron pruebas de funcionamiento y evaluación de parámetros de los sistemas que integran el robot. Así también el aporte como recurso psicopedagógico para el desarrollo intelectual, sensorial, emocional y social de niños de 2 a 7 años.

3.1 Pruebas de funcionalidad

3.1.1 Evaluación del sistema de reconocimiento facial

La luminosidad y el tipo de pose, son dos parámetros que afectan el reconocimiento del rostro. El umbral predictivo del niño es variante, el uso de accesorios, cambio de nivel de luz y expresiones faciales modifican este valor. Las imágenes deben ser captadas en diferentes ambientes y poses, para entrenar la red neuronal.

En la **Tabla 1-3** y **Tabla 2-3** se aprecian los valores del umbral predictivo obtenidos de las pruebas realizadas a los dos participantes. Leandro y Samantha de 5 y 3 años de edad respectivamente.

Tabla 1-3: Valores de umbral predictivo del rostro de Leandro

PRUEBAS LEANDRO					
Nivel de luz	Tipo de pose				μ_{luz}
	Pose 1	Pose 2	Pose 3	Pose 4	
Alto	969,48	1049,14	1046,13	1343,21	1101,99
Medio	671,45	679,37	535,9	781,07	666,95
Bajo	361,46	654,89	699,18	531,92	561,86
μ_{pose}	667,46	794,47	760,40	885,40	776,93

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Las poses de Leandro fueron: sonriente, serio, despeinado y con lentes de sol, a medida que incrementa la intensidad luminosa, crece el valor de umbral con el cual identifica. Del mismo modo al bajar el nivel de luz decrece el umbral.

Tabla 2-3: Valores del umbral predictivo del rostro de Samantha

PRUEBAS SAMANTHA					
Nivel de luz	Tipo de pose				μ_{luz}
	Pose 1	Pose 2	Pose 3	Pose 4	
Alto	1250,35	1116,55	1084,61	1800,2	1312,93
Medio	880,83	574,03	512,57	664,33	657,94
Bajo	361,11	288,1	676,67	744,91	517,70
μ_{pose}	830,76	659,56	757,95	1069,81	829,52

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Las poses de Samantha fueron sonriente, cabello recogido, cabello suelto y con lentes de sol. Al igual que el caso anterior la conclusión es la misma. Para la niña el umbral resultó ser más alto, ya que el tipo de peinado modifica las facciones.

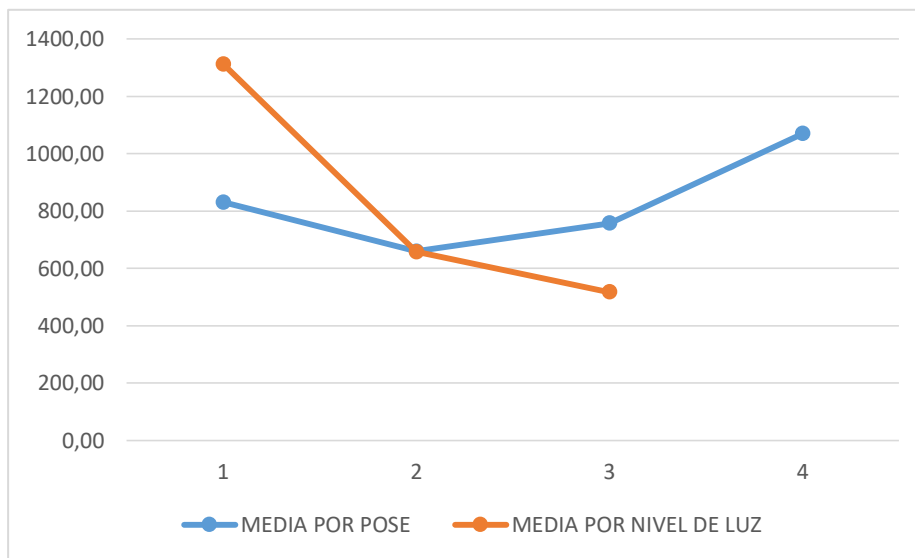


Gráfico 1-3: Influencia de luminosidad y pose en el reconocimiento facial

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Para que la detección sea satisfactoria en todo momento, se proyectó la media de los valores predictivos, dicho resultado corresponde al umbral predictivo. El **Gráfico 1-3** muestra la

variación del umbral predictivo respecto a la influencia de la luz y de la pose, claro está el uso de accesorios y la intensidad lumínica afectan notablemente al sistema de reconocimiento Facial.

A fin de reducir un pequeño porcentaje el efecto producido por los constantes cambios de nivel de luz, se acopló un juego de led infrarrojos al módulo de cámara de la Raspberry. Las imágenes de la **Figura 1-3** se capturaron en la noche con poca presencia de luz, las dos primeras fueron tomadas sin el uso de led infrarrojos y las consecuentes con el juego de led. Es evidente como las fotografías 216 y 217 se tornan más claras.

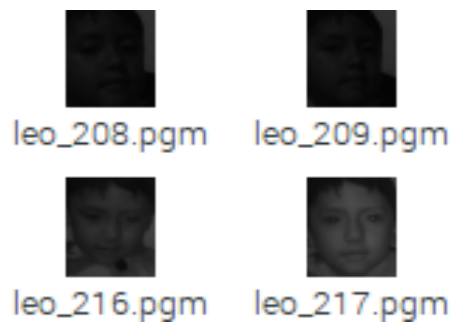


Figura 1-3: Fotografías captadas con picam

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Modificado el umbral perceptrónico (positive_threshold) y usando la cámara infrarrojo se evaluó la efectividad de la red neuronal para el reconocimiento facial, implementado en el juguete inteligente. Los resultados se evidencian en la **Tabla 3-3**, con un porcentaje de efectividad de 95.7%. Intrusos hace referencia a individuos no registrados en la base de datos, las no personas puede ser cualquier fotografía que no presente cara.

Tabla 3-3: Resultado de efectividad de la red neuronal

VARIABLES DE PRUEBA	INDIVIDUOS			
	Leandro	Samantha	Intrusos	No personas
Nº de imágenes	15	15	15	15
Tasa de detección	12	13	8	0
Tasa de falsos positivos	0	1	1	0
Tasa de falsos negativos	12	12	7	0
Tiempo de detección	10 ms	11 ms	11 ms	10 ms
Nº imagen rechazadas	3	2	7	15
% de detección	99	92,67	91	100
% EFECTIVIDAD	95,7			

Realizado por: Salinas Lissette, Chamorro Edison, 2017

La tasa de detección hace referencia únicamente a las capturas con caras, siendo el número de imágenes rechazadas las que no presentaron rostro humano. Esto se debe a varios factores, el más común es cuando los niños se mueven y el lente no enfoca correctamente, también cuando hay más de una cara o el infante está de perfil. Por no tratarse de un error de reconocimiento más bien de diseño se le ha asignado el 10% como fallo.

La primera vez identificó a Samantha como intruso, ante la existencia de un porcentaje de error de 4.7% se admiten tres intentos para ser reconocidos, si todos resultan negados volverá a la interfaz gráfica.

3.1.2 Pruebas del sistema de reconocimiento de colores y figuras 2D

La detección de colores y figuras geométricas de cuerpos 2D se desbloquea con el rostro del niño. La forma más efectiva para la identificación fue por tracking de colores, es decir, detecta únicamente los tonos de colores establecidos, desechando el resto para hallar la forma.

El tiempo de respuesta fue sumamente rápido en el orden de los milisegundos. Por lo que frecuentemente se encuentra analizando todos los cuerpos que el lente de la cámara alcance a detectar. Funciona solo en ambientes controlados.

Ambientes controlados se refiere a espacios específicos, debido a la influencia de la luz u otros factores que modifican ciertas condiciones en los rangos de colores, por tal razón se añadió un programa que calcula los tonos altos y bajos en formato RGB.

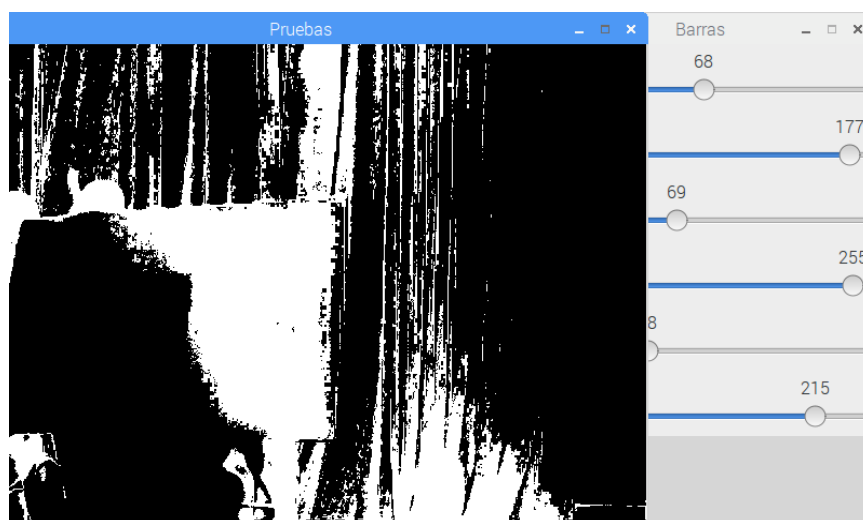


Figura 2-3: Filtro en tiempo real para hallar el rango de colores

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Filter real time dispone de 6 barras, para mover hasta que el cuerpo enfocado aparezca completamente en la pantalla en color blanco y lo que le rodea debe tornarse negro. Los números sobre las barras son los valores de las bandas espectrales.

En el ejemplo de la **Figura 2-3**, se determinó el rango del color azul en una habitación clara, los resultados son los siguientes: para alto [68, 69, 8] y tono bajo [177, 255, 215].

3.1.3 Pruebas del sistema de control de movimientos

3.1.3.1 Sistema sensorial

Con el propósito de mejorar los giros y trayectorias, se calibraron los sensores acorde a la posición física en el prototipo. Tomando como referencia un valor estándar en la distancia de separación de 45 a 50 cm. La **Figura 3-3** muestra los resultados obtenidos en las lecturas de los sensores ultrasónicos.

S1 :51 cm	S2 :51 cm	S3 :51 cm	S4 :60 cm	S5 :51 cm	S6 :29
S1 :51 cm	S2 :51 cm	S3 :51 cm	S4 :70 cm	S5 :51 cm	S6 :44
S1 :51 cm	S2 :51 cm	S3 :51 cm	S4 :58 cm	S5 :51 cm	S6 :29
S1 :51 cm	S2 :51 cm	S3 :51 cm	S4 :65 cm	S5 :51 cm	S6 :40
S1 :51 cm	S2 :51 cm	S3 :44 cm	S4 :51 cm	S5 :42 cm	S6 :28
S1 :51 cm	S2 :51 cm	S3 :41 cm	S4 :62 cm	S5 :51 cm	S6 :37
S1 :51 cm	S2 :34 cm	S3 :36 cm	S4 :51 cm	S5 :51 cm	S6 :27
S1 :51 cm	S2 :51 cm	S3 :34 cm	S4 :61 cm	S5 :51 cm	S6 :35
S1 :51 cm	S2 :51 cm	S3 :36 cm	S4 :51 cm	S5 :51 cm	S6 :27
S1 :51 cm	S2 :51 cm	S3 :38 cm	S4 :60 cm	S5 :51 cm	S6 :31
S1 :51 cm	S2 :51 cm	S3 :43 cm	S4 :68 cm	S5 :51 cm	S6 :28
S1 :51 cm	S2 :31 cm	S3 :44 cm	S4 :51 cm	S5 :51 cm	S6 :5
S1 :51 cm	S2 :31 cm	S3 :47 cm	S4 :51 cm	S5 :51 cm	S6 :25
S1 :51 cm	S2 :51 cm	S3 :49 cm	S4 :58 cm	S5 :51 cm	S6 :41
S1 :51 cm	S2 :51 cm	S3 :50 cm	S4 :66 cm	S5 :51 cm	S6 :29
S1 :51 cm	S2 :30 cm	S3 :49 cm	S4 :58 cm	S5 :51 cm	S6 :37
S1 :51 cm	S2 :30 cm	S3 :45 cm	S4 :70 cm	S5 :51 cm	S6 :30
S1 :51 cm	S2 :51 cm	S3 :41 cm	S4 :60 cm	S5 :51 cm	S6 :34
S1 :51 cm	S2 :51 cm	S3 :37 cm	S4 :51 cm	S5 :51 cm	S6 :27
S1 :51 cm	S2 :51 cm	S3 :34 cm	S4 :57 cm	S5 :51 cm	S6 :31
S1 :51 cm	S2 :51 cm	S3 :31 cm	S4 :51 cm	S5 :51 cm	S6 :49
S1 :51 cm	S2 :51 cm	S3 :28 cm	S4 :59 cm	S5 :51 cm	S6 :30
S1 :51 cm	S2 :51 cm	S3 :25 cm	S4 :54 cm	S5 :51 cm	S6 :48
S1 :51 cm	S2 :51 cm	S3 :22 cm	S4 :65 cm	S5 :51 cm	S6 :31
S1 :51 cm	S2 :51 cm	S3 :21 cm	S4 :57 cm	S5 :51 cm	S6 :28
S1 :51 cm	S2 :51 cm	S3 :19 cm	S4 :66 cm	S5 :51 cm	S6 :28
S1 :51 cm	S2 :51 cm	S3 :17 cm	S4 :66 cm	S5 :51 cm	S6 :45
S1 :51 cm	S2 :51 cm	S3 :15 cm	S4 :70 cm	S5 :51 cm	S6 :26
S1 :51 cm	S2 :51 cm	S3 :14 cm	S4 :51 cm	S5 :51 cm	S6 :42
S1 :51 cm	S2 :51 cm	S3 :13 cm	S4 :51 cm	S5 :51 cm	S6 :28
S1 :51 cm	S2 :51 cm	S3 :11 cm	S4 :56 cm	S5 :51 cm	S6 :40
S1 :51 cm	S2 :51 cm	S3 :11 cm	S4 :69 cm	S5 :51 cm	S6 :28
S1 :51 cm	S2 :51 cm	S3 :9 cm	S4 :62 cm	S5 :51 cm	S6 :26
S1 :51 cm	S2 :51 cm	S3 :8 cm	S4 :59 cm	S5 :51 cm	S6 :27

Figura 3-3: Distancias sensadas por HC-SR04

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Los datos fueron tomados desde un computador y se vinculó a Excel, para determinar el comportamiento de los sensores una vez que el juguete inteligente esté encendido e interactuando con niños, los resultados obtenidos se aprecian en el **Gráfico 2-3**.

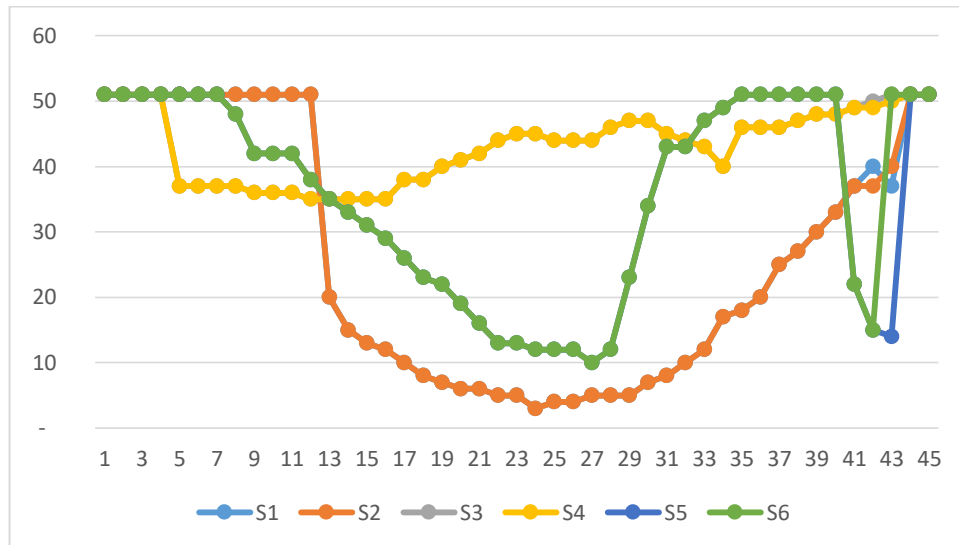


Gráfico 2-3: Comportamiento de los sensores HC-SR04

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Donde:

- S1 es el sensor superior delantero
- S2 es el sensor inferior delantero
- S3 es el sensor del costado derecho
- S4 es el sensor del costado izquierdo
- S5 es el sensor de la esquina delantera derecha
- S6 es el sensor de la esquina delantera izquierda

3.1.3.2 Control de giros

Los giros en el sistema diferencial de locomoción, son controlados por la velocidad de los motores desde el arduino. El driver es el encargado de invertir los giros para ejecutar los movimientos requeridos.

En la siguiente tabla se indica los valores que tomaron los motores. El motor de la izquierda o 1 y el motor derecho o 2. El prototipo avanza en sentido horario y retrocede en sentido anti horario, es decir, cuando tenga signo negativo.

Tabla 4-3: Control de velocidad de motores para giros

GIRO	MOTOR 1	MOTOR 2
Paro	0	0
Avanza	100	100
Retrocede	-100	-100
Izquierdo	125	-125
Derecho	-125	125

Realizado por: Salinas Lissette, Chamorro Edison, 2017

3.2 Pruebas de interacción con niños

Para evaluar el desempeño del juguete inteligente se realizaron pruebas con niños de la Escuela de Educación Básica “Archipiélago de Galápagos”, ubicada en el barrio Bellavista, Parroquia de Santa Rosa de la ciudad de Ambato. La institución cuenta con 192 alumnos en la jornada matutina, de primer a sexto a grado.

Durante la formación se realizó una pequeña exposición y demostración del trabajo de titulación. Pushaq se presentó y recorrió las instalaciones del plantel, cautivó a niños, profesores y padres de familia.

La **Figura 4-3** fue tomada mientras Pushaq se presentaba, los niños se aglomeraron por acercarse y acariciarlo.



Figura 4-3: Presentación de Pushaq en la escuela Archipiélago de Galápagos

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Pushaq visitó las aulas, donde los niños interactuaron con el robot. Los alumnos de cuarto grado en adelante, sintieron curiosidad por conocer como estaba construido, preguntaban constantemente su funcionamiento. Les motivó para aprender electrónica y robótica.



Figura 5-3: Pushaq narrando cuentos infantiles

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Luego de interactuar, se solicitó a los alumnos levantar la mano a quienes les haya gustado Pushaq y cual era razón. Las cifras se encuentran en la **Tabla 5-3** y se evidencia en la **Figura 6-3**.



Figura 6-3: Niños encantos con Pusahq levantando la mano

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Tabla 5-1: Resultados de aceptación de Pushaq

Juicio Valorativo	Resultados
Les gusto	189
No les gusto	3
Porcentaje de aceptación	98.44%

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Al 98.44% de alumnos les fascinó Pusahq, las respuestas más comunes de la razón por la cual les agradaba eran porque caminaba, tomaba fotos, cantaba, era inteligente, les enseñaba colores y figuras. Los niños que no levantaron la mano fueron tres, el uno argumento que no le gustaba porque no era real y los otros dos coincidieron al decir que era un juguete para niños pequeños.

Los niños sugirieron que los ojos sean iguales, añadirle brazos, orejas, pelo, piernas, expresiones faciales, entre otras cosas. En sí ellos quieren un robot humanizado, con características similares a los humanos.

Pese a su alto nivel de aceptación, unos niños estaban más animados y emocionados que otros. Para determinar el grado de interés se analizó el comportamiento de cada uno de los infantes cuando interactuaban con el robot, considerando la edad.

Tabla 6-3: Nivel de interés por edades, datos numéricos

NIVEL INTERÉS EDAD	ALTO	MEDIO	BAJO	TOTAL
[2, 3] años	1	2	0	3
[3, 4] años	3	1	0	4
[4, 5] años	24	3	2	29
[5, 6] años	24	5	4	33
[6, 7] años	17	9	6	32
	69	20	12	101

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Tabla 7-3: Nivel de interés por edades, datos porcentuales

NIVEL INTERÉS EDAD	ALTO	MEDIO	BAJO	TOTAL
[2, 3] años	33.33 %	66.67 %	0.00 %	100 %
[3, 4] años	75.00 %	25.00 %	0.00 %	100 %
[4, 5] años	82.76 %	10.34 %	6.90 %	100 %
[5, 6] años	72.73 %	15.15 %	12.12 %	100 %
[6, 7] años	53.13 %	28.13 %	18.75 %	100 %
	63.39 %	29.06 %	7.55 %	100 %

Realizado por: Salinas Lissette, Chamorro Edison, 2017

El **Gráfico 3-3** esquematiza los resultados arrojados de la interacción con niños, es notable como el grado de interés aumenta y disminuye acorde a la edad.

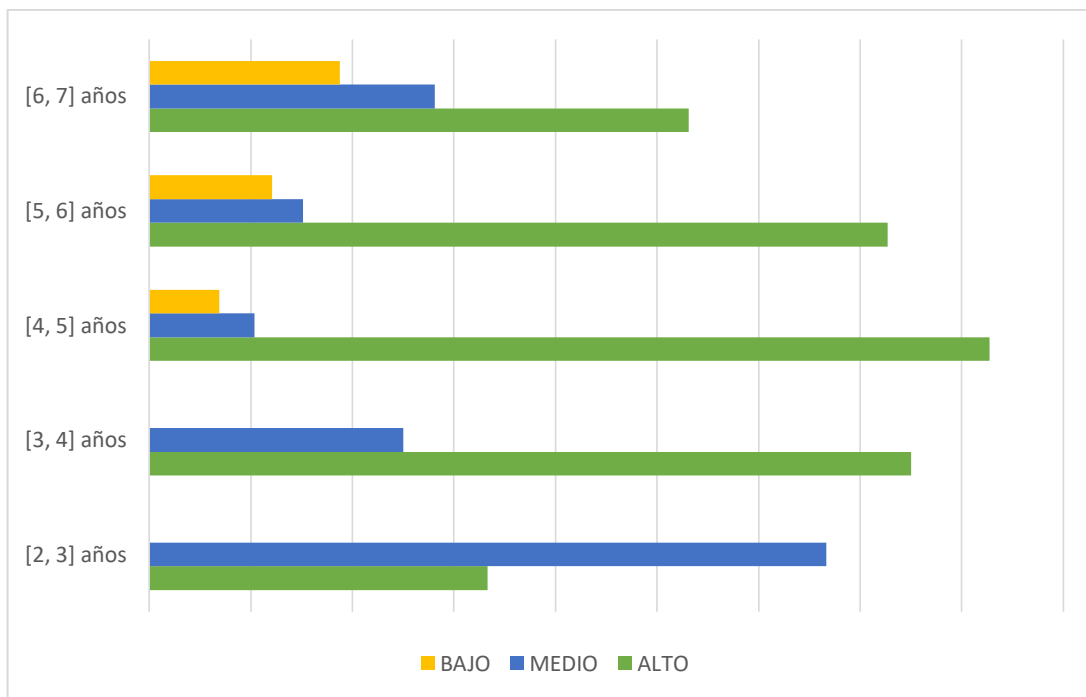


Gráfico 3-3: Niveles de interés, por edades.

Realizado por: Salinas Lissette, Chamorro Edison, 2017

En los primeros dos años de vida los niños aprenden por medio de imágenes y símbolos. Se interesan más por el juego, los cuentos y canciones pasan a segundo plano. A medida que crecen se vuelven más curioso y su interés aumenta, alcanzando al máximo el nivel entre los 4 y 5 años. A partir del sexto año los infantes desean descubrir nuevas cosas por lo que el interés por el robot empieza a descender.

El **Gráfico 4-3** muestra los porcentajes generales del nivel de interés de los niños. Apenas el 7.55% no empatizó con PUSAHQ, el 29.06% presentaron una conducta normal y el 63.39% se impresionaron y esperaban con ansias su turno para interactuar con el juguete inteligente.

De forma global el nivel de interés alto predominada en toda la etapa pre operacional, según especialistas esto se debe a que el juego desempeña un rol importante en el proceso de desarrollo infantil. Es inhabitual ver casos en donde los juguetes, actividades lúdicas u otros recursos didácticos no generen interés alguno en un impúber

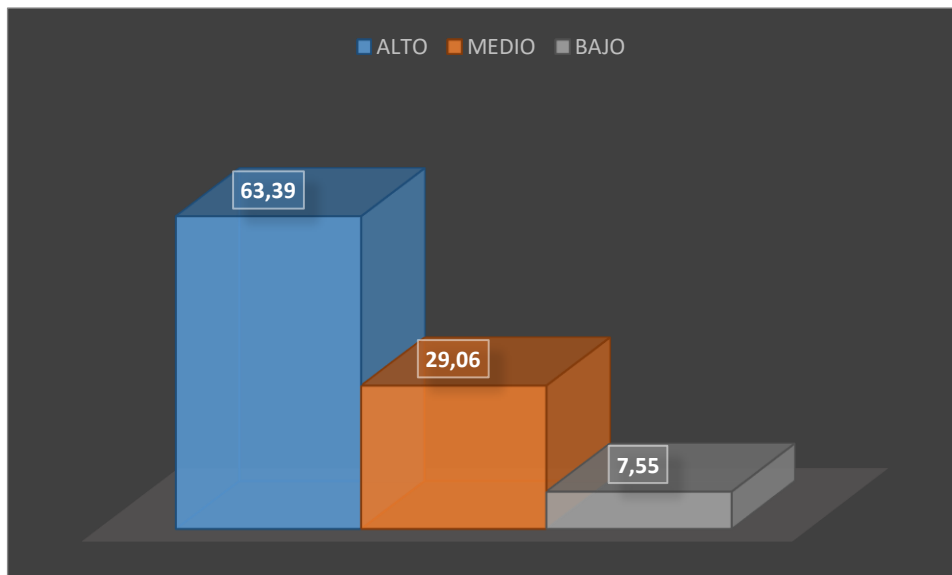


Gráfico 4-3: Nivel de interés global

Realizado por: Salinas Lissette, Chamorro Edison, 2017

También los docentes y personal del DCE (Departamento de Consejería Estudiantil) de la institución, dieron su juicio valorativo. Pusaq le simpatizó y recalcaron su labor como recurso didáctico dentro del proceso de aprendizaje y estimulación de estudiantes.

Tabla 8-3: Juicio valorativo por parte de docentes y personal de DCE.

Juicio Valorativo	Medición cuantitativa	Resultados
Excelente	10	4
Muy bueno	9	3
Bueno	8 - 7	1
Regular	6 - 5	0
Malo	0 - 4	0
Promedio		Muy bueno

Realizado por: Salinas Lissette, Chamorro Edison, 2017

Según los resultados de la **Tabla 8-3** Pusaq es considerado un juguete inteligente innovador muy bueno para la enseñanza a niños de 2 a 7 años. El promedio sobre 10 es de 9,31 equivalente al 93.1%.

Unos de los resultados más satisfactorios fue el aporte dentro de la educación especial, desempeñando el papel de robot psicopedagógico. Esto se constató al estimular el lenguaje del pequeño Adrián Carrasco. Según su maestra de segundo año de educación básica, el niño presenta necesidades educativas especiales y trastorno de lenguaje nivel 3.

Adrián usualmente no participa en clases, sin embargo fue el primero en levantarse para saludar a Pushaq, a la hora de tomarle fotos el robot le pregunto el nombre y él lo dijo claramente. En la **Figura 7-3** se indica la interacción de Adrián y Pushaq.



Figura 7-3: Adrián Carrasco saludando a Pushaq

Realizado por: Salinas Lissette, Chamorro Edison, 2017

CONCLUSIONES

- ✓ El simple hecho de que el juguete inteligente hable y se mueva es motivo suficiente para que niños de 2 a 7 años interactúen de forma lúdica y cognitiva. En el pecho del robot aparece una interfaz gráfica con varias opciones disponibles como: cuentos, canciones, fotos y juegos, para una adecuada percepción de colores y figuras geométricas.
- ✓ La soft computing es la hibridación de dos o más técnicas de inteligencia artificial, cada una de ellas aporta de diferente manera para dar solución a problemas complejos carentes de modelos matemáticos. Las redes neuronales junto con los sistemas evolutivos son los encargados del aprendizaje, se usa a menudo para el reconocimiento de patrones, por otro lado la lógica difusa se ocupa de la impresión y los métodos probabilísticos tratan la incertidumbre.
- ✓ La selección de la técnica de soft computing para el procesamiento inteligente de imágenes, se realiza en base a la compatibilidad con el lenguaje de programación y los recursos hardware.
- ✓ El sistema de reconocimiento facial (Face ID) implementado es una adaptación del modelo propuesto por Viola-Jones, haciendo uso de la imagen integral, clasificadores cascada y un algoritmo Adaboost para entrenar una red neuronal, reduciendo el costo computacional. Sin duda es uno de los modelos biométricos más eficientes para la identificación de rostros alcanzando el 95,7% de efectividad con una baja tasa de error.
- ✓ El juguete inteligente construido tiene forma de un muñeco extraterrestre, de un grupo de 192 niños al 98.44% les fascinó. El diseño del prototipo se realizó en el programa de modelamiento asistido por computador SolidWorks para seguidamente imprimir en 3D usando material PLA y se pintó con aerosol biodegradable.
- ✓ El invaluable aporte de la electrónica dentro de la robótica permite integrar el software al hardware y sistema de locomoción. Tal es el caso de los sensores encargados de esquivar obstáculos, planificar la trayectoria del juguete y posicionarlo a 50 cm del niño, distancia óptima para la captura y procesamiento inteligente de imágenes.
- ✓ El juguete inteligente implementado es innovador y bueno para la estimulación y desarrollo de habilidades, destrezas y competencias de niños en etapa pre-operacional. Tiene un valioso aporte dentro de la educación especial, como un robot psicopedagógico.

RECOMENDACIONES

- ✓ Es conveniente dar una breve inducción del funcionamiento y manipulación del juguete inteligente, tanto al niño como a la persona adulta responsable. A fin de evitar un mal uso que impida la adecuada interacción o aun peor ponga en riesgo la integridad del infante y del robot. Ya que muchas veces es obviado la lectura del manual.
- ✓ Para evitar contratiempos en la ejecución de los procesos en caso de mejorar o expandir la aplicación del prototipo, se recomienda usar un procesador con mejores características, mayor velocidad y más cantidad de memoria RAM.
- ✓ La red neuronal para el reconocimiento facial debe ser entrenada múltiples veces, con gran cantidad de imágenes tomadas en distintos ambientes y con diferentes poses; para asegurar su efectividad. De ser necesario se debe cambiar el valor del umbral perceptrónico (positive_threshold) en el subprograma de configuración, en base al umbral predictivo del niño o niña.
- ✓ El mayor inconveniente de visión e inteligencia artificial es la luz, los constantes cambios afectan a la ejecución normal de los procesos. Por la motivo se recomienda implementar fuentes de iluminación para corregir los efectos provocados por la luminosidad, de esta manera no limitar el uso de dispositivos o aplicaciones a espacios controlados.
- ✓ Para futuros proyectos lúdicos es aconsejable solicitar ayuda o sugerencias de los usuarios finales, en este caso los niños ya que ellos van a orientar de mejor manera en el diseño y construcción de prototipos, generando un vínculo estrecho.
- ✓ Fomentar el uso de materiales biodegradables para solventar el latente problema de contaminación y enfermedades causado por el uso de sustancias tóxicas.
- ✓ Se recomienda a estudiantes e investigadores profundizar estudios sobre el tema, para mejorar el prototipo o diseñar nuevos dispositivos juguetrónicos para contribuir al levantamiento de las bases cognitivas.

BIBLIOGRAFÍA

Algoritmos genéticos. [En línea]. Temas genéticos. [Consulta: 15 julio 2017]. Disponible en: <http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/temageneticos.pdf>

ALZATE GÓMEZ, Juan Pablo; & LOTERO MARTÍNEZ, Carlos Andrés. *Inventario y pronóstico de demanda en ensamblaje de equipos mecánicos: motorreductor* [En línea], (Tesis) (Pregrado). Universidad EAFIT. Departamento de Ingeniería Mecánica. Medellín (Colombia). 2008. pp.82-84. [Consulta: 19 abril 2017]. Disponible en: <http://bit.ly/2xvmSjs>

ARDUINO-HOME. *Introduction Arduino*. [Blog]. [Consulta: 20 abril 2017]. Disponible en: <https://www.arduino.cc>

ARÉVALO, V.; et al. *La librería OpenCV, aplicación a la docencia e investigación* [En línea]. España: Universidad de Málaga, 2002. [Consulta: 21 abril 2017]. Disponible en: <http://mapir.isa.uma.es/varevalo/drafts/arevalo2004lva1.pdf>

BALLESTEROS, Olga. *La lúdica como estrategia didáctica para el desarrollo de competencias científicas*. [En línea], (Tesis) (Postgrado). Universidad Nacional de Colombia. Facultad de Ciencias. Bogotá (Colombia). 2011. pp. 20-22. [Consulta: 17 septiembre 2017]. Disponible en: <http://www.bdigital.unal.edu.co/6560/1/olgapatriaballesteros.2011.pdf>

BARGH, J. *Social Psychology and the unconscious*. Nueva York: Psychology Press, 2007.

BRAVO, Flor; & FORERO, Alejandro. *La robótica como un recurso para facilitar el aprendizaje y desarrollo de competencias generales*. [En línea]. España: Universidad de Salamanca, 2012, 3 (2). [Consulta: 06 abril 2017]. ISSN 1138-9737. Disponible en: <http://hdl.handle.net/10366/121799>

CARRASCO, Jesús; & MARTÍNEZ, José. “Reconocimiento de Patrones”. *Researchgate* [En línea], 2011 (México), 3(2), pp.5-8. [Consulta: 01 junio 2017]. Disponible en: https://www.researchgate.net/profile/Eduardo_Morales11/publication/236027639

CASTAÑEDA CHATE, Juan; & CAYAMBE YAMBAY, Diana. *Inteligencia Artificial aplicada a robot asistencial para la interacción con niños en edad temprana* [En línea], (Tesis) (Pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica,

Ingeniería Electrónica en Control y Redes Industriales. Riobamba (Ecuador). 2016. [Consulta: 14 abril 2017]. Disponible en: <http://bit.ly/2xxgQ2P>

EL COMERCIO. *Presentan un robot educativo hecho en el Ecuador.* 2015, p.5

CHIN-WEI Chang; et al. “Exploring the Possibility of Using Humanoid Robots as Instructional Tools for Teaching a Second Language in Primary School”. *Journal of Educational Technology & Society*. [En línea], 2010 (Japón), 13(2), pp. 13-24. [Consulta: 27 mayo de 2017]. Disponible en: <http://www.jstor.org/stable/jeductechsoci.13.2.13>

D’NEGRI Carlos; & DE VITO Eduardo. “Introducción al razonamiento aproximado: lógica difusa”. *Revista Argentina de Medicina Respiratoria* [En línea], 2006 (Argentina), 1(4), pp.126-136. [Consulta: 12 abril 2017]. Disponible en: <http://bit.ly/2jHJPuu>

DIEZ, R. P., et al. *Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva.* [En línea]. Universidad de Oviedo, 2001, pp. 10-26. [Consulta: 03 abril 2017]. Disponible en: <http://bit.ly/2xhHTvS>

FISHER PRICE. [En línea]. *Robot-Robi.* [Consulta: 15 marzo 2017]. Disponible en: http://www.fisher-price.com/es_ES/products/Robot-Robi

FU, K.; et al. Lee. *Robótica: Control, detección, visión e inteligencia.* Madrid- España: McGraw Hill, 1989.

GARCÍA SANTILLÁN, Iván Danilo. *Visión artificial y procesamiento digital de imágenes usando Matlab* [En línea]. Ibarra-Ecuador: Universidad Católica, 2008. [Consulta: 16 abril 2017]. ISBN 978-9942-01-790-1. Disponible en: <http://bit.ly/2xQu3UI>

GUEVARA, Martha Lucía; et al. “Detección de rostros en imágenes digitales usando clasificadores en cascada”. *Scientia et Technica*, 1,38 (2008), (Colombia) pp. 1.6

GÓMEZ GONZÁLES, Sergio. *El gran libro de SolidWorks.* 2ª ed. Barcelona-España: Marcombo, 2014, pp. 7-50.

GONZÁLEZ, Juan Ramón, *Herramientas de Soft Computing para la Comparación de Estructuras de Proteínas* [En línea] (Tesis) (Doctoral) Universidad de Granada, (Granada). 2008. [Consulta: 22 julio 2017]. Disponible en: <http://tdx.cat/handle/10803/16441?locale-attribute=en>

GONZÁLEZ, K.; et al. “Estudio sobre computadores de placa reducida Raspberry pi modelo B y Cubieboard2 en la Creación de Proyectos con Base Tecnológica de Gran Impacto Social”. *ENGI Revista electrónica de la Facultad de Ingeniería* [En línea], 2014, 3(1), pp. 6-10. [Consulta: 19 abril 2017]. ISSN 2256-5612. Disponible en: <http://bit.ly/2heXBUC>

HARRINGTON, William. *Learning Raspbian* [En línea]. Birmingham: Packt Publishing, 2015. [Consulta: 20 abril 2017]. Disponible en: <http://bit.ly/2xeg5e>

ILBAY LLANGARÍ, Luis Guido. *Evaluación de Algoritmos de Tracking 3D para la Simulación de una Brazo Robótico, mediante Kinect* [En línea], (Tesis) (Pregrado). Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Ingeniería Electrónica en Control y Redes Industriales. Riobamba (Ecuador). 2015. [Consulta: 14 abril 2017]. Disponible en: <http://bit.ly/2xxO9y8>

IZAURIETA, Fernando & SAAVEDRA, Carlos. “Redes neuronales artificiales”. *Departamento de Física, Universidad de Concepción Chile*: 2000, pp. 1-14

LOGITECH, HD Web C920 [Blog]. [Consulta: 15 abril 2017]. Disponible en: <http://www.logitech.com/es-roam/product/hd-pro-webcam-c920>

MINISTERIO DE EDUCACIÓN, *Currículo Educación General Inicial y Básica Preparatoria* [En línea], 2016. Ecuador. [Consulta: 12 julio 2017]. Disponible en: <https://educacion.gob.ec>

MONTERO RAÉZ, Arturo; & UR, Luis Alfonso. “Binary Classifiers versus AdaBoost for labeling of digital documents”. *Procesamiento del Lenguaje Natural*. [En línea], 2006, (España), 37(1), pp. 1-8. [Consulta: 17 julio 2017]. Disponible en: <http://bit.ly/2hzYcfX>

MORALES SÁNCHEZ, Alberto Alejandro. *Uso de características no lineales para identificar llantos de recién nacidos con un conjunto clasificador* [En línea]. (Tesis) (Pregrado). Universidad de las Américas Puebla, Departamento de Computación, Electrónica y Mecatrónica, Escuela de Ingeniería. México: 2015. pp. 21-31. [Consulta: 17 julio 2017]. Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lmt/morales_s_aa/capitulo3.pdf

MORENO DÍAS, Ana Belén. *Reconocimiento facial automático mediante técnicas de visión tridimensional.* (Tesis) (Doctoral). Universidad Politécnica de Madrid, Facultad de Informática, Licenciatura en Ciencias Físicas, España. 2004. pp. 9-30.

NILSSON, Nils J. *Inteligencia Artificial: una nueva síntesis.* Madrid-España: McGraw Hill, 2001.

OPENCV. *Face Recognition with OpenCv* [En línea]. OpenCV 2.4.13.3 documentation. [Consulta: 20 julio 2017]. Disponible en:
http://docs.opencv.org/2.4/modules/contrib/doc/face_recognition/face_recognition_tutorial.html

OTTADO, Guillermo. *Reconocimiento de caras: Eigenfaces y Fisherfaces.* [En línea]. 2010. Archivo 514. [Consulta: 04 septiembre 2017]. Disponible en: <http://bit.ly/2xfHf1J>

PAJARES, Gonzalo; et al. *Imágenes Digitales: Procesamiento práctico con Java.* Madrid-España: Alfaomega, 2003.

PAZ, Néstor. *Adaboost con aplicación a detección de caras mediante algoritmo de Viola-Jones* [En línea]. 2009, pp. 2-40. [Consulta: 18 julio 2017]. Disponible en:
https://eva.fing.edu.uy/file.php/514/ARCHIVO/2008/TrabajosFinales2008/NestorPaz2008_informe.pdf

PIAGET, Jean; & Inhelder, B. *A child's conception of space: Norton Edition, 1967; p. 178*

PÉREZ GÓMEZ, Ángel. *Educarse en la era digital.* Madrid-España: Morata, 2012

PLANELLAS LERMA, Joaquín. *“Implementación del algoritmo de detección facial de Viola-Jones”* [En línea], (Tesis) (Pregrado). Universidad Politécnica de Valencia, Ingeniería Informática. Valencia (España). 2009. pp. 8-30. [Consulta: 18 julio 2017]. Disponible en <https://riunet.upv.es/handle/10251/8774>

RAMA, Antonio; & TARRÉS, Francesc. *Un nuevo método para la detección de caras basado en integrales difusas.* Universidad Politécnica de Catalunya. Barcelona – España, 2007

RASPERRY. *Módulo cámara Raspberry Pi* [En línea]. [Consulta: 20 mayo 2017]. Disponible en: <https://www.raspberrypi.org/products/camera-module-v2/&prev=search>

RENTERÍA, Arantxa; & RIVAS, María. *Robótica industrial: fundamentos y aplicaciones.* España: McGraw Hill, 2000.

SERRANO, Antonia J.; et al. “Redes neuronales artificiales”. *Universidad de Valencia (Escuela Técnica Superior Ingeniería, Departamento Ingeniería Electrónica):* Valencia, España, 2009, pp. 13-16

SUPER ROBÓTICA, *Sensor ultrasónico HC-SR04.* [Blog]. [Consulta: 19 abril 2017]. Disponible en: <http://www.superrobotica.com/S320103.htm>

VERDEGAY, José Luis; “De los Conjuntos Fuzzy a la Soft Computing”. *Researchgate* [En línea]. 2005 (España), 24 (2), pp. 29-47. [Consulta: 12 abril 2017]. ISSN 0211-6642. Disponible en: <http://bit.ly/2jJug5O>

TRIPERO, Andrés; “Robótica y juguetería educativas: fronteras inmediatas de la innovación pedagógica en España”. *E-Innova UCM.* [En línea], 2017 (España), 1827, pp. 1-4. [Consulta: 17 septiembre 2017]. Disponible en: <https://biblioteca.ucm.es/revcul/e-learning-innova/127/art1827.pdf>

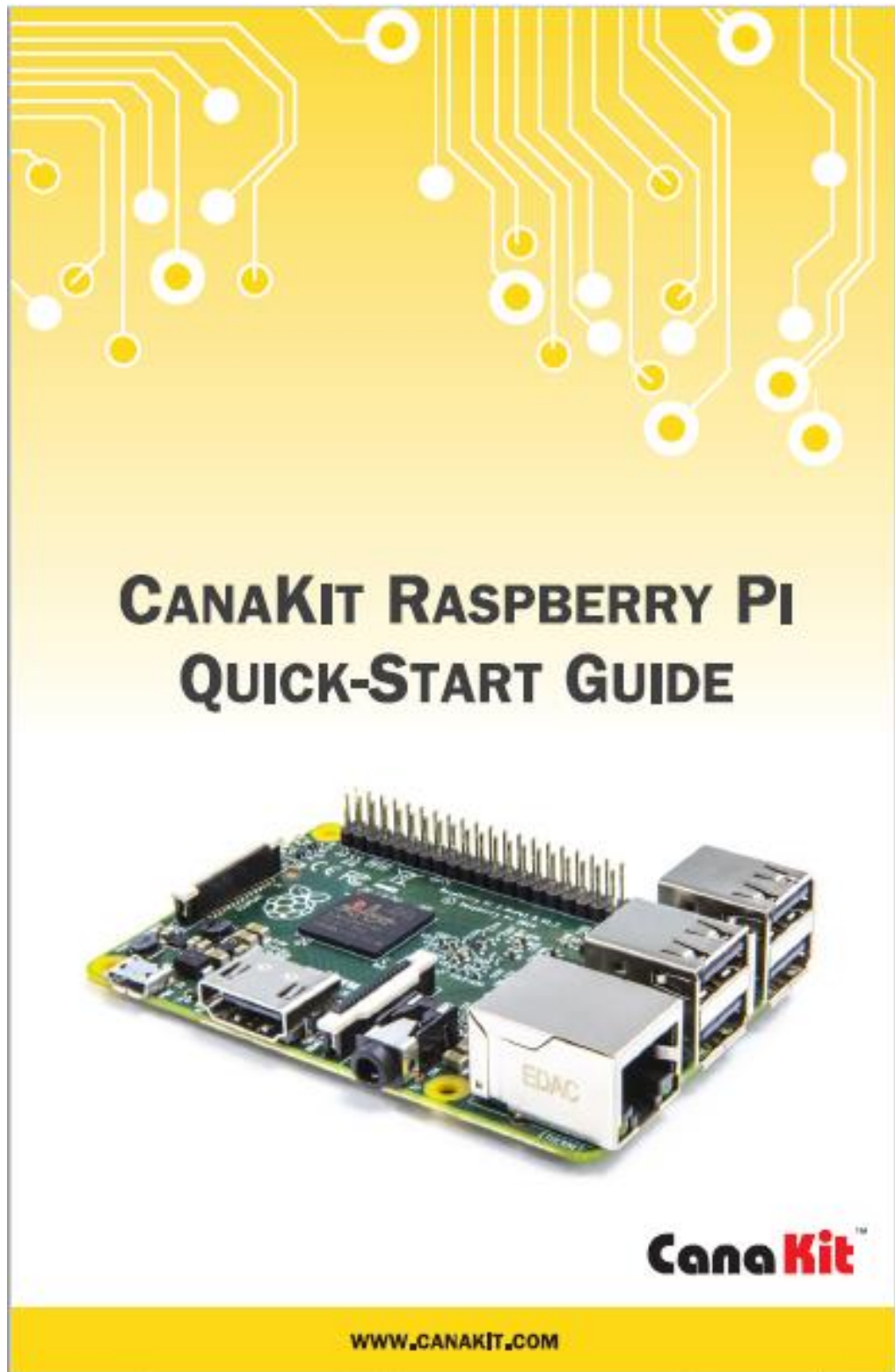
VIOLA, Paul; & JONES, Michael J. “Detecting pedestrians using patterns of motion and appearance”. *Null IEEE* [En línea], 2003, (United State of America), p. 734. [Consulta: 18 julio 2017]. Disponible en: <https://pdfs.semanticscholar.org/59e2/cee8a565ed8c569ac5946bd45e8f419e340b.pdf>

VIOLA, Paul; & JONES, Michael J. “Robust real-time face detection”. *International journal of computer vision.* 57,2 (2004), (United State of America) pp. 137-154

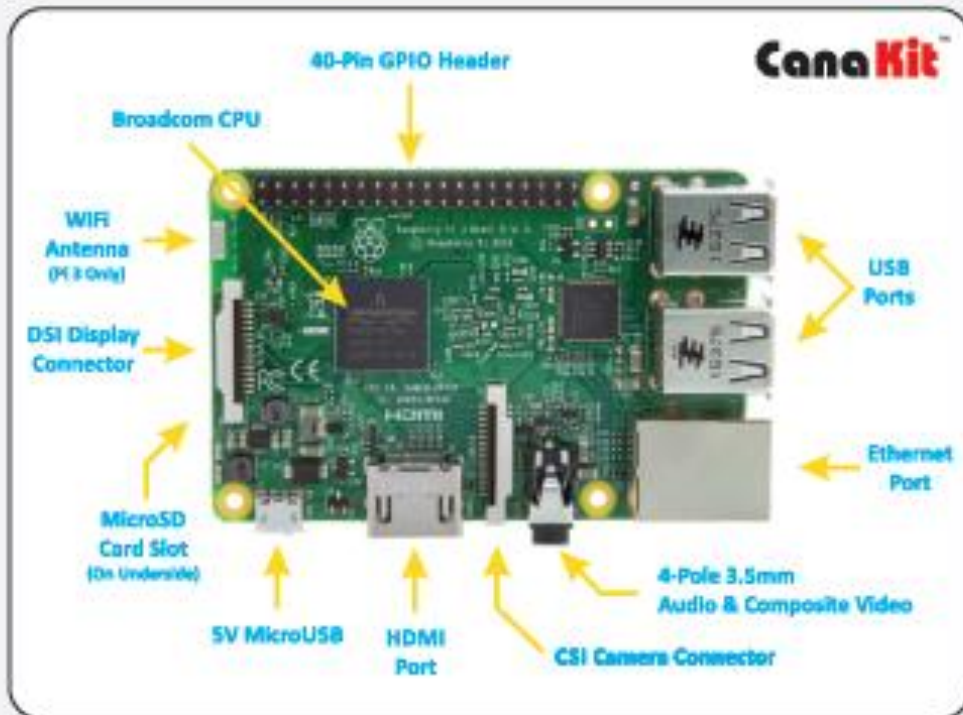
ZADEH, L. A., “Soft Computing and Fuzzy Logic”. *IEEE Software* [En línea], 1994, 11(6), pp. 48-56. [Consulta: 11 abril 2017]. DOI 10.1109/52.329401. Disponible en: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=329401&isnumber=7791>

ANEXOS

ANEXO A: Manual de minicomputador Raspberry Pi 3 modelo B



GETTING STARTED



1. If you have a case, start by installing the Raspberry Pi inside the case. You can find video instructions for CanaKit Raspberry Pi cases at:

www.canakit.com/pi-case

2. Insert a MicroSD card pre-loaded with NOOBS into the MicroSD card slot on the bottom side of the Raspberry Pi. If you don't have a pre-loaded MicroSD card you can find instructions on how to make one at:

www.raspberrypi.org/help/noobs-setup

3. Connect a USB keyboard and mouse to the USB ports.
4. Connect a HDMI monitor or TV to the Raspberry Pi using a HDMI cable. Ensure your monitor or TV is turned on and that you have selected the correct input.

5. If you intend to use an Operating System other than Raspbian, connect the Raspberry Pi to your network using an ethernet cable. If a wired Internet connection is detected, you will be presented with additional choices of Operating Systems. **Please ensure the Operating System that is selected is compatible with the version of the Raspberry Pi you are using.**

6. Once all connections have been made, it is now time to connect the power adapter to the board. When the power is connected, the Raspberry Pi will start to boot and you should be presented with the following menu.



Note: Once an Operating System has been installed, the NOOBS menu will no longer appear. If you need to get back to the NOOBS menu, press and hold the SHIFT key on your keyboard when turning on the Raspberry Pi.

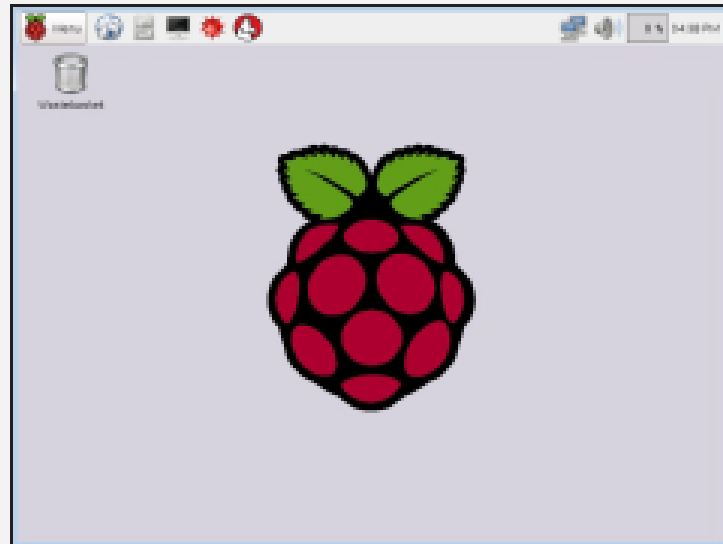
7. Select "Raspbian" or your choice of Operating System(s) from the menu. Click "Install" to begin the installation process.



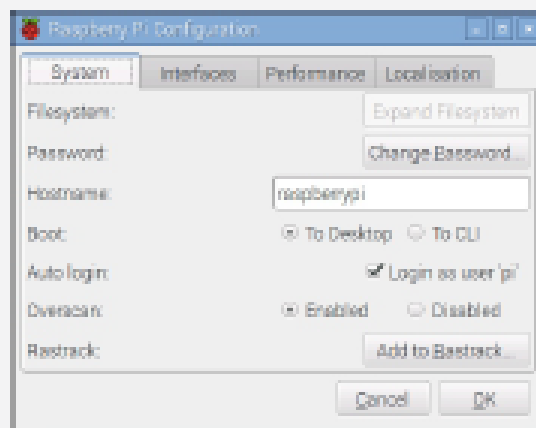
8. NOOBS will now start the installation of the selected Operating System(s). This process will take a few minutes. Once completed, you will see a message indicating the installation has completed. Press OK and the Raspberry Pi will reboot.

RASPBIAN JESSIE

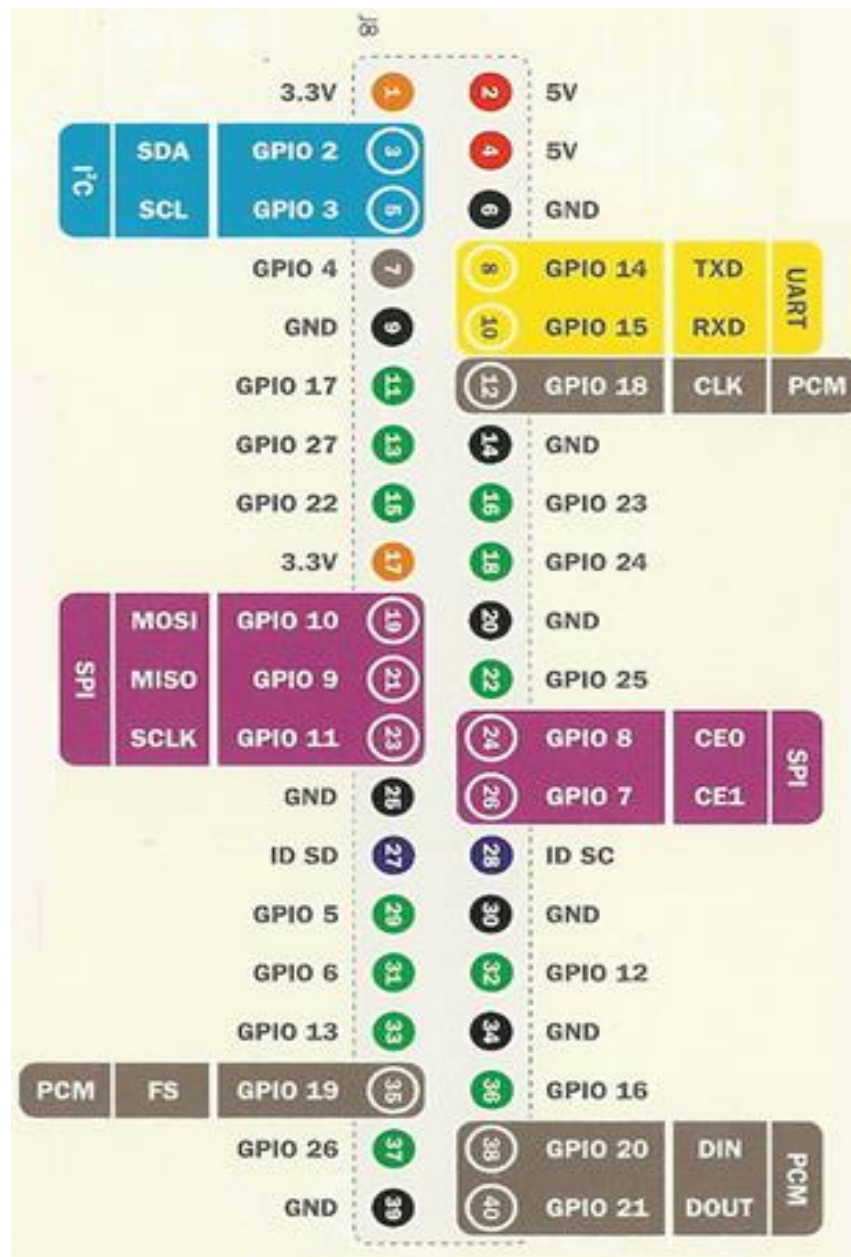
Starting with Version 4.1 of Raspbian (code-named Raspberry Jessie), the Raspberry Pi will automatically boot straight to the graphical desktop interface and not to the Linux command line.



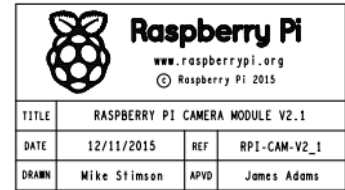
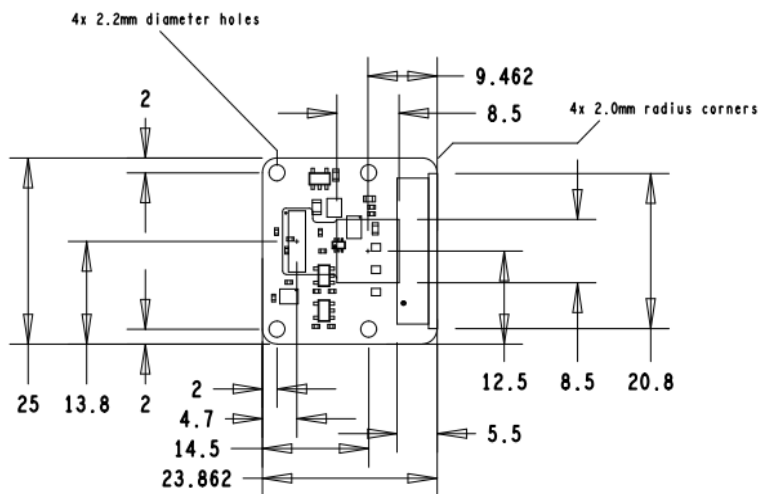
The first time you start your Raspberry Pi, it is recommended to launch the Raspberry Pi Configuration application in order to set system configuration parameters, such as default Password, Hostname, Locale, Timezone and Keyboard Layout. This tool can be found under the Preferences entry under the Main Menu.



ANEXO B: Disposición de pines GPIO en placa Raspberry Pi 3 B+



ANEXO C: Especificaciones técnicas del módulo de la cámara Raspberry Pi



HARDWARE SPECIFICATION	
Still resolution	8 Megapixels
Video modes	1080p30, 720p60 y 640 × 480p60/90
Linux integration	V4L2 driver available
C programming API	OpenMax IL and others available
Sensor	Sony IMX219
Sensor resolution	3280 x 2464 pixels
Sensor image área	3.68 x 2.76 mm (4.6 mm diagonal)
Pixel size	1/4 "
Focal length	3.04 mm
Horizontal field of view	62.2 degrees
Vertical field of view	48.8 degrees

SOFTWARE FEATURES	
Picture formats	JPEG, JPEG+RAW, GIF, BMP, PNG, YUV420, RGB888
Video formats	raw h.264
Effects	Negative, solarise, posterize, whiteboard, blackboar, sketch, denoise, emboss, oilpaint, hatch, gpen, pastel, watercolour, film, blur, saturation
Exposure modes	Auto, night, nighthpreview, blacklight, spotlight, sports, snow beach, verylong, fixedfps, antishake, Fireworks
Meteting modes	Average, spot, backlit, matrix
Automatic balance	Off, auto, sun, cloud, shade, tungsten, fluorescent, incandescent, flash
Triggers	Keypres, UNIX signal, timeout
Extra modes	Motion vectors, segmented video, live preview on 3D models



Sabertooth 2x12 User's Guide
April 2012



Input voltage: 6-24V nominal, 30V absolute max.

Output current: Up to 12A continuous per channel. Peak loads may be up to 25A per channel for a few seconds. These ratings are for input voltages up to 18v in still air without additional heatsinking.

5V switching BEC: Up to 1A continuous and 1.5A peaks across the entire range of input voltages.

Recommended power sources are:

- 5 to 18 cells NiMH or NiCd
- 2s to 6s lithium ion or lithium polymer. Sabertooth motor drivers have a lithium battery mode to prevent cell damage due to over-discharge of lithium battery packs.
- 6v to 24v lead acid
- 6v to 24v power supply (when in parallel with a suitable battery).

Dimensions:

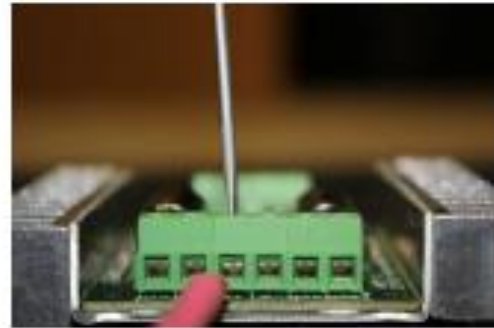
Size: 2.5" x 2.95" x .6" 64 x 75 x 16mm
Weight: 2.2oz

Hooking up the Sabertooth motor driver

All connections to the Sabertooth are done with screw terminals. This makes it easy to set up and reconfigure your project. If you've never used screw terminal connections before, here is a quick overview.



Step 1: Strip the wire which you are using approximately 1/4". The wires may be 14 gauge to 30 gauge



Step 2: With a small screwdriver, turn the top screw counter-clockwise until it stops gently.



Step 3: Insert the stripped portion of the wire into the opening in the screw terminal



Step 4: Turn the top screw clockwise until you encounter resistance, then tighten the screw firmly. Pull on the wire gently to ensure that it is secured.

Motor1 Terminals

Motor 1 is connected to terminals M1A and M1B as shown below. If the motor runs in the opposite way that you want, you may reverse the motor wires to reverse rotation.

Motor 2 is connected to terminals M2A and M2B.



The motors connect to terminals M1A/B and M2A/B

Signal Input Terminals S1 and S2

The input signals that control the Sabertooth are connected to terminals S1 and S2. If you are running in analog mode, it is important to have both the signal connected before applying power to the device. Otherwise, the motors may start unexpectedly.



The input signals connect to terminal S1 and/or S2

Battery Terminals B+ and B-

The battery or power supply is connected to terminals B- and B+. B- connects to the negative side of the battery (usually black.) B+ connects to the positive side of the battery (usually red or yellow.) It is usually best to connect the battery through a connector instead of directly to the motor driver. This

makes it easy to unplug the battery for charging, and prevents plugging in the battery backwards.



The battery connects to terminals B+ and B-

Warning! Be very careful to wire and plug in the battery and connector correctly. Connecting the battery backwards will destroy the Sabertooth and will void the warranty.

Power terminals 0V and 5V

The 0V and 5V connections are used to power and interface to low-power control circuits.

The 5V connection is a 5v power output. The 2x12 utilizes a 1A switching BEC to power the onboard electronics as well as to provide power to your receiver and up to 4 standard analog servos. You can power anything that requires 5V straight from the Sabertooth 2x12. There is no need for an external BEC unless you need more than 1 Amp! The BEC will work at full rated output throughout the Sabertooth's operating voltage range. You can use the BEC at full capacity whether you are running 7V or 24V in!

The 0V connection is the signal ground for the Sabertooth. In order to receive input signals correctly, it must be connected to the ground of the device sending the signals.

Using the 0V and 5v connections to power a radio receiver in R/C mode and potentiometer in analog mode is shown in Figures 2.1 and 2.2. *If you are using multiple Sabertooths running from the same radio receiver, only one should have the 5v line connected.* You can either take the red lead out of the connection housing or just clip the wire with a pair of cutters.



The 5V terminal can be used to power loads totaling no more than 1A, like a potentiometer or a radio receiver and a servo to two. The 0V signal must be connected to the ground of the device generating the input signal.



Figure 2.1: Analog input using a potentiometer powered from terminal 5V



Figure 2.2: R/C input using a receiver powered from terminal 5V

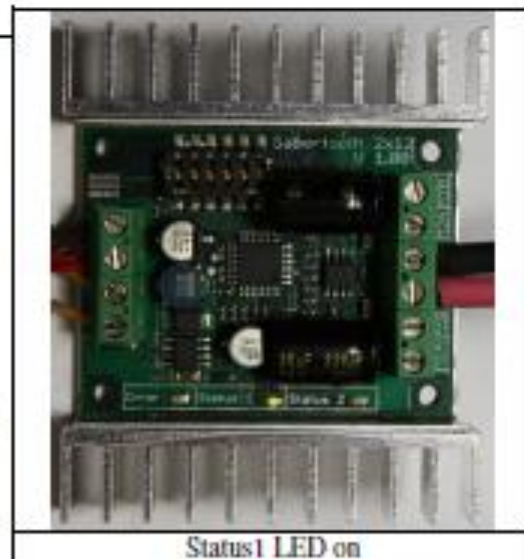
Status and Error LEDs

Sabertooth 2x12 has three indicator LEDs.

The green LED marked Status1 is used to communicate various information about the current state. In most cases Status1 acts as a power indicator. In R/C mode, it glows dimly if there is no RC link present and brightly if there is an RC link.

The green LED marked Status2 is used in lithium mode. It blinks to indicate the number of lithium cells detected. Also, when Status2 and Error are flashing at the same time, and you are experiencing no output from the motors, the unit is displaying low voltage mode. Charge your batteries first, and if that does not work, you will need a larger battery. Keep in mind that when a lot of current is pulled from a battery, the voltage will drop. The more depleted the battery, the worse the voltage drop. The Sabertooth will hold the unit in the error state for longer than the battery voltage drops. This is to stop the unit from stuttering and gives the user enough time to diagnose the flashing LEDs.

The red Error LED illuminates if the Sabertooth has detected a problem. It will light if the driver has shut down due to overheating or overcurrent.



Mounting your Sabertooth 2x12

The Sabertooth is supplied with four mounting holes. These can be used to attach it to your robot. The centers of the mounting holes form a 1.5" x 2" rectangle. The holes are .125 inches in diameter. The proper size screw is a 4-40 round head machine or wood screw. Four 5/8" long machine screws and nuts are included.

If your robot or device is constructed from insulating materials such as wood or plastic, it may be necessary to mount the Sabertooth on standoffs to allow air to circulate. This is shown in Figure 2.3

If your robot or device is constructed from metal, it is usually better to attach the bottom heat spreader of the Sabertooth directly to the frame, without standoffs. This will allow your frame to act as a heat sink and will cause the Sabertooth to run cooler. If your chassis is grounded, you must insulate the heatsink from the chassis. This is shown in Figure 2.4



Figure 2.3: Mounted to a wood frame using standoffs



Figure 2.4: Mounted directly to a metal frame

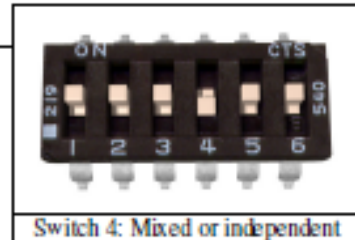
Mode 1: Analog Input

Analog input mode is selected by setting switches 1 and 2 to the UP position. Switch 3 should be either up or down, depending on the battery type being used. Inputs S1 and S2 are configured as analog inputs. The output impedance of the signals fed into the inputs should be less than 10k ohms for best results. If you are using a potentiometer to generate the input signals, a 1k, 5k or 10k linear taper pot is recommended. In all cases, an analog voltage of 2.5V corresponds to no movement. Signals above 2.5V will command a forward motion and signals below 2.5V will command a backwards motion.

There are three operating options for analog input. These are selected with switches 4, 5 and 6. All the options can be used independently or in any combination.

Switch 4: Mixed Mode

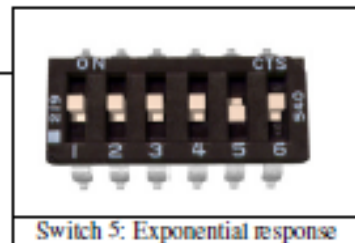
If switch 4 is in the UP position, the Sabertooth 2x12 is in **Mixed** mode. This mode is designed for easy steering of differential-drive vehicles. The analog signal fed into S1 controls the forward/back motion of the vehicle, and the analog signal fed into S2 controls the turning motion of the vehicle. If Switch 4 is in the DOWN position, the Sabertooth 2x12 is in Independent mode. In Independent mode, the signal fed to S1 directly controls Motor 1 (outputs M1A and M1B) and the signal fed to S2 controls Motor 2.



Switch 4: Mixed or independent

Switch 5: Exponential response

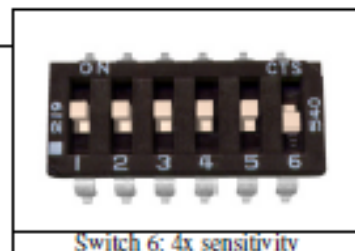
If switch 5 is in the DOWN position, the response to input signals will be exponential. This softens control around the zero speed point, which is useful for control of vehicles with fast top speeds or fast max turning rates. If switch 5 is in the UP position, the response is linear.



Switch 5: Exponential response

Switch 6: 4x sensitivity

If switch 6 is in the UP position, the input signal range is from 0v to 5v, with a zero point of 2.5v. If switch 6 is in the DOWN position, 4x sensitivity mode is enabled. In this mode, the input signal range is from 1.875V to 3.125V, with a zero point of 2.5v. This is useful for building analog feedback loops



Switch 6: 4x sensitivity

Note on using filtered PWM in Analog Mode

If you are using a filtered PWM signal from a microcontroller to generate the analog voltage, an R/C filter with component values 10k ohms and at least .1uf is recommended as shown in **Figure 4.1**. Using a larger value filter capacitor such as 1uf or 10uf will result in smoother motor operation, at a cost of slower transient response. A PWM frequency higher than 1000Hz is recommended.

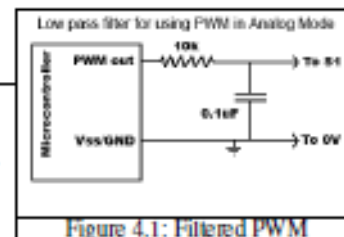


Figure 4.1: Filtered PWM

Mode 2: R/C Input

R/C input mode is used with a standard hobby Radio control transmitter and receiver, or a microcontroller using the same protocol. R/C mode is selected by setting switch 1 to the DOWN position and switch 2 to the UP position. If running from a receiver, it is necessary to obtain one or more servo pigtails and hook them up according to figure 5.1. If using a receiver pack, do not connect power to the 5V line of the Sabertooth because the maximum voltage it can tolerate is 6V.



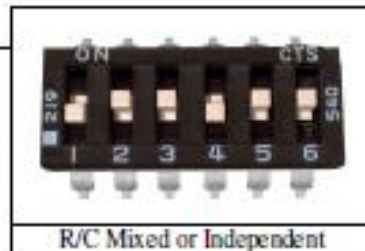
Figure 5.1: R/C connection

There are three operating options for R/C mode. These are selected with switches 4, 5 and 6.

Switch 4: Mixed Mode

When Switch 4 is in the UP position, Mixed mode is selected. In this mode, the R/C signal fed to the S1 input controls the forward/backwards motion of the vehicle. This is usually connected to the throttle channel of a pistol grip transmitter, or the elevator channel of a dual stick transmitter. The R/C signal fed to the S2 input controls the turning of the vehicle.

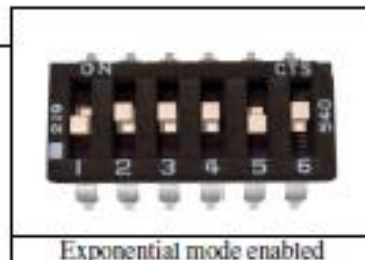
When switch 4 is in the DOWN position, Independent mode is selected. In this mode, the signal fed to the S1 input directly controls Motor 1 (M1A and M1B) and the signal fed to S2 controls Motor 2.



R/C Mixed or Independent

Switch 5: Exponential response

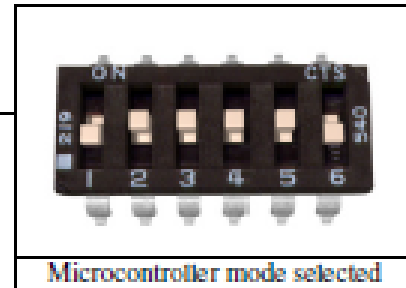
If switch 5 is in the UP position, the response is linear. If switch 5 is in the DOWN position, the response to input signals will be exponential. This softens control around the zero speed point, which is useful for control of vehicles with fast top speeds or fast max turning rates.



Exponential mode enabled

Switch 6: R/C Mode/Microcontroller mode select

If switch 6 is in the UP position, then the Sabertooth is in standard R/C mode. This mode is designed to be used with a hobby-style transmitter and receiver. It automatically calibrates the control center and endpoints to maximize stick usage. It also enables a Timeout Failsafe, which will shut down the motors if the Sabertooth stops receiving correct signals from the receiver.



If switch 6 is set in the DOWN position, then Microcontroller mode is enabled. This disables the Timeout Failsafe and auto-calibration. This means that the Sabertooth will continue to drive the motor according to the last command until another command is given. If the control link is possible unreliable – like a radio – then this can be dangerous due to the robot not stopping. However, it is extremely convenient if you are controlling the Sabertooth from a microcontroller. In this case, commanding the controller can be done with as little as three lines of code.

```
Output_High(Pin connected to S1)
Delay(1000us to 2000us)
Output_Low(Pin connected to S1)
```

A note on certain microprocessor receivers

Some receivers, such as the Spektrum AR6000, will output servo pulses before a valid transmitter signal is present. This will cause the Sabertooth to autocalibrate to the receiver's startup position which may not correspond to the center stick position, depending on trim settings. This may cause the motors to move slowly, even when the transmitter stick is centered. If you encounter this, either consult your receiver manual to reprogram the startup position, or adjust your transmitter trims until the motors stop moving. As a last resort, you can enter R/C microcontroller mode which will disable Sabertooth's autocalibration.





Mode 3: Simplified Serial Mode

Simplified serial uses TTL level single-byte serial commands to set the motor speed and direction. This makes it easy to interface to microcontrollers and PCs, without having to implement a packet-based communications protocol. Simplified serial is a one-direction only interface. The transmit line from the host is connected to S1. The host's receive line is not connected to the Sabertooth. Because of this, multiple drivers can be connected to the same serial transmitter. If using a true RS-232 device like a PC's serial port, it is necessary to use a level converter to shift the -10V to 10V rs-232 levels to the 0v-5v TTL levels the Sabertooth is expecting. This is usually done with a Max232 type chip. If using a TTL serial device like a microcontroller or a USB-to-TTL serial converter, the TX line of the microcontroller may be connected directly to S1.

Because Sabertooth controls two motors with one 8 byte character, when operating in Simplified Serial mode, each motor has 7 bits of resolution. Sending a character between 1 and 127 will control motor 1. 1 is full reverse, 64 is stop and 127 is full forward. Sending a character between 128 and 255 will control motor 2. 128 is full reverse, 192 is stop and 255 is full forward. Character 0 (hex 0x00) is a special case. Sending this character will shut down both motors.

Baud Rate Selection

Simplified Serial operates with an 8N1 protocol – 8 data bytes, no parity bits and one stop bit. The baud rate is selected by switches 4 and 5 from the following 4 options:

	
2400 Baud: 01x00x	9600 Baud: 01x10x
	
19200 Baud: 01x01x	38400 Baud: 01x11x

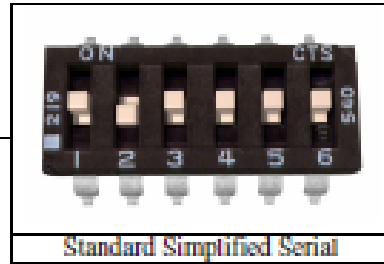
What baud rate to use is dependent on what your host can provide and the update speed necessary. 9600 baud or 19200 baud is recommended as the best starting points. If communication is unreliable, decrease the baud rate. If communications are reliable, you may increase the baud rate. The maximum update speed on the Sabertooth is approximately 2000 commands per second. Sending characters faster than this will not cause problems, but it will not increase the responsiveness of the controller either.

The baud rate may be changed with power on by changing the DIP switch settings. There is no need to reset or cycle power after a baud rate change.

There are 2 operating options for Simplified Serial, selected by the position of Switch 6.

Option 1: Standard Simplified Serial Mode

Serial data is sent to input S1. The baud rate is selected with switches 4 and 5. Commands are sent as single bytes. Sending a value of 1-127 will command motor 1. Sending a value of 128-255 will command motor 2. Sending a value of 0 will shut down both motors.



Standard Simplified Serial

Option 2: Simplified Serial with Slave Select

This mode is used when it is desirable to have multiple Sabertooth motor drivers running from the same serial transmitter, but you do not wish to use packetized serial. A digital signal (0v or 5v) is fed to the S2 input. This is controlled by the host microcontroller. If the signal on S2 is logic high (5v) when the serial command is sent, the driver will change to the new speed. If the signal on S2 is not high when the command is sent, then command will be ignored. Pseudo-code demonstrating this is shown below. After sending the signal, allow about 50 us before commanding the Slave Select line to a logic LOW to allow time for processing. A hookup diagram and example pseudo-code are shown in Figures 6.2 and 6.3.



Simplified Serial with Slave Select

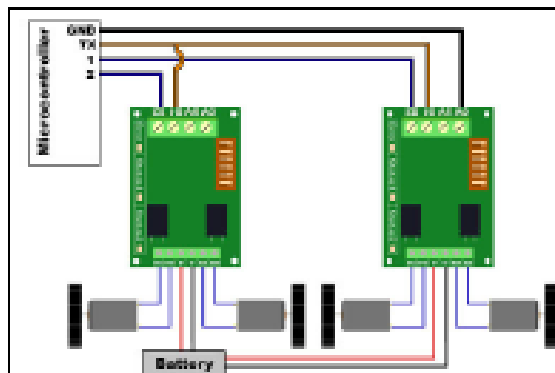


Figure 6.2: Hookup for Slave Select





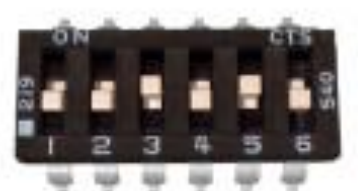
```
//set controller 1's speed
Output_High (S2 pin on controller 1)
USART_TX(controller 1 speed, 0 to 255)
Delay_us(50)
Output_Low (S2 pin on controller 1)

//set controller 2's speed
Output_High (S2 pin on controller 2)
USART_TX(controller 2 speed, 0 to 255)
Delay_us(50)
Output_Low (S2 pin on controller 2)
```

Figure 6.3: Pseudocode for Slave Select

Address Byte Configuration

Address bytes are set by switches 4, 5 and 6. Addresses start at 128 and go to 135. The switch settings for the addresses are shown in the chart below

 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>	 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>
Address: 128	Address: 129
 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>	 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>
Address: 130	Address: 131
 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>	 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>
Address: 132	Address: 133
 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>	 <p>A 6-position DIP switch assembly with positions 1-6. Position 1 is labeled 'ON' and position 6 is labeled 'CTS'. Position 1 has a small square indicator. Position 6 has a small square indicator. The numbers 219 and 540 are printed on the left and right sides respectively.</p>
Address: 134	Address: 135

Ultrasonic Ranging Module HC - SR04

Product features:

Ultrasonic ranging module HC - SR04 provides 2cm - 400cm non-contact measurement function, the ranging accuracy can reach to 3mm. The modules includes ultrasonic transmitters, receiver and control circuit. The basic principle of work:

- (1) Using IO trigger for at least 10us high level signal,
- (2) The Module automatically sends eight 40 kHz and detect whether there is a pulse signal back.
- (3) IF the signal back, through high level , time of high output IO duration is the time from sending ultrasonic to returning.

Test distance = (high level time×velocity of sound (340M/S) / 2,

Wire connecting direct as following:

- 5V Supply
- Trigger Pulse Input
- Echo Pulse Output
- 0V Ground

Electric Parameter

Working Voltage	DC 5 V
Working Current	15mA
Working Frequency	40Hz
Max Range	4m
Min Range	2cm
MeasuringAngle	15 degree
Trigger Input Signal	10uS TTL pulse
Echo Output Signal	Input TTL lever signal and the range in proportion
Dimension	45*20*15mm

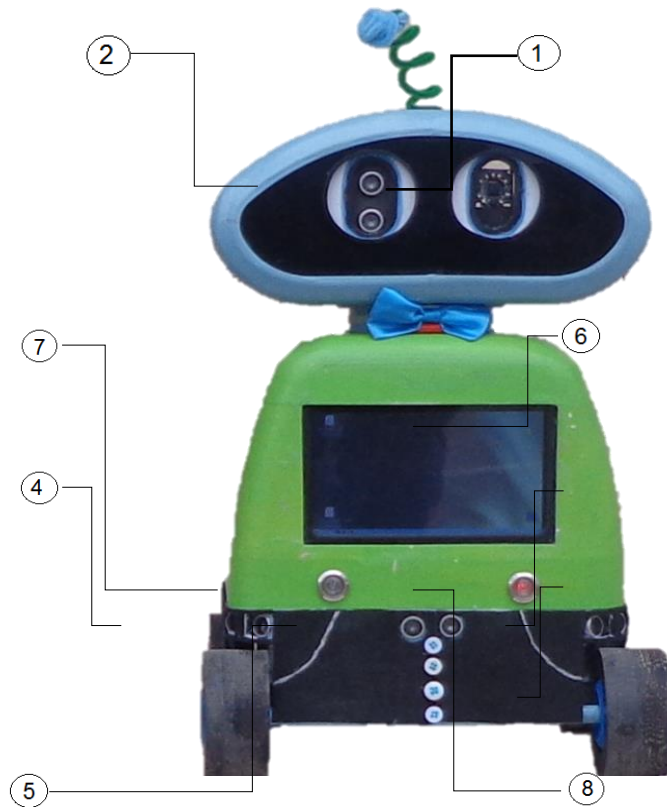
MANUAL DE USUARIO



PUSHAQ EL
JUGUETE
INTELIGENTE

PUSHAQ

COMPONENTES



1. Raspberry Pi Camera v2
2. Sensor ultrasónico HC-SR04
3. Sensor ultrasónico HC-SR04
4. Sensor ultrasónico HC-SR04
5. Sensor ultrasónico HC-SR04
6. LCD 7" Touchscreen
7. Switch (motores)
8. Switch (Raspberry pi)
9. Sistema de locomoción



- 10. Sensor ultrasónico HC-SR04
- 11. Sensor ultrasónico HC-SR04
- 12. Motor Nemesis
- 13. Sistema de locomoción
- 14. Sensor Sharp “GP2Y0D815Z0F”

Pushaq es robot inteligente que tiene la posibilidad de interactuar directamente con un niño a la vez, para tener acceso al juego, el rostro del niño debe estar almacenado en la base de datos.

Cuenta con una cámara infrarroja encargada de capturar escenas.

El sistema sensorial evita la colisión con objetos y mantendrán siempre al robot a una distancia prudente del infante.

El sistema de locomoción implementado será puesto en marcha al presionar su respectivo switch. Verificando antes que el piso sea plano sin agujeros ni escales.

El switch del numeral 8 que se muestra en la imagen de los componentes del robot, enciende el sistema. Inicialará con una presentación y posteriormente aparecerá en pantalla la interfaz gráfica, en la cual el niño tendrá la posibilidad de escoger entre varias opciones para la interacción lúdica.

INTERFAZ GRÁFICA DE USUARIO (GUI)



1. Narra cuentos infantiles.
2. Reproduce canciones infantiles.
3. Identifica el rostro para desbloquear el reconocimiento de figuras y colores de los objetos que tome el niño.
4. Almacena y visualiza fotografías.
5. Se activa el juego decorando la casa de Puchaq, consisten en ir colocando las figuras geométricas sobre la estructura de la casa, el robot asistirá al niño, diciéndolo el nombre de la pieza que debe colocar.
6. Al presionar esta opción la raspberry se apagará dando por terminado el proceso

DATOS TÉCNICOS

Raspberry pi 3:

- Procesador Quad Core a una velocidad de 1.2 GHz
- Memoria RAM de 1 GB
- 4 puertos USB
- 40 pines GPIO
- Puerto full HMI y Ethernet
- Conectividad Bluetooth 4.1 y Wi-Fi 802.11n
- VideoCore IV núcleo de gráficos 3D
- Tarjeta Micro SD de 32 Bit para cargar el SO
- Interfaz de cámara (CSI) y pantalla (DSI)
- Fuente de alimentación de 2.5 A, a 5 V en DC.

Arduino uno:

- Microcontrolador AVR
- Puertos digitales y analógicos de E/S
- Tensión de 5 VDC
- Oscilador de cristal de 16 MHz
- Compatible con todo los SO

Sabertooth 2x12:

- Frecuencia de conmutación ultrasónica
- 12A continuos, 25A pico por canal
- 6V hasta 24V
- Protección térmica y de sobrecorriente
- Modo de protección de litio
- Modos de entrada: analógico
- Tamaño: 2.3 "x 3" x .7 "
- 59 x 75 x 17 mm

Cámara raspberry pi:

- Sensor IMX219
- 8 megapíxeles.
- 15 pines.
- Resolución 2592 x 1994 pixeles

Motor:

- 12V, 7A
- 2750RPM

Sensor ultrasónico HC-SR04:

- Angulo de medición 30°
- Frecuencia 40 KHz
- detecta hasta 4m de distancia.

Baterías lipo:

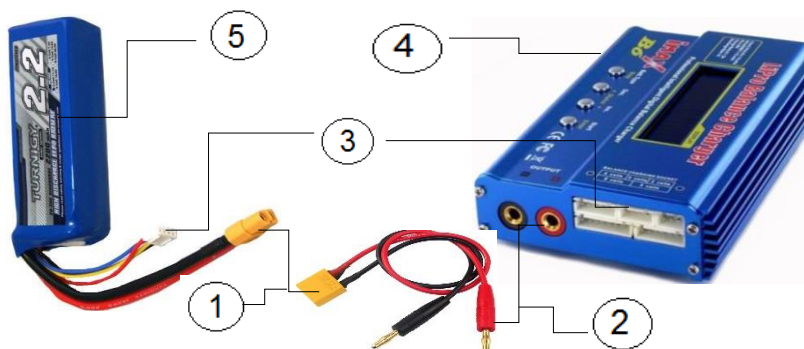
- Litio y Polímero.
- Tres celdas (3S).
- 11.1V y 2200mAh.
- Tasa de descarga de 20C de constante y 30C puntales.

Prototipo:

- Dimensiones 50x30 cm y una profundidad de 17cm.
- Espesor de 3mm
- Consta de 3 partes principales, la pieza denominada chasis, el cuerpo y la cabeza.

El material que está construido es PLA

PASOS PARA CARGAR BATERÍAS



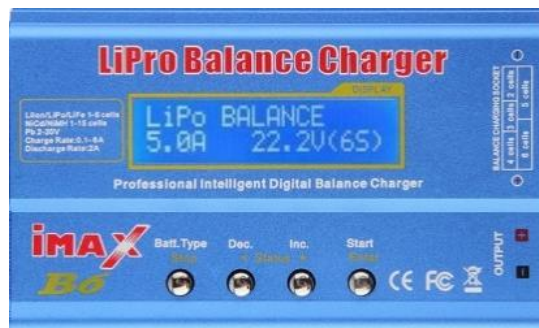
1. Conector Anderson, “macho – hembra”
2. Plug banana, “macho – hembra”

3. Conector espadín, “macho – hembra”
4. Cargador balanceador de baterías lipo
5. Batería lipo 11.1V a 2.2A

COMO INICIAR LA CARGA DE LAS BATERÍAS

El conector Anderson hembra se conecta el conector Anderson macho, luego los plug banana macho al plug banana hembra del cargador. De la misma forma el conector espadín hembra de la batería al conector espadín macho del cargador en el puerto 3 celdas.

Una vez echas la conexiones cargador batería se configura el cargador en *LIPO BALANCE* con los selectores, como se indica en la figura.



Una vez seleccionado lipo balance, con los selectores se configura a 2.2A que es el amperaje y 3S que son el número de celdas compuestas por las baterías de pushaq.



ADVERTENCIA

Cumpla con todas las advertencias y normas de seguridad relacionadas con el uso del dispositivo robótico.

- A la hora de cargar las baterías asegurarse que este fuera del alcance de los niños.
- No utilice sustancias químicas o detergentes, puede provocar la decoloración o cortocircuitos.
- Si el dispositivo se calienta, interrumpir el uso.
- No utilice el dispositivo si la pantalla está quebrada o rota. El vidrio puede causar lesiones en las manos y rostro.
- No desarme, modifique ni repare el prototipo. En caso de avería o alguna modificación, comunicarse a los contactos de servicio técnico.
- Cuando los niños utilizan el dispositivo asegurarse que lo haga correctamente.
- No manipule las baterías lipo en caso de que se encuentre averiadas comunicarse con los contactos de servicio técnico.
- No guarde el dispositivo en área húmeda o mojada.
- No permita que el producto se caiga ni lo sometan a impactos fuertes.
- Para la ejecución 3 de la interfaz verificar que el espacio sea el mismo en el que se hallaron los rangos de colores
- No dejar la batería descargada durante mucho tiempo, podría acortar su vida útil y su rendimiento. También podría provocar el funcionamiento incorrecto del dispositivo.
- La batería que energiza la raspberry tiene una duración de 10 horas.
- El sistema de locomoción puede estar activado una hora.

SERVICIO TÉCNICO:

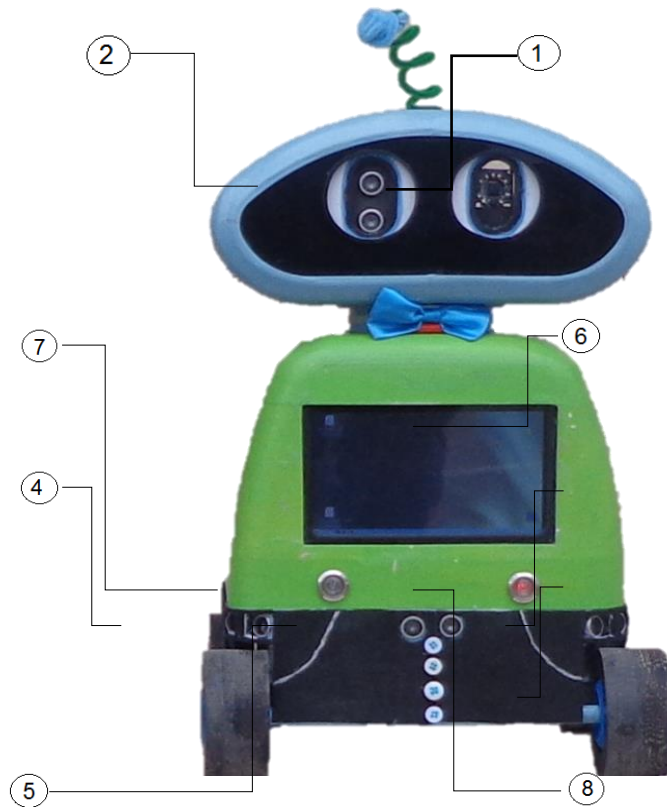
estefasalin46@gmail.com (0983293007)
marcelo199167@hotmail.com (0994378723)

MANUAL TÉCNICO



PUSHAQ EL
JUGUETE
INTELIGENTE

COMPONENTES



1. Raspberry Pi Camera v2
2. Sensor ultrasónico HC-SR04
3. Sensor ultrasónico HC-SR04
4. Sensor ultrasónico HC-SR04
5. Sensor ultrasónico HC-SR04
6. LCD 7" Touchscreen
7. Switch (motores)
8. Switch (Raspberry pi)
9. Sistema de locomoción



- 10. Sensor ultrasónico HC-SR04
- 11. Sensor ultrasónico HC-SR04
- 12. Motor Nemesis
- 13. Sistema de locomoción
- 14. Sensor Sharp “GP2Y0D815Z0F”

Pushaq es robot inteligente que tiene la posibilidad de interactuar directamente con un niño a la vez, para tener acceso al juego, el rostro del niño debe estar almacenado en la base de datos.

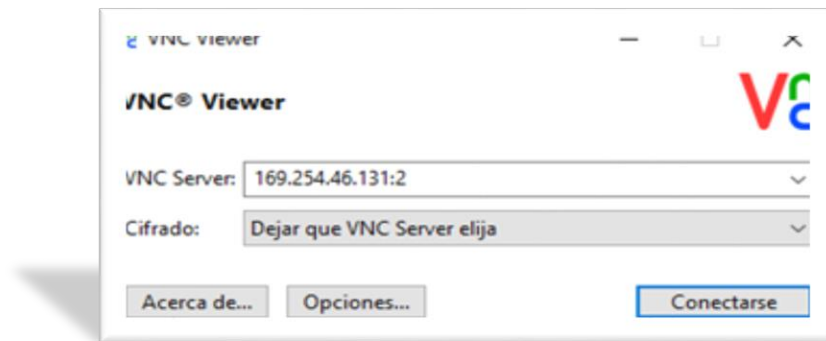
COMO INGRESAR A LA RASPBERRY PI

Ingresa mediante un escritorio remoto “VNC” a la raspberry pi, la cual debe estar conectada vía Ethernet a su computador.

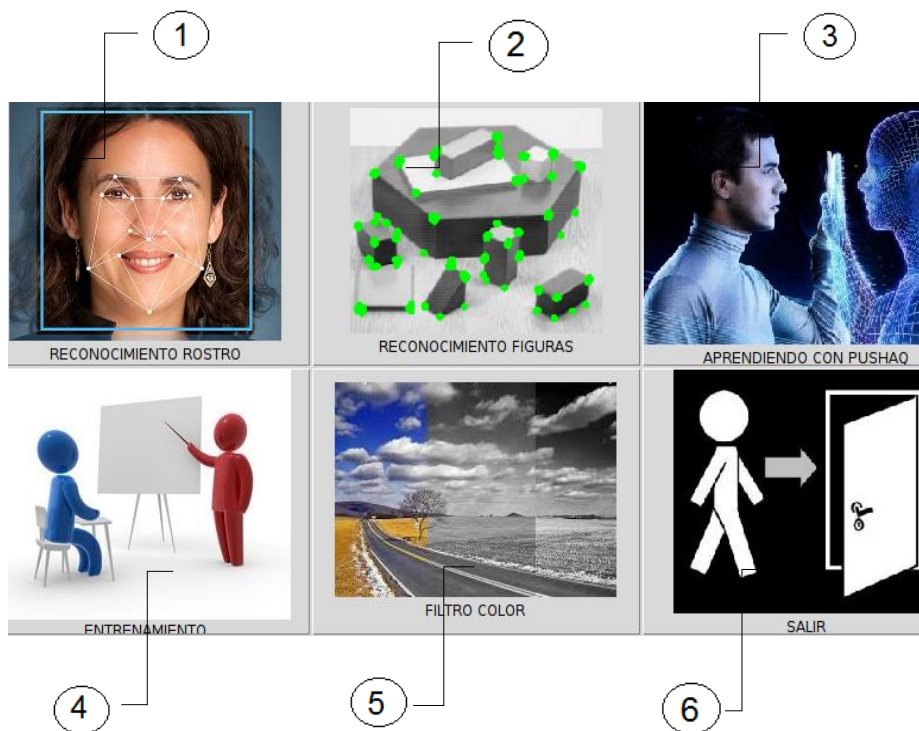
Como primer paso se debe configurar la IP de su computador, ingresando una IP estática que se encuentre en el mismo dominio de la IP de la placa Raspberry Pi para poder establecer comunicación.

Se ingrese al minicomputador con la Dirección IP “**169.254.46.131**”. Una vez configurada la IP de su computador y conectada se procede a dar e permiso de ingreso con el comando

vncserver:1, donde el :1 es el puerto por el que se establecerá la comunicación, si todo está correcto le pedirá una clave que es *raspberry*.



INTERFAZ GRÁFICA DEL TÉCNICO



1. Pruebas de reconocimiento. (reconocimiento.py)
2. Pruebas de reconocimiento de figuras geométricas (figuras.py)
3. Pruebas identifica el rostro para desbloquear el reconocimiento de figuras y colores de los objetos que estén frente a la cámara.(GUI.py)
4. Realiza el entrenamiento de los rostros almacenados en la base de datos. (train.py)
5. Esta opción es para calcular los valores RGB de los colores que desea reconocer. (filtro_tiempo_real.py)
6. Al presionar esta opción la raspberry se apagará dando por terminado el proceso.

DATOS TÉCNICOS

Raspberry pi 3:

- Procesador Quad Core a una velocidad de 1.2 GHz
- Memoria RAM de 1 GB
- 4 puertos USB
- 40 pines GPIO
- Puerto full HMI y Ethernet
- Conectividad Bluetooth 4.1 y Wi-Fi 802.11n
- VideoCore IV núcleo de gráficos 3D
- Tarjeta Micro SD de 32 Bit para cargar el SO
- Interfaz de cámara (CSI) y pantalla (DSI)
- Fuente de alimentación de 2.5 A, a 5 V en DC.

Arduino uno:

- Microcontrolador AVR
- Puertos digitales y analógicos de E/S
- Tensión de 5 VDC
- Oscilador de cristal de 16 MHz
- Compatible con todo los SO

Sabertooth 2x12:

- Frecuencia de conmutación ultrasónica
- 12A continuos, 25A pico por canal
- 6V hasta 24V
- Protección térmica y de sobre cargas
- Modo de protección de litio
- Modos de entrada: analógico
- Tamaño: 2.3 "x 3" x .7 "
- 59 x 75 x 17 mm

Cámara raspberry pi:

- Sensor IMX219
- 8 megapíxeles.
- 15 pines.
- Resolución 2592 x 1994 pixeles

Motor:

- 12V, 7A
- 2750RPM

Sensor ultrasónico HC-SR04:

- Angulo de medición 30°
- Frecuencia 40 KHz
- detecta hasta 4m de distancia.

Baterías lipo:

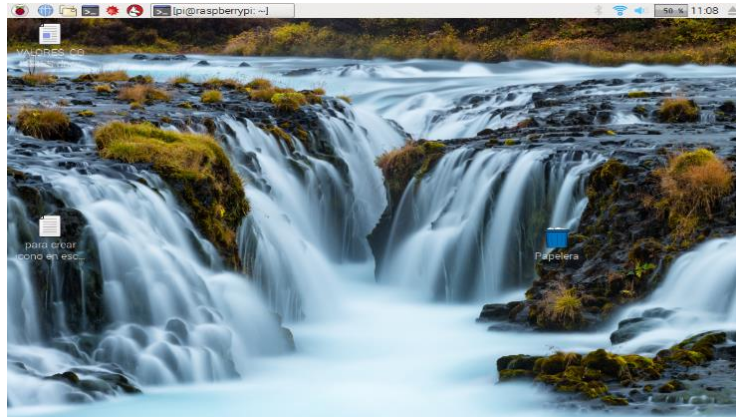
- Litio y Polímero.
- Tres celdas (3S).
- 11.1V y 2200mAh.
- Tasa de descarga de 20C de constante y 30C puntales.

Prototipo:

- Dimensiones 50x30 cm y una profundidad de 17cm.
- Espesor de 3mm
- Consta de 3 partes principales, la pieza denominada chasis, el cuerpo y la cabeza.
- El material que está construido es PL

RECONOCIMIENTO

Ingreso al sistema operativo de la raspberry pi.



En la siguiente imagen se muestra la interfaz gráfica del sistema operativo con el que cuenta el robot.

Ingresar al directorio `/home/pi/tesis/GUI`. Dentro de este directorio se encontrarán varios archivos que dependiendo del error al abrir el que corresponda, dando clic derecho y abrir con python 2.7.3

RECONOCIMIENTO DE ROSTROS

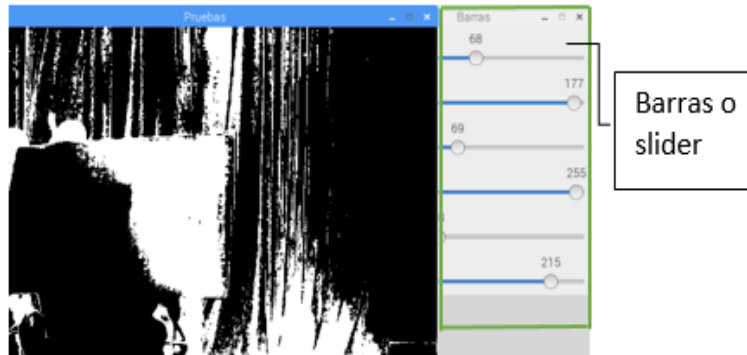
En caso de existir inconvenientes en el reconocimiento de rostros por cambios de ambientes de trabajo del robot, se debe modificar el archivo `config.py`, disminuyendo o aumentando el **umbral perceptrónico** según sea el caso, tomando en cuenta que los valores óptimos para el reconocimiento oscilan desde 1000 a 3000.

```
# Define el nivel de confianza en la variable POSITIVE para el reconocimiento facil (3000)  
POSITIVE_THRESHOLD = 1950.0 #3000.0
```

Umbral Perceptrónico

RECONOCIMIENTO DE FIGURAS GEOMÉTRICAS

Si el ambiente de funcionamiento del robot cambia también se debe cambiar los valores RGV, ejecutando *filtro_color_real.py* o filtro color como se muestra en la ilustración 5 de la interfaz gráfica del técnico.



Se dispone de 6 barras, para mover hasta que el cuerpo enfocado aparezca completamente en la pantalla en color blanco y lo que le rodea debe tornarse negro. Los números sobre las barras son los valores de las bandas espectrales. Repitiendo este proceso para cada color que desea detectar. Y posteriormente modificando en *figuras.py* los siguientes valores.

```
##Rojo
lower_red    = np.array([164, 119, 144])
upper_red    = np.array([179, 145, 171])
#Naranja
lower_orange = np.array([169, 110, 140])
upper_orange = np.array([175, 130, 175])
#Azul
lower_blue   = np.array([110, 102, 112])
upper_blue   = np.array([127, 147, 145])
#Verde
lower_green  = np.array([112, 44, 100])
upper_green  = np.array([125, 66, 135])
#Amarillo
lower_yellow = np.array([18, 60, 106])
upper_yellow = np.array([31, 85, 135])
###Purpura
lower_purple = np.array([ 0, 46, 110])
upper_purple = np.array([150, 70, 130])
```

ADVERTENCIA

Cumpla con todas las advertencias y normas de seguridad relacionadas con el uso del dispositivo robótico....

- No utilice sustancias químicas o detergentes, puede provocar la decoloración o cortocircuitos.
- Si el dispositivo se calienta, interrumpir el uso.
- No utilice el dispositivo si la pantalla está quebrada o rota. Cámbiela o suspenda su uso inmediatamente.
- Antes de manipular los dispositivos electrónicos, asegúrese que la energía este suspendida de la misma manera manipular con pinzas o guantes para evitar cargas electrostáticas.
- No guarde el dispositivo en área húmeda o mojada.
- No permita que el producto se caiga ni lo sometan a impactos fuertes.
- No cortocircuite las baterías en caso de fallas ya que puede ocasionar daños irreparables.
- No dejar la batería descargada durante mucho tiempo, podría acortar su vida útil y su rendimiento. También podría provocar el funcionamiento incorrecto del dispositivo.
- Modificar únicamente los parámetros especificados en este manual, caso contrario perderá la garantía.

DATOS DEL FABRICANTE:

estefasalin46@gmail.com (0983293007)

marcelo199167@hotmail.com (0994378723)