



Universidad
Continental

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería Mecatrónica

Trabajo de Investigación

**Diseño e implementación de un robot humanoide
para terapia de niños con trastorno del
espectro autista**

Dante Jim Randal Gallo Torres

Huancayo, 2019

Para optar el Grado Académico de Bachiller
en Ingeniería Mecatrónica



Repositorio Institucional Continental

Trabajo de Investigación



Obra protegida bajo la licencia de [Creative Commons Atribución-NoComercial-SinDerivadas 2.5 Perú](https://creativecommons.org/licenses/by-nc-nd/2.5/peru/)

DEDICATORIA

Este proyecto de investigación lo dedico a mis padres Norma y Humberto, porque son los pilares más importantes en mi vida y gracias a ellos he logrado culminar mis estudios superiores, de igual manera a mi hermano Oliver, por haberme brindado siempre su apoyo en temas académicos y a mi hermano Miguel que cuida mis pasos desde el cielo.

ÍNDICE

RESUMEN.....	vii
ABSTRACT.....	viii
CAPÍTULO I: PLANTEAMIENTO DEL ESTUDIO.....	9
1.1 Planteamiento del problema	9
1.2 Formulación del problema	10
1.2.1 Problema general.....	10
1.2.2 Problema específico	10
1.3 Objetivos.....	10
1.3.1 Objetivo general.....	10
1.3.2 Objetivo específico	10
1.4 Justificación e importancia	11
CAPÍTULO II: MARCO TEÓRICO.....	12
2.1. Antecedentes del problema	12
2.1.1. Antecedentes Nacionales.....	12
2.1.2. Antecedentes Internacionales	13
2.2. Bases teóricas	13
2.2.1. Autismo	13
2.2.1.1. Definición.....	13
2.2.1.2. Causas	15
2.2.1.3. Tratamiento	15
2.2.1.4. Tipos de tratamiento	16
2.2.1.5. Terapia con robots.....	17
2.2.2. Visión Artificial.....	18
2.2.2.1. Definición.....	18
2.2.2.2. Procesamiento de imágenes	19
2.2.2.3. Redes neuronales aplicado a la visión artificial.....	20

2.2.3.	Control de robots.....	21
2.2.3.1.	Sensores	21
2.2.3.2.	Actuadores	22
2.2.4.	Telemetría.....	23
2.3.	Definición de términos básicos	24
2.3.1.	Robot Humanoide	24
2.3.2.	Inteligencia artificial.....	24
2.3.3.	Autoaprendizaje	24
CAPÍTULO III: METODOLOGÍA.....		25
3.1.	Introducción.....	25
3.2.	Lista de exigencias	25
3.3.	Caja negra.....	28
3.3.1.	Entradas.....	28
3.3.1.1.	Voz del niño.....	28
3.3.1.2.	Orden desde el aplicativo móvil	28
3.3.1.3.	Orden desde la pantalla touch	28
3.3.2.	Salidas	29
3.3.2.1.	Reproducir mismo audio.....	29
3.3.2.2.	Mover articulaciones del robot	29
3.3.2.3.	Comunicación verbal con el niño	29
3.4.	Estructura de funciones	30
3.4.1.	Estructura de funciones por dominio	30
3.4.1.1.	Dominio electrónico	30
3.4.1.2.	Dominio de control.....	31
3.4.1.3.	Dominio mecánico	31
3.4.2.	Caja de cristal	31
3.5.	Matriz morfológica	32
3.5.1.	Matriz morfológica de dominio electrónico.....	32

3.5.2.	Matriz morfológica de dominio de control	34
3.5.3.	Matriz morfológica de dominio mecánico.....	35
3.6.	Estudio técnico	36
3.6.1.	Concepto de solución de dominio de electrónica.....	36
3.6.2.	Concepto de solución de dominio de control	38
3.6.3.	Concepto de solución del dominio mecánico.....	39
CAPÍTULO IV: DISEÑO.....		40
4.1.	Dominio Electrónico – Análisis de planos por etapa.....	40
4.1.1.	Primera etapa.....	40
4.1.2.	Segunda Etapa	42
4.1.2.1.	Sensor Ultrasónico	42
4.1.2.2.	Módulo Bluetooth HC 06.....	43
4.1.2.3.	Módulo DFPlayer mini MP3	44
4.1.2.4.	Módulo ISD 1820.....	46
4.1.2.5.	LM 386	47
4.1.3.	Tercera etapa – Pantalla TFT con Arduino Uno	47
4.1.4.	Cuarta etapa – Interfaz Arduino – Matlab.....	48
4.1.5.	Quinta etapa – Motores.....	49
4.1.5.1.	Motorreductor con driver L298N	49
4.1.5.2.	Servomotor MG	50
4.2.	Dominio de control.....	51
4.2.1.	Diagrama de flujo general	51
4.2.2.	Primera etapa – Energizar robot y detectar presencia.....	52
4.2.3.	Segunda etapa – Reconocimiento de rostro.....	53
4.2.4.	Tercera etapa – Aplicativo Móvil.....	65
4.2.5.	Quinta etapa – Mover robot.....	67
4.3.	Dominio Mecánico	67
4.3.1.	Diseño y planos.....	67

4.3.1.1.	Diseño y planos del brazo.....	67
4.3.1.2.	Diseño y planos del antebrazo.....	69
4.3.1.3.	Diseño y planos del pecho.....	70
4.3.1.4.	Diseño y planos de la cabeza y ensamblaje del robot.....	72
BIBLIOGRAFÍA.....		74
ANEXOS.....		75
1.	Procesamiento de imágenes 1	75
2.	Procesamiento de imágenes 2	78
3.	Procesamiento de imágenes 3	79
4.	Etapa de prueba.....	80
5.	Primera neurona – Reconocimiento	81
6.	Segunda neurona – Reconocimiento	83
7.	Tercera Neurona – Validación.....	84
8.	Cuarta neurona – Validación	86

RESUMEN

El autismo no es una enfermedad es una condición, sin embargo, esta condición aqueja a la sociedad actual no solo en el país, sino en todo el mundo, el TEA (trastorno del espectro autista), no siempre es tratado adecuadamente, en Perú, solo existen 3 organizaciones públicas que tratan a personas con TEA, y tampoco existe un registro oficial de la cantidad de personas autistas en el Perú, los tratamientos en Perú son por terapias y métodos, sin embargo no se hace un uso tecnológico adecuado para las terapias, es por ello que esta investigación estará enfocado en optimizar los tratamientos.

El robot que será presentado contara con ciertas cualidades que serán elegidas según los tipos de terapias, y se hará un fusión para que el robot sea capaz de dar tratamiento en diferentes enfoques y optimice las múltiples terapias, los enfoques de hablas y voz podrán unificarse, como cualidades adicionales del robot, este será orientado al trabajo en conjunto con mas de un niño de ser necesario, para ello contará con visión artificial, en este punto se dará un aporte muy importante acerca de técnicas y algoritmos de visión artificial, ya que en esta ocasión el trabajo de visión artificial será por medio de algoritmos de redes neuronales, además de ello el robot tendrá tipos de controles como manual, automático y teleoperado, la combinación de todos estos, y la aplicación en la terapia infantil, es un aporte en beneficio del avance tecnológico en el país.

ABSTRACT

Autism is not a disease is a condition, however, this is a condition for society today not only in the country, but throughout the world, the ASD (autism spectrum disorder), is not always treated correctly, in the Peru, there are only 3 public organizations that are related to people with ASD, and that there is no official record of the number of people with autism in Peru, treatments in Peru are therapies and methods, however, it is not a appropriate use for therapies That is why this research will focus on optimizing treatments.

The robot that will be presented will have the people chosen according to the types of therapies, and a fusion will be made so that the robot is able to give treatment in different ways and optimal the multiple therapies, speaking skills and voice for unify, As it is a robot, you can guide the work in conjunction with the child and the child, for example, in the sense of artificial vision, at this point. this occasion the work of artificial vision the means of algorithms of neural networks, in addition to the robot will have types of controls like manual, automatic and teleoperado, the combination of all these, and the application in the infantile therapy, is a contribution in Benefit of the advance technology in the country.

CAPÍTULO I: PLANTEAMIENTO DEL ESTUDIO

1.1 Planteamiento del problema

EL espectro del trastorno autista (TEA), es un problema que aqueja a niños, jóvenes y adultos en nuestra sociedad, los TEA se pueden definir como un conjunto de complejos trastornos que afectan el desarrollo cerebral, según la Organización Mundial de Salud (OMS), la tasa media mundial de prevalencia es de 62 cada 10000 habitantes, lo que significa que un niño de cada 160 padece un trastorno del espectro autista, en Perú no existen cifras oficiales exactas de cuantas personas viven con el TEA, sin embargo, hay cifras de casos atendidos por el Ministerio de Salud (MINSA), en el año 2016 se atendió a 4447 personas de ambos sexos, siendo el 80% menores de 18 años. Y según el Ministerio de Educación (MINEDU), hay 5039 estudiantes con un tipo de trastorno autista, de los cuales 72.05% estudian en escuelas públicas y el 27.94% en escuelas privadas.

Según un estudio del diario El Comercio realizado en abril de 2018, indica que en el Perú solo hay 3 entidades públicas que realizan diagnóstico para detectar y tratar el TEA, los cuales son: El hospital Larco Herrera, el Instituto Nacional de Salud Mental Honorio Delgado Hideyo Noguchi (INSM), y el hospital Hermilio Valdizán, todos ellos laboran en Lima, en base a lo precisado por Rolando Pomalima, jefe de psiquiatría del INSM, quien indicó que en el Perú solo hay un promedio de 50 a 60 especialistas en psiquiatría infantil,

El tratamiento de este trastorno, no es algo barato, un costo promedio mensual va desde 1500 soles a 4 mil soles, los cuales son solo para cubrir gastos de terapia.

En la actualidad existen muchos métodos para el tratamiento del autismo en niños, el que estudiaremos en este proyecto de investigación es el tratamiento mediante robots, y como ir dándole a este robot beneficios adicionales que ayuden al tratamiento, y que no representen un gasto mayor.

1.2 Formulación del problema

Para realizar este proyecto de investigación con éxito se plantean los siguientes problemas, los cuales serán resueltos en la presente investigación.

1.2.1 Problema general

- Las clínicas u hospitales que realizan tratamiento del autismo, ¿cuentan con robots humanoides como parte de terapia en niños autistas?

1.2.2 Problema específico

- ¿Tener un robot para terapia en el tratamiento de niños autistas podrá ser de bajo costo?
- ¿Será posible realizar un robot con visión artificial para el tratamiento del autismo?
- ¿Será posible diseñar un prototipo a escala para demostrar la funcionalidad del sistema?

1.3 Objetivos

A continuación, se presentan los objetivos principales de este proyecto de investigación, y en base a ellos direccionar la investigación a puntos específicos.

1.3.1 Objetivo general

- Diseñar e implementar un robot humanoide para terapia de niños con trastorno del espectro autista.

1.3.2 Objetivo específico

- Diseñar un robot humanoide de bajo costo, para terapia de niños con TEA.
- Desarrollar algoritmos de visión artificial y aplicarlos en robots humanoides para terapia de niños con TEA.
- Diseñar un prototipo a escala para demostrar la funcionalidad del sistema propuesto.

1.4 Justificación e importancia

Este estudio aportará información acerca de técnicas y algoritmos de visión artificial, los cuales serán complementados con redes neuronales para el entrenamiento de reconocimiento de personas, envío y conversión de imágenes a señales digitales, además, de agregar nuevos temas de estudios que relacionan la robótica con el tratamiento del trastorno del espectro autista, puesto que, considerando la actualidad y la realidad del país, el cual aún no utiliza muchos métodos de alta tecnología para tratar a sus pacientes, debido a que estos métodos suelen ser costosos y además que en el Perú, no existe abundancia en cuanto a tecnología, y menos en cuanto se refiere a robots o inteligencia artificial.

Por otro lado, el desarrollo de este robot, causará un alto impacto tecnológico, debido al espacio geográfico en el cual es creado, y este puede llegar a ser un gran aporte a la masificación de robots utilizados para tratamientos médicos en el Perú, sin mencionar que el objetivo de este proyecto es también desarrollar este tipo de tecnologías a bajo costo, o en su defecto, un costo moderado.

Los principales beneficiarios de este estudio, serían las personas con TEA, los hospitales, clínicas u ONGs, que tratan el autismo, además este estudio también servirá a estudiantes o interesados en desarrollar robots para tratamientos psicológicos, psiquiátricos, o que necesiten implementar visión artificial a sus proyectos de ingeniería.

CAPÍTULO II: MARCO TEÓRICO

2.1. Antecedentes del problema

2.1.1. Antecedentes Nacionales

Caso 1: Fred (2016), en su tesis titulada: Diseño del sistema de control de un brazo robótico de asistencia a personas discapacitadas, tiene como objetivo diseñar e implementar un sistema de control de un brazo robótico de asistencia, el cual será controlado mediante señales EEG, para lograr dicho control se hizo uso de técnicas orientadas en reducir el grado de fatiga mental en la operación y tener la capacidad de integración a una silla de ruedas automatizada. De acuerdo al autor se entiende que el brazo robótico va ir incrustado en la silla de ruedas y va ser controlado por el paciente mediante movimientos cervicales. Lo cual le va permitir evitar fatigas y malestares al paciente permitiendo controlar al robot mediante señas, esta investigación va a favorecer a pacientes que, postrados con limitaciones de capacidad muscular, esclerosis lateral amiotrofia, lesión de la medula espinal, entre otros.

Caso 2: Según Chávez y Cabrera (2013,), en su tesis titulada: Diseño e implementación de un robot humanoide asistencial controlado por computador para aplicaciones en pacientes parapléjicos, tiene por objetivo diseñar e implementar un robot humanoide asistencial, el control de dicho robot, es por computador, el cual fue aplicado en pacientes parapléjicos. Se entiende que en nuestro país la calidad de vida en las personas cambia precipitadamente y por lo cual la tecnología cumple un rol importante en avances de robots que van a ayudar a las personas con diferentes discapacidades en el tratamiento de su salud. En este caso el paciente va controlar un robot mediante un computador para realizar sus diferentes actividades cotidianas, en los diferentes lugares donde se encuentre, tornándose así de fácil manejo y aportando una mejora en la salud del paciente. Estas

actividades también lo pueden realizar mediante un computador las personas con discapacidad motora, ya que muchas veces no hay personas que les puedan cuidar y hacer sus tareas cotidianas. La mejor opción es el robot humanoide que va mejorar la calidad de vida de las personas parapléjicas y otras discapacidades que poseen las personas.

2.1.2. Antecedentes Internacionales

Caso 3: David y Luis (2014), en su tesis titulada: Diseño e implementación de un prototipo de robot asistente para personas con discapacidad motriz y adultos mayores, basado en inteligencia artificial, señalaron que: “la robótica es una de las ramas de la tecnología que dan pasos cada vez más grandes, tanto a nivel industrial, automotriz y sobre todo al servicio y asistencia de la humanidad” (p.17). Se entiende que en este mundo globalizado ya existen robots que pueden cumplir diferentes actividades cotidianas beneficiando en el tratamiento a miles de personas que sufren diferentes discapacidades y les cuesta realizar sus actividades cotidianas, estos robots solo existen en algunos países debido al gran costo que tienen para ser comercializados en países de bajo desarrollo. En base a la investigación que se planteó en esta tesis se puede concluir que el cuidado del adulto mayor en el Ecuador viene siendo prioridad para mejorar ya que existen pocos centros para la atención de estas personas. Con la construcción de este robot va ser de bastante ayuda para para adultos mayores y personas con diferentes discapacidades.

2.2. Bases teóricas

2.2.1. Autismo

2.2.1.1. Definición

La definición de la terminología “Autismo”, fue insertada en 1913 por el psiquiatra suizo Eugen Bleuler, el cual en su obra *Dementia Praecox or the Group Schizophrenias*, utilizó el vocablo autismo para hacer referencia a uno de los síntomas principales de la esquizofrenia. El término autismo proviene del griego eaftismos, cuyo significado es “encerrado en uno mismo”.

A finales de los 30', se empezó a desatarse el autismo como síndrome, al ser visible un número de niños de ambos sexos que tenían una característica muy diferente y única, la cual para esos años era difícil de tratar, al ser algo totalmente nuevo. En 1943 el estadounidense Leo Kanner, empezó a dar descripciones iniciales del síndrome autista. En su escrito describió 11 casos de los cuales 8 eran niños, indicó que todos ellos presentaban un conjunto de características esenciales comunes, concluyendo que tenían una incapacidad para socializar normalmente. El autor amplía las características de ese rasgo básico con expresiones literales de los padres referidas a sus hijos afectados: "autosuficiente"; "como dentro de una concha"; "más feliz cuando se lo deja solo"; "comportándose como si la gente no estuviera presente".

La principal deficiencia, radica en la enorme dificultad para socializar o establecer relaciones interpersonales, en el estudio de Leo Kanner se da mucha relevancia a las disfunciones del lenguaje, sin embargo de los 11 niños, 8 de ellos lograron desarrollar el lenguaje hablado, pero en todos los casos no se logró el carácter comunicativo, mas bien se caracterizaron por carácter memorístico y repetitivo, entre ellos la ecolalia. Desde los estudios de Kanner hasta los actuales, se hace mención que la ecolalia se utiliza como método de terapia, siendo muy eficiente este por ser una forma de comunicación para las personas con TEA, es por tal motivo que este proyecto será dotado con características de ecolalia para el tratamiento y funcionamiento que dará el robot.



Figura 01. Representación de un niño autista

Fuente. OMS

2.2.1.2. Causas

El Autismo como tal no tiene causas definidas, generalmente las causas son desconocidas, en la mayoría de casos se sospecha que se da por las mutaciones o cambios en los genes, ya que no todos los genes que desarrollan esta condición han sido identificados, los estudios de las causas del autismo han dado como resultado lo siguiente:

Se pudo demostrar que los agentes genéticos son base fundamental para la aparición del trastorno del espectro autista. Se estudió casos clínicos de mellizos, en el cual se observó que cuando se detecta el TEA en uno de los mellizos, existe una probabilidad muy alta (90%), de que el otro mellizo también desarrolle el trastorno, en los casos familiares existe 5% de posibilidad de aparición en el neonato, en el caso de tener un hermano con TEA, esta posibilidad es muy alta en comparación de que no existan hermanos con el trastorno autista

2.2.1.3. Tratamiento

En la actualidad el trastorno del espectro autista no tiene cura, bajo ningún tipo de método, ni tampoco es posible tratar los síntomas principales, sin embargo, existen medicamentos que pueden ayudar a controlar los niveles de energía, la concentración. Las convulsiones y la depresión.

Cada cuerpo reacciona diferente es por ello que es evidente que los medicamentos no actúen en los niños de la misma manera, el tomar medicamentos debe ser un proceso controlado por un personal médico y vigilar los avances que estos causan en los niños, además estos medicamentos pueden generar efectos secundarios en los menores, así que existen casos, en los que no es posible aplicarles medicamentos.

Los estudios e investigaciones muestran que los servicios de tratamiento de intervención temprana, pueden llegar a mejorar el desarrollo del niño, Los servicios de intervención temprana se aplican desde el nacimiento hasta los 36 meses, en ese tiempo aprenden destrezas importantes. El tratamiento de síntomas específicos, así como el del habla para casos de retrasos en el lenguaje, este estudio no requiere que se diagnostique formalmente un TEA en el niño, la intervención temprana es de suma importancia ya que este ayudará con el desarrollo del niño, además de que evitará que el nivel de autismo no aumente.

2.2.1.4. Tipos de tratamiento

El tratamiento del trastorno del espectro autista o TEA, es sumamente variado, existen muchos tratamientos, entre ellos: la terapia en base a vitaminas, la terapia antilevadura, el entrenamiento auditivo y el de pruebas discretas, existen también otros métodos terapéuticos como la musicoterapia, la fisioterapia, la integración sensorial, entre otros. Por lo general los tratamientos se dividen en enfoques, los cuales se explicarán brevemente en este proyecto.

El enfoque DIR o enfoque basado en el desarrollo, las diferencias individuales y las relaciones interpersonales, este enfoque se centra en trabajar el estado emocional y las relaciones, además se trabaja en la forma en la que el niño percibe los sonidos y olores.

El TEACCH por sus siglas en inglés, es un método que hace uso de la visión para la enseñanza de destrezas, en este método se hace uso de las tarjetas con imágenes, las pantallas o dibujos, estos causan estímulos en los niños, además de que sirven como enseñanza, para vestirse, comer, entre otras actividades.

Entre las terapias mas comunes tenemos la terapia ocupacional, la cual enseña destrezas de vida, se hace con la intención de independizar al niño con hábitos comunes de vida. La terapia de integración sensorial se utiliza para enseñar a lidiar con información de todos los sentidos de una persona, esto ayuda a que los sonidos no sean molestos para los niños, ni tampoco molestarse al ser tocado. La terapia del habla es un tipo de terapia que trabaja en las habilidades comunicativas del niño, en ocasiones donde el niño no logra expresarse verbalmente se desarrolla otro tipo de comunicación como por medio de ademanes, imágenes o paneles.

El sistema PECS por sus siglas en inglés es un sistema de comunicación por intercambio de imágenes, en este sistema se hace uso de imágenes, dibujos, símbolos, entre otros, el método se aplica mediante preguntas y respuestas con imágenes, es una forma de comunicación en la que el niño se siente a gusto ya que no se interrumpe la paz en la que él vive, además de que aprende un sistema de comunicación, es por este método donde los niños aprenden rápido una forma de comunicación mas acorde a su condición, se utiliza normalmente en casos de niños que no logran desarrollar la comunicación verbal.



Figura 02. Terapia de imágenes

Fuente. <https://www.lavanguardia.com/tecnologia/20160702/402901418986/aplicaciones-tratamiento-autismo-infantil.html>

2.2.1.5. Terapia con robots

Una mezcla de conocimientos en terapia de conducta e ingeniería robótica son los ingredientes de este revolucionario tratamiento para niños con dicha condición.

En la actualidad existen estos prototipos, uno de los más conocidos es de la Universidad Miguel Hernández (UMH) de Elche – España, su robot es AISOY y esta siendo utilizado para pruebas en el tratamiento de niños autistas.

La finalidad de dar terapia con un robot es para estimular al niño de poder interactuar con sus juguetes, es una fusión de terapia ya que el robot se vuelve el terapeuta y al mismo tiempo una herramienta de terapia.

Las cualidades con las que el robot debe estar dotado para dar tratamiento es la de combinar los diferentes tipos de terapia nombrados en el inciso 2.2.1.4, como por ejemplo tener ecolalia, es decir repetir lo que el niño dice, así mismo poder mostrar imágenes y tener algún tipo de comunicación por imagen, es decir un sistema PECS, además deberá ser capaz de reconocer con que niño esta trabajando.

El robot puede tener cualidades extras como la de humanoide, animal, o tipo rueda, es indistinto mientras cumpla su función principal de ser revolucionario a comparación de cualquier método o terapia actual.



Figura 03. Robot ayudante de terapia autista.

Fuente. Universidad Miguel Hernandez.

2.2.2. Visión Artificial

2.2.2.1. Definición

La visión artificial, es un campo de estudio relativamente nuevo, al cual muchos autores lo suelen relacionar con la robótica, automatización, y otras ramas afines, por talo motivo no existe unanimidad en cuanto a la definición de dicho término, García, I.D (2008) habla de la visión artificial como una disciplina compleja que involucra a otras ciencias e incluye estudios de ingeniería, que en conjunto se logra el continuo desarrollo de algoritmos, funciones y aplicaciones, que hacen de la visión artificial una tecnología óptima en todo área.

Entonces la visión artificial es una disciplina que intenta emular las habilidades de visión que tienen los seres vivos, la cual trata de ver una escena, entenderla, interpretarla y ejecutar una acción en base a ello. La Visión artificial como una rama de la inteligencia artificial ha evolucionado mucho en las últimas décadas



Figura 04. Representación gráfica de visión artificial.

Fuente. Sistemas de visión artificial y automatismos.

2.2.2.2. Procesamiento de imágenes

El procesamiento digital de imágenes es un conjunto de métodos, cuya finalidad es la de mejorar la calidad de la imagen, la adquisición de la imagen se puede generar de muchas maneras, las más relevantes son por medios ópticos y medios digitales, para entender el fundamento del procesamiento de imágenes, iremos a la teoría del matemático Joseph Fourier quien afirma “que una gráfica o función, cualquiera que sea su forma, se puede representar con alta precisión dentro de un intervalo dado, mediante la suma de una gran cantidad de funciones senoidales, con diferentes frecuencias”, con ello es posible afirmar que al atenuar algún componente senoidal, tendrá efecto en la calidad de la imagen sea positivo o negativo.

El procesamiento digital de imágenes consta en dividir una imagen en píxeles, es decir se crea un arreglo gráfico rectangular de píxeles, lo siguiente a ello es dar un valor numérico en relación a la luminosidad de cada pixel o elemento, entonces la imagen queda definida por la coordenada y el valor asignado a pixel, todo esto es almacenado por el sistema de control. Finalmente se suele alterar los valores de luminosidad hasta conseguir una calidad mejor a la adquirida inicialmente.

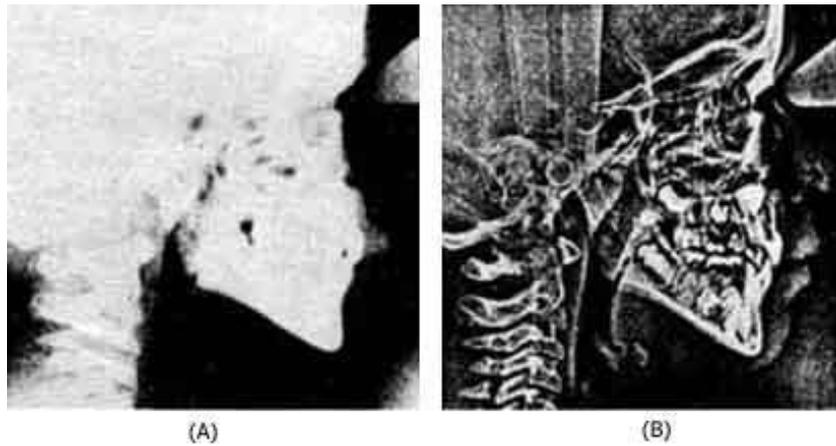


Figura 05. Ejemplo de procesamiento de imágenes digitales.

Fuente. ILCE.

2.2.2.3. Redes neuronales aplicado a la visión artificial

Matich, D. J (2001), expresa que una red neuronal es una forma computacional moderna, que está inspirada en modelos biológicos, y que se compone por un modelo matemático que abarca un gran número de proceso organizados en niveles. De la premisa dada por Matich, se deduce que una neuronal es un modelo matemático computacional formado por algoritmos que tratan de emular el funcionamiento de un cerebro humano. El principio de funcionamiento de una red neuronal artificial basa en un conjunto de neuronas que componen una red, y esta es entrenada en cada iteración, ajustando de manera autónoma los pesos dados a cada neurona.

Para entrenar un red neuronal y aplicarlo en el campo de la visión artificial, se requiere una base de datos, ya sea que le sistema lo obtiene por si mismo al ejecutarse o previo al entrenamiento, en ambos casos entrena cada pixel de una imagen, y por cada iteración que se hace en la red, esta compara a la base de datos, cuando se consiga cierta similitud en características y parámetros propios de la imagen se pasa a la etapa de convalidación y verificación, para en base a ello el sistema tome una decisión en función a lo programado.

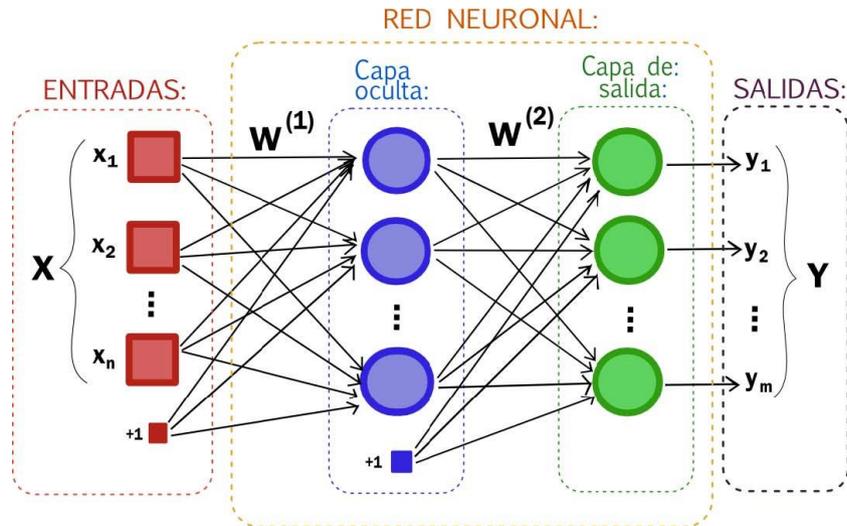


Figura 06. Red neuronal.

Fuente. <https://imscablog.wordpress.com/2017/05/10/redes-neuronales/>

2.2.3. Control de robots

2.2.3.1. Sensores

Un sensor es un dispositivo que tiene la propiedad de medir magnitudes y variables, estas pueden ser modificables por los cambios del ambiente, como la temperatura, flujo de viento, entre otros.

Un sensor es el encargado de recolectar datos numéricos de la variable, estas pueden ser manipuladas o controladas, generalmente un sensor va junto a un transductor, estos sirven para transformar la información de los sensores, las variables pueden ser de cambio rápido, como el movimiento, la velocidad, la masa, el volumen, entre otros, y las variables de cambio lento o las de capacidad, son las que demoran en cambiar las magnitudes de la variable, como la temperatura, la capacidad, etc.

La diferencia de un sensor con un transductor radica en el tiempo de contacto, es decir un sensor siempre trabaja con la variable a medir, sin embargo, el transductor se encarga de transformar la información del sensor, para poder leer los datos recolectados.



Figura 07. Sensores Industriales.

Fuente. Automatización Industrial 2009.

2.2.3.2. Actuadores

Un actuador es un dispositivo que utiliza y transforma todo tipo de energía, como la energía hidráulica, eólica, eléctrica, solar, entre otras, para luego ser aplicado en distintos procesos, para ser activados normalmente se utiliza un controlador, este envía señales de control para activar a los actuadores, estos actuadores modifican su magnitud según el tipo de señal de entrada.

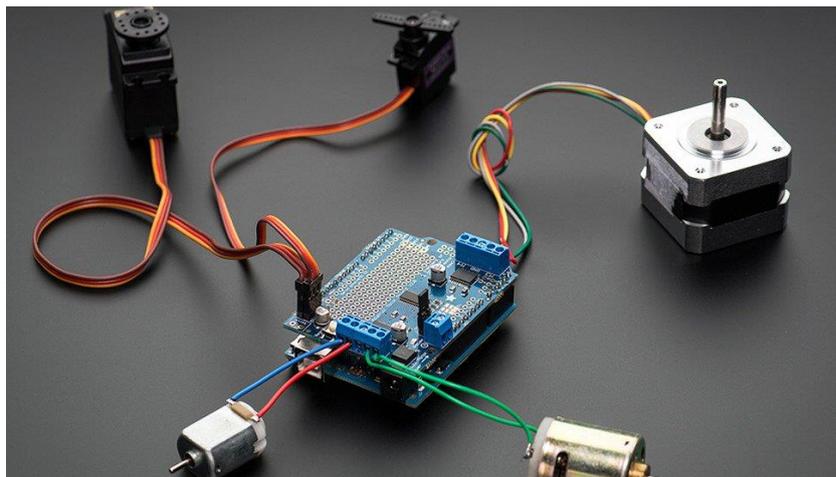


Figura 08. Actuadores eléctricos.

Fuente. Introducción al IDE Arduino.

2.2.4. Telemetría

La telemetría es una tecnología que permite trabajar a distancia, actualmente se hace uso de la telemetría como una nueva técnica de control, donde el tipo de comunicación entre la estación (centro de control) y el sistema de trabajo, se hace mediante conexión inalámbrica, la comunicación a cortas distancias se hace por medio de Wifi, Bluetooth, infrarrojo, entre otros, incluso se suele utilizar cableado fino, a distancias largas se suele hacer uso de redes como internet, ethernet, entre otros.

La telemetría inició en el siglo XX, inicialmente se utilizó para distribuir energía eléctrica, según fue pasando el tiempo, esta tecnología se fue diversificando, y empezó a ser aplicada en vigilancia de líneas telefónicas, en 1930 inició la aeronáutica gracias a la telemetría, ya que se utilizó para controlar las condiciones atmosféricas, en la actualidad se utiliza mucho la telemetría en sectores de riesgo humano, como la minería, la química, la energía, transporte, líneas de alta tensión, entre otros, económicamente estos sectores perciben ganancias debido al coste que se gana, haciendo uso de este tipo de tecnologías.

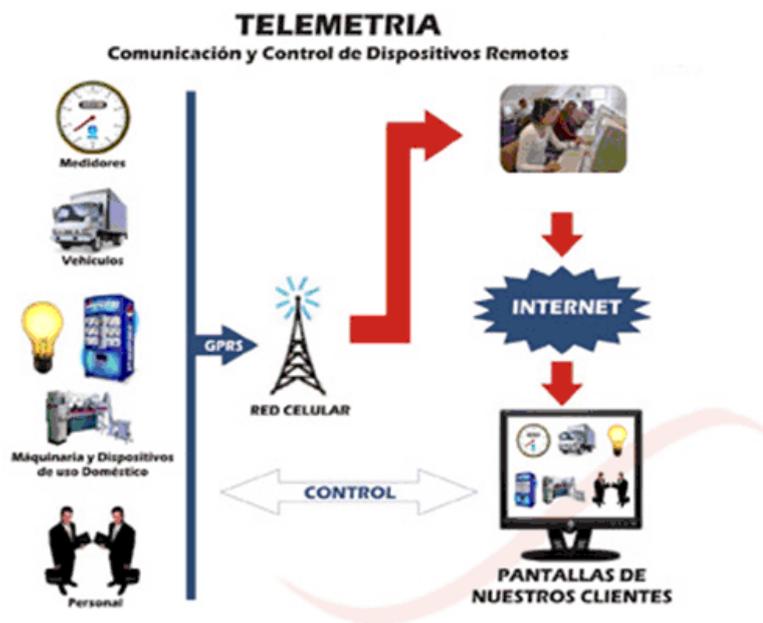


Figura 09. Telemetría.

Fuente. Electricad.

2.3. Definición de términos básicos

2.3.1. Robot Humanoide

El avance la tecnología ha avanzado de tal forma, que en la actualidad existen robots capaces de simular las actividades cotidianas del ser humano, así como imitar sus actos, su forma de caminar, entre otras cualidades, a este tipo de robots se les dio el nombre de robots humanoides.

Los robots humanoides tienen como fin desarrollar las actividades humanas, y de ser necesario, mejorar la calidad del servicio, hoy en día, se viene estudiando la locomoción bípeda en robots humanoides, según la cualidad por la que fue creado cada robot, puede cambiar sus rasgos físicos, la mayoría de robots humanoides, solo representa la habilidad física humana, de la cintura para arriba, como los robots ara usos terapéuticos, o los robots de réplica muscular, que solo toman la forma física del cuello para abajo.

2.3.2. Inteligencia artificial

La AI (Artificial Intelligence), es una tecnología revolucionaria, que en la actualidad se esta explotando mediante programación, las máquinas inteligentes son agentes racionales flexibles, la inteligencia artificial es un campo que aún no se ha estudiado a fondo, es una rama en crecimiento con la cual, se ha logrado granes avances tecnológicos, la informática y programación son base en la inteligencia artificial, en este proyecto se explotará un campo de la inteligencia artificial, solo se abarcará la visión artificial, la cual se realizará mediante redes neuronales.

2.3.3. Autoaprendizaje

En sistemas dotados de inteligencia artificial, el autoaprendizaje es un término muy utilizado para referenciar que dicho sistema es de lazo cerrado y además aprende por cada proceso superado, es decir mejora su eficacia en cada iteración.

CAPÍTULO III: METODOLOGÍA

3.1. Introducción

Esta investigación será guiada por la metodología VDI 2221, la cual es denominada como el enfoque sistemático para el diseño de sistemas técnicos y productos, esta norma fue desarrollada por ingenieros alemanes, esta norma primero indica establecer la necesidad, el cual será plasmado en una lista de exigencias, donde se distinguirá los deseos y necesidades por cada fase y dominio del proyecto, seguido a ello se debe estructurar la planificación, definición del producto y el diseño conceptual, estos tres serán tratados en el inciso 3.3, el cual se denomina “Caja Negra”, luego se elaborará la estructura de funciones, en base a ello crear la matriz morfológica o matriz de soluciones, finalmente se hará un estudio técnico – económico de las soluciones de la matriz.



Figura 10. Estructura de metodología según Ullman.

Fuente. Elaboración propia según Ullman.

3.2. Lista de exigencias

A continuación, se presenta la lista de exigencias por cada dominio y una de interés general del robot.

Tabla 1. Lista de exigencias generales

LISTA DE EXIGENCIA – GENERAL		
Categoría	Exigencia (E) / Deseo (D)	Descripción
Función principal	E	El robot será de gran ayuda en el tratamiento de niños autistas.
Función específica	E	El robot podrá trasladado de un sitio a otro.
Geometría	E	El robot será de tamaño pequeño
Masa	D	Se buscará que el robot sea liviano y no pase de 10 Kg.

Costo	E	El costo del robot deberá ser lo menos costoso posible.
-------	---	---

Fuente. Elaboración propia.

Tabla 2. *Lista de exigencias dominio electrónico*

LISTA DE EXIGENCIA – ELECTRÓNICA		
Categoría	Exigencia (E) / Deseo (D)	Descripción
Función principal	E	El sistema electrónico permitirá la programación, control y recepción de datos que permitan su funcionamiento.
Función específica	E	Se hará uso de una cámara y sensores de proximidad.
Compatibilidad	E	Los sensores y actuadores del robot deberán ser compatibles al sistema de control.
Comunicación	E	La cámara y los sensores deben de comunicarse con el sistema de control.
Geometría	E	Las dimensiones de la tarjeta deberán ser las mínimas posibles
Energía	E	El robot deberá ser alimentado por baterías DC.
Señales	E	Las señales de entrada al sistema deberán ser señales sin ruido.
Seguridad	E	Aislamiento entre etapas.
Costo	E	Los componentes electrónicos deberán ser lo menos costoso posible.

Fuente. Elaboración propia

Tabla 3. *Lista de exigencias dominio de control*

LISTA DE EXIGENCIA – CONTROL		
Categoría	Exigencia (E) / Deseo (D)	Descripción
Función principal	E	Controlar los observadores para eliminar ruidos y perturbaciones del robot.
Función	E	Desarrollar algoritmos de visión artificial con redes

principal		neuronales.
Función principal	E	El robot tendrá control telemétrico mediante un aplicativo móvil.
Función específica	E	Dividir en subsistemas y hacer independiente a cada uno de ellos.
Comunicación	D	La comunicación entre la app y el controlador será bajos protocolos Bluetooth.
Plataforma	E	El sistema será desarrollado en un ordenador
Seguridad	E	El código será almacenado en una base datos confiable.
Mantenimiento	E	La programación será documentada.

Fuente. Elaboración propia

Tabla 4. *Lista de exigencias dominio mecánico*

LISTA DE EXIGENCIA – MECÁNICA		
Categoría	Exigencia (E) / Deseo (D)	Descripción
Función principal	E	Los brazos del robot serán de un grado de libertad, y podrá deslizarse por el suelo.
Función específica	E	Se debe garantizar impermeabilidad total interna del robot
Montaje	D	El robot será de fácil montaje y desmontaje.
Fabricación	D	Los componentes utilizados serán fabricados en el país.
Material	E	El material del robot no deberá ser tóxico, ni tampoco será pintado con aerosoles que pueden dañar la salud del niño.

Fuente. Elaboración propi

3.3. Caja negra

A continuación, se presenta un esquema de función total del robot, este esquema representa la funcionalidad total del sistema en forma concreta especificando sus entradas y sus salidas.



Figura 11. Estructura de metodología según Ullman.

Fuente. Elaboración propia.

3.3.1. Entradas

Las entradas son los datos, señales o información que el robot recibe como fuente principal, y es lo que inicia el funcionamiento de este, sin estas entradas, el robot podrá estar encendido, pero sin funcionar, o simplemente estar apagado.

3.3.1.1. Voz del niño

Una forma de que el robot entre en modo de funcionamiento es mediante la voz del niño, esto se logrará mediante un parlante o altavoz conectado al robot, para que este una vez escuchado la voz del niño puede ejecutar una orden, dependiendo cual sea.

3.3.1.2. Orden desde el aplicativo móvil

Otro método para interactuar con el robot es mediante un aplicativo móvil, ya que como se sabe el niño suele ser bastante huraño para conversar, entonces se crea esta alternativa de comunicación no verbal robot – usuario. Sin embargo, bajo esta alternativa existe la posibilidad de comunicación verbal solo por parte del robot.

3.3.1.3. Orden desde la pantalla touch

De forma bastante similar al aplicativo móvil, se puede enviar ordenes desde la pantalla touch, pero la diferencia radica en que este medio, obliga al contacto físico robot – usuario, lo cual puede seguir siendo no verbal, o también verbal solo por parte del verbal, pero este medio de comunicación se creó más para mostrar figuras amigables y entretenidas.

3.3.2. Salidas

Todos los datos de entrada del sistema serán procesados y en base a ello emitir señales a los actuadores del robot, para que estos cumplan con la función encomendada.

3.3.2.1. Reproducir mismo audio

Según el Dr. Bratch (2017), indicó que, para el tratamiento del autismo en niños, es muy importante que ellos escuchen su propia voz, sin importar mucho lo que se diga, en función a ello, el robot estará en la capacidad de poder imitar lo que el niño diga y almacenar ese audio en una base de datos.

Esta función se hará cuando el niño o terapeuta lo considere necesario, para ello bastará que el robot este encendido.

3.3.2.2. Mover articulaciones del robot

Esta función se ejecutará cuando se envíe una señal al robot, ya sea desde la pantalla touch, o desde la app, para ambos controles existirá un comando específico para esta función.

3.3.2.3. Comunicación verbal con el niño

El robot estará dotado de una amplia base de datos para poder mantener una conversación parametrizada con el niño, esta comunicación se logrará solo cuando el niño así lo disponga, y por medio de la app.

3.4. Estructura de funciones

En este inciso se desglosará lo mostrado en la caja negra, con este método lo que se lograr es dar funciones específicas a cada dominio, y en base a ello poder hacer un buen armado del robot.

3.4.1. Estructura de funciones por dominio

Para este proyecto la funcionalidad del robot se ha dividido en 3 dominios, a continuación, se mostrará las entradas y salidas principales por cada dominio.

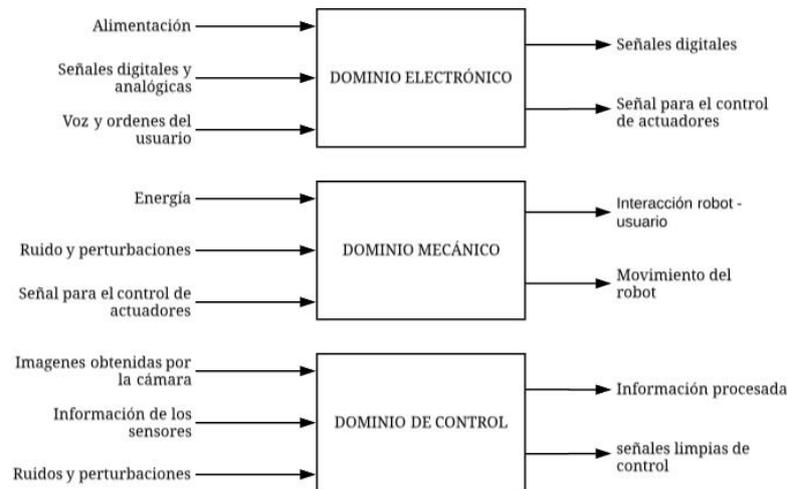


Figura 12. Estructura de funciones por dominio.

Fuente. Elaboración propia.

En concordancia a la figura 12, en la cual se ha estructurado cada dominio, a continuación, se mostrará literalmente las funciones que realizará cada dominio.

3.4.1.1. Dominio electrónico

Este dominio consta de 7 ciclos: el primero de ellos será el encargado de **energizar** al sistema y **regular** la tensión y corriente que afectaran al robot; el segundo ciclo será el encargado de **inducir** la tensión al controlador; el tercer ciclo será el encargado de **sensar** y **detectar** la presencia del niño; el cuarto ciclo será el encargado de **comunicar** el controlador con la app; el quinto ciclo, será el encargado de **recibir** y **procesar** lo que el usuario diga; el sexto ciclo será para **recibir** y **replicar** el audio escuchado; el séptimo ciclo se encargará de **amplificar** el audio enviado desde el control; por último se **mostrará** imágenes en una pantalla touch.

3.4.1.2. Dominio de control

Este dominio consta de 3 ciclos: el primer ciclo se encargará de **adquirir**, **procesar** la imagen y **enviar** la señal al controlador; el segundo ciclo será el encargado de **controlar** el aplicativo móvil; el tercer ciclo será el encargado de **administrar** las señales de los sensores y módulos del robot, para enviar señales de control a los actuadores.

3.4.1.3. Dominio mecánico

En este dominio es en el que se realiza las acciones enviadas desde el controlador, este consta de 2 etapas y 5 ciclos: en la primera etapa se **diseñará** la estructura física del robot, en la segunda etapa se **fabricará** según el diseño dado; el primer ciclo de este dominio será el encargado de **mover** los brazos del robot; el segundo ciclo es el **transportador** del robot; el tercer ciclo **direccionará** el movimiento del robot; el cuarto ciclo será el **regulador** de velocidad del robot; el último ciclo será el que se encargue de **rotar** el ángulo de la cámara.

3.4.2. Caja de cristal

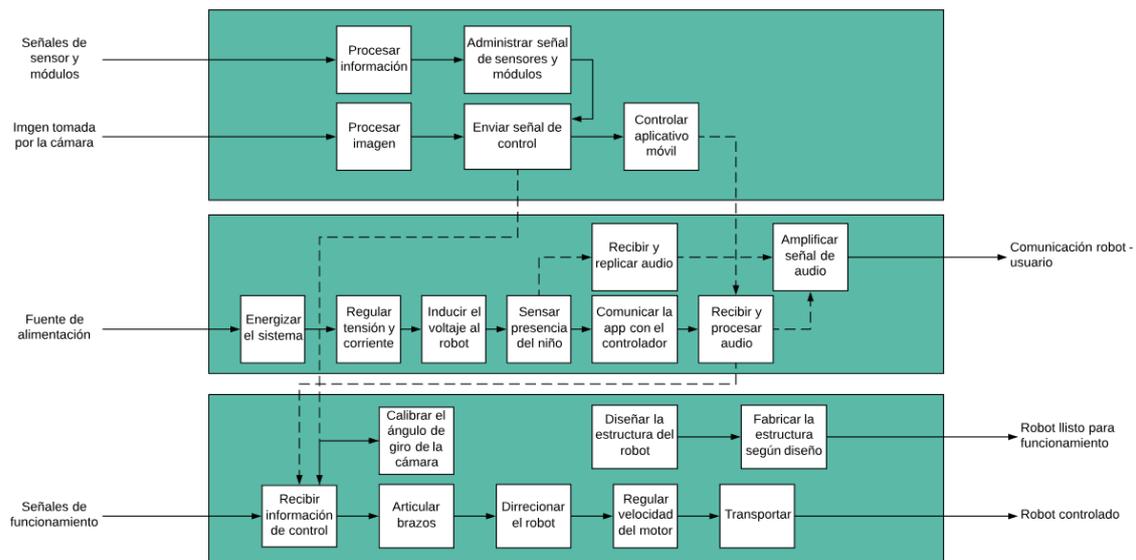


Figura 13. Caja de cristal del sistema

Fuente. Elaboración propia.

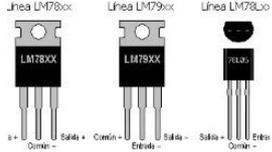
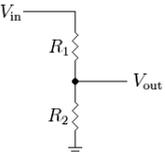
3.5. Matriz morfológica

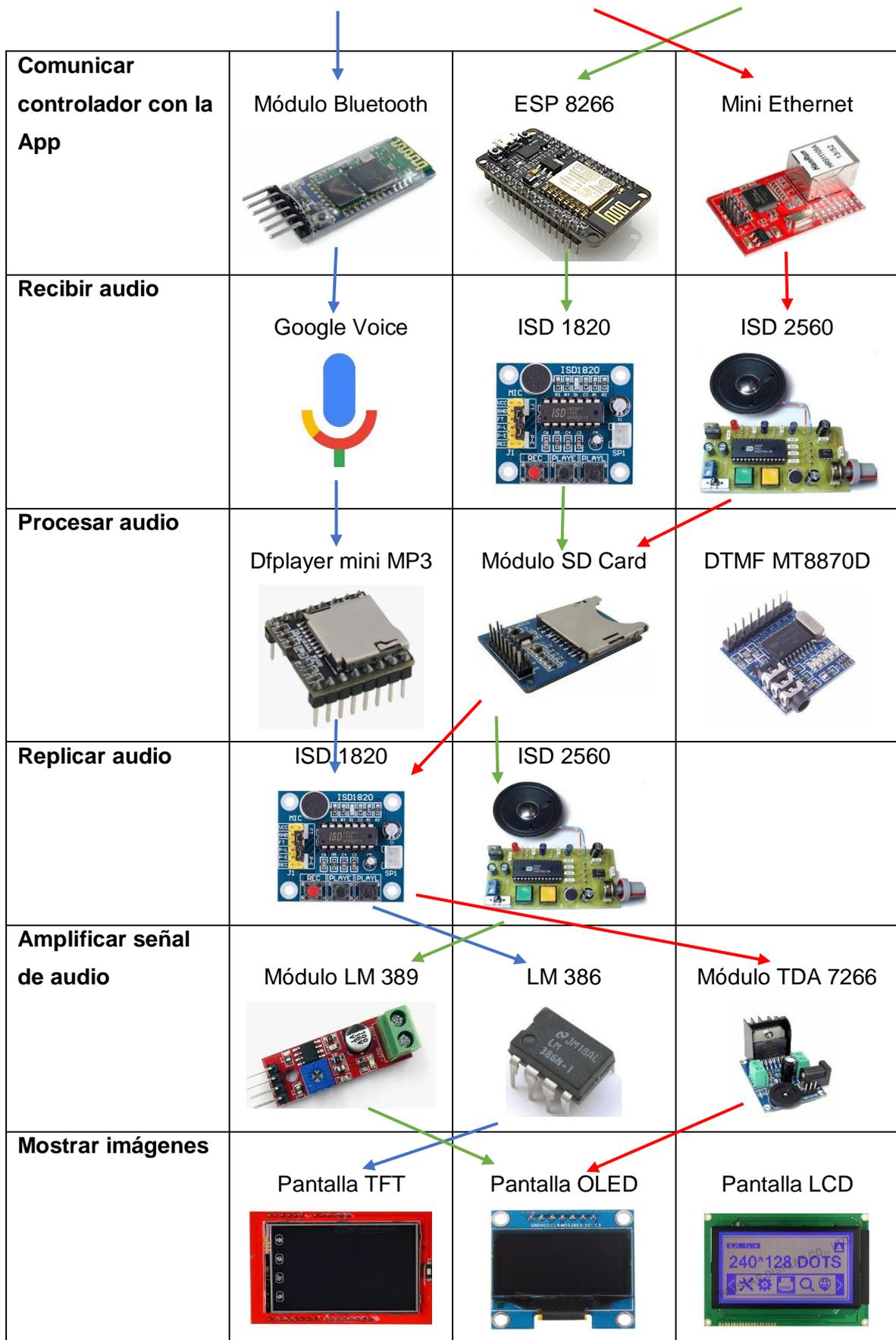
A continuación, se presenta una matriz morfológica por cada dominio, con este método se pretende ampliar las alternativas de solución para el problema planteado en la presente investigación.

3.5.1. Matriz morfológica de dominio electrónico

En la siguiente tabla se presentará las diferentes soluciones lógicas para el dominio electrónico del robot, las soluciones planteadas son las más utilizadas en este tipo de proyectos, es por eso que han sido consideradas en la siguiente matriz.

Tabla 5. Matriz morfológica del dominio electrónico

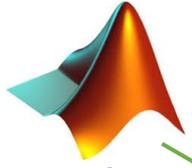
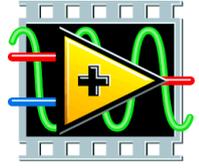
FUNCIONES PARCIALES	PORTADORES DE FUNCIONES		
	Alternativa 1	Alternativa 2	Alternativa 3
Energizar el sistema	220 V AC 	Batería 	Batería recargable 
Regular la tensión	Convertor AC/DC 	Regulador LM78XX 	Divisor de tensión 
Inducir voltaje al robot	Interruptor ON/OFF 	Interruptor I/O 	Pulsador 
Sensor presencia	Ultrasónico 	Sensor Inductivo 	Sensor infrarrojo 



Fuente. Elaboración propia

3.5.2. Matriz morfológica de dominio de control

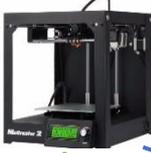
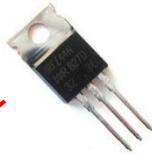
Tabla 6. Matriz morfológica del dominio de control

FUNCIONES PARCIALES	PORTADORES DE FUNCIONES		
	Alternativa 1	Alternativa 2	Alternativa 3
Adquirir imagen	Cámara WEB 	VGA OV 7670 	PIXY CAM 
Procesar imagen	Matlab 	C++ 	Labview 
Enviar señal de control	Arduino nano 	Arduino Uno 	PIC 
Controlar aplicativo móvil	MIT AI2 	Android Studio 	Java 
Administrar señales de control	Arduino Uno 	Arduino Mega 	Raspberry PI 

Fuente. Elaboración propia

3.5.3. Matriz morfológica de dominio mecánico

Tabla 7. Matriz morfológica del dominio mecánico

FUNCIONES PARCIALES	PORTADORES DE FUNCIONES		
	Alternativa 1	Alternativa 2	Alternativa 3
Diseñar la estructura del robot	SolidWorks 	AutoCAD 	Inventor 
Fabricar la estructura del robot	Impresora 3D 	CNC 	Impresora Metal X 
Mover las articulaciones y movilizar al robot	Motorreductor 	Motor paso a paso 	Motor DC 
Regulador de velocidad del robot	L298N 	IRFZ44N 	ULN2003 
Direccionar ángulo de giro del robot	Encoder 	Potenciómetro 	Eslabón 

Rotar base de la cámara	 <p>Micro Servo SG90</p>	 <p>Servomotor MG995</p>	 <p>Servomotor MG996</p>
-------------------------	---	--	---

Fuente. Elaboración propia

3.6. Estudio técnico

A continuación, se analizará cada alternativa de solución bajo criterios de funcionamiento, dándole a cada uno un valor numérico en rango de 1 a 4, esta numeración lo establece la Universidad Católica del Perú PUCP, ya que permite un valorar cada solución en un rango corto sin entrar a mucho detalle numérico, posteriormente se detallará para cada dominio las soluciones obtenidas, y cuál es la óptima.

3.6.1. Concepto de solución de dominio de electrónica

Tabla 8. *Concepto de solución del dominio de electrónica.*

CARACTERÍSTICAS TÉCNICAS Y ECONÓMICAS	SOLUCIONES			
	S1	S2	S3	IDEAL
Funciones	3	3	4	4
Rapidez	3	2	3	4
Tiempo muerto	3	3	2	4
Calidad de trabajo	2	3	3	4
Facilidad de montaje	4	4	2	4
Facilidad de mantenimiento	4	4	2	4
Consumo de energía	2	3	3	4
Productividad	4	4	4	4
Costo de tecnología	2	3	1	4
Costo de operación	2	3	2	4
TOTAL	29	32	26	40

Fuente. Elaboración propia.

La solución más óptima es la solución 2, para este dominio se plantea alimentar el robot con baterías recargables, ya que al ser un prototipo robótico que trabajará con varios motores, se intuye que el consumo de voltaje es alto, y trabajarlo con baterías no recargables, el costo que generaría el robot sería muy alto, y como se mencionó en la lista de exigencias el robot debería de ser de fácil traslado, así que no es buena opción trabajarlo con alimentación cableada, en cuanto a la regulación del voltaje en esta solución se optó por utilizar los reguladores de la gamma LM 78 – XX, obviamente este iría acompañado de un circuito para proteger al regulador de picos altos de tensión, el robot tendrá un interruptor ON / OFF para cambiar el estado de funcionamiento, en cuanto a la detección de presencia se hará uso de sensores ultrasónicos, estos son de fácil manejo y son muy eficientes para distancias cortas, la comunicación del controlador con la app será por medio de bluetooth, debido a que el manejo del robot será a cortas distancias no es necesario optar por una comunicación como el bluetooth o redes industriales, ya que eso genera un costo de materiales más alto e innecesario, la obtención de la voz será por medio del software libre Google voice, este es compatible con la mayoría de plataformas de programación de apps, así que este no representará mayor esfuerzo de programación, el procesamiento y almacenamiento de audios estará a cargo de un módulo de reproducción DFPlayer Mini MP3, este en comparación a las otras soluciones tiene librerías y comandos muy útiles específicos para lo que son audios o grabaciones, el módulo SD Card también tiene librerías similares, sin embargo ese módulo es usado más para almacenamiento de datos o información de sensores, para la grabación y réplica del audio se hará uso de un módulo ISD 1820, este módulo es de los pocos que ofrece este beneficio a un bajo costo, ya que normalmente los módulos padres como el ISD 2560 o el ISD 2562, tienen ciertos beneficios pero son mucho más costosos, en el mercado actual esos módulos no bajan de los 100 soles, sin embargo el ISD 1820 se puede encontrar a un precio promedio de 30 soles y en línea hasta los 3 dólares americanos, la amplificación de señal se hará mediante un integrado TDA 2050, este integrado no va solo para amplificar señal, se necesita hacer un circuito externo, sin embargo su amplificación es más limpia, los filtros utilizados con este integrado ayudan a depurar la señal de amplificación, los módulos de las otras soluciones si bien ya no es necesario un circuito externo, sus costos suelen ser mayores, y en algunos si es necesario colocar filtros en su salida, para finalizar este dominio, se hará un sub control mediante una pantalla touch, la cual será una pantalla TFT de 2.4 pulgadas, se eligió este módulo debido a su simplicidad de programación y su compatibilidad con Arduino Uno y Arduino Mega.

3.6.2. Concepto de solución de dominio de control

Tabla 9. Concepto de solución de dominio de control.

CARACTERÍSTICAS TÉCNICAS Y ECONÓMICAS	SOLUCIONES			
	S1	S2	S3	IDEAL
Funciones	2	3	3	4
Rapidez	2	3	3	4
Tiempo muerto	2	3	3	4
Calidad de trabajo	2	3	3	4
Facilidad de montaje	4	4	3	4
Facilidad de mantenimiento	4	4	3	4
Consumo de energía	3	3	3	4
Productividad	2	3	4	4
Costo de tecnología	4	3	1	4
Costo de operación	4	3	2	4
TOTAL	29	32	28	40

Fuente. Elaboración propia.

La solución óptima para el dominio de control es la solución 2, entonces este dominio hará uso una cámara WEB para adquirir las imágenes, esta opción es la mas usada en reconocimiento de rostros, entre las soluciones puestas, porque no se eligió la PIXY CAM que es un sensor cámara y es altamente eficiente en visión artificial, la respuesta es simple, la PIXY CAM, es muy buena pero muy costosa casi 8 veces lo que costaría una cámara WEB, y el prototipo busca ser económico, el procesamiento de imágenes será en MATLAB debido a que este software es muy eficiente en cuanto se refiere a inteligencia artificial, el interfaz de comunicación del Matlab – Controlador será por medio de un Arduino nano, el desarrollo de la app se realizará en la plataforma MIT AI2, esta plataforma es de acceso libre y es suficiente para este robot, ya que no se necesita mayor funcionalidad, de ser el caso se hubiese optado por Android Studio, el control del robot, en el cual se unirá todo el código, entonces el cerebro del robot será una placa Arduino Mega.

3.6.3. Concepto de solución del dominio mecánico

Tabla 10. Concepto de solución de dominio mecánico

CARACTERÍSTICAS TÉCNICAS Y ECONÓMICAS	SOLUCIONES			
	S1	S2	S3	IDEAL
Funciones	3	3	3	4
Rapidez	3	2	3	4
Tiempo muerto	3	3	3	4
Calidad de trabajo	3	3	3	4
Facilidad de montaje	2	2	2	4
Facilidad de mantenimiento	2	2	2	4
Consumo de energía	3	2	2	4
Productividad	3	2	3	4
Costo de tecnología	3	3	3	4
Costo de operación	3	3	3	4
TOTAL	28	25	27	40

Fuente. Elaboración propia.

La solución más óptima para el dominio de control es la primera solución, esta establece diseñar el robot en el software SolidWorks, este software es elegido debido a que hay mas conocimiento utilizando esta herramienta, en cuanto a la fabricación se decidió hacer uso de impresora 3D, la cual no tiene un costo de producción muy bajo, pero es económico en comparación a impresoras más sofisticadas como la Metal – X o las impresoras láser, las cuales tienen un acabado más estético pero muy costosos, en cuanto a la movilización del robot y al movimiento de las articulaciones se usarán motorreductores de 9 a 12 VDC, como es solo movimiento no es necesario un control exacto como el obtenido por un motor a pasos, el driver para estos motores será el módulo L298N, como ya se sabe estos módulos pueden controlar hasta dos motores y soportan corrientes altas de trabajo, el disipador es tamaño medio, pero considerando que se trabajará con motorreductores de 9 a 12 VDC, este módulo es más que suficiente, el direccionamiento de las llantas será por medio de un eslabón con engranes conectado a la salida del motor, en este punto dependiendo la eficiencia de este se podrá evaluar si se cambia por un encoder, para la base de la cámara se utilizará un servomotor negro modelo MG996R, debido a que tiene alta fuerza de trabajo.

CAPÍTULO IV: DISEÑO

4.1. Dominio Electrónico – Análisis de planos por etapa

La elaboración de este inciso, se hizo con la referencia de las hojas técnicas de cada componente (datasheet)

4.1.1. Primera etapa

En esta primera etapa se diseñará lo respectivo a energizar, regular e inducir el voltaje, para esto primero se debe de tomar en consideración que esta etapa es para alimentar los 2 arduinos, 2 motores DC, un motorreductor, un servomotor MG996R, entonces en base a sus hojas técnicas se necesita un voltaje promedio de 25V, como se trabajará con baterías recargables y cada una de estas trabajará independiente entonces se propone trabajar con 3 baterías de 9 V a 2000 mA cada una. Para realizar el esquema de conexiones de esta primera etapa, debemos de considerar conexiones de seguridad en caso existan picos altos de voltaje o corriente.

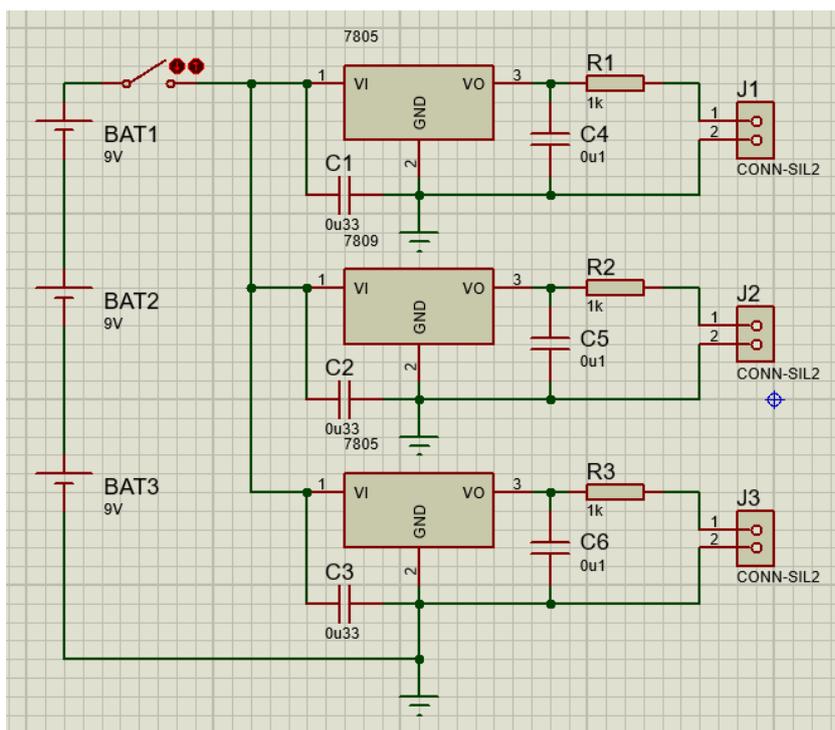


Figura 14. Conexión de las baterías.

Fuente. Elaboración propia.

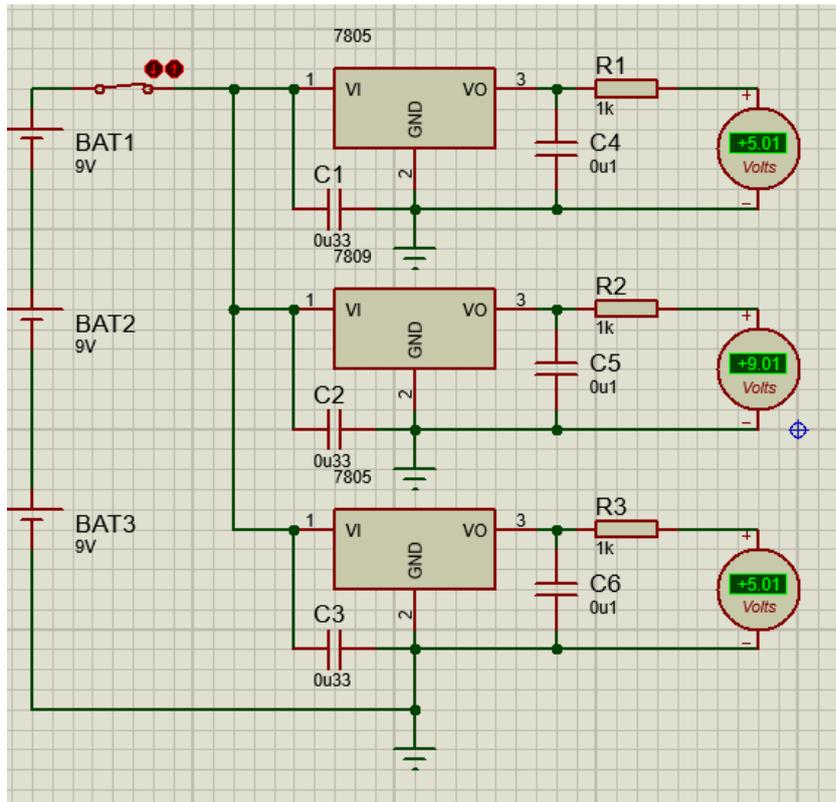


Figura 15. Simulación de alimentación del robot.

Fuente. Elaboración propia.

El circuito basa en limpiar las señales de la batería, para ello se utiliza un condensador cerámico de 0.33uF, el cual va conectado desde la fuente de alimentación, en este caso las 3 baterías de 9V en serie, y en la entrada Vin del regulador LM7805, este primer condensador se utiliza como filtro, ya que evita todo tipo de ruido proveniente de las baterías, y con esto se aseguro que la entrada sea limpia y netamente DC, ya que no se necesita ninguna entrada AC como las ondulaciones o perturbaciones, el trabajo del primer condensador es de un condensador de derivación, ya que corta la señal AC y la envía a tierra, y solo el porcentaje de DC es el que entra al regulador.

Una vez pasado el primer filtro en la entrada, se debe asegurar que la señal brindada del por el regulador también sea una señal limpia, para ello se conecta un condensador cerámico de 0.1uF, el cual va al pin Vout del LM7805, para alimentar un Arduino o cualquier microcontrolador, es necesario que la señal sea limpia, ya que se esta usando un chip lógico, y es crucial que la señal de entrada sea continua, ya que un pico alto podría dañar el controlador, para ello se usa un segundo filtro el cual se corta en tierra, y la señal DC pasa limpia del regulador al Arduino.

4.1.2. Segunda Etapa

Para esta segunda etapa se trabajará las conexiones del Arduino el cual será el cerebro del robot, el esquema de conexiones será mostrado en Fritzing, para la vista de protoboard y en proteus para las conexiones de PCB.

4.1.2.1. Sensor Ultrasónico

El sensor ultrasónico HC-SR04, en la industria de robótica educativa, es uno de los sensores más utilizados y más comerciales, debido a su bajo costo (10 soles), además de su intervalo de detección que va desde los 17,7 mm. hasta los 495 mm., la resolución del sensor de 3mm. El sensor está compuesto dos piezoeléctricos: el primero es un transistor de potencia media y un receptor con similares cualidades, ambos tienen forma cilíndrica, El transmisor está encargado de enviar una señal ultrasónica, esta señal rebota en el objeto a censar, al volver del rebote llega al cilindro receptor. Para lograr la distancia del sensor al objeto, se trabaja por medio del tiempo en que la señal demora en ir y volver del sensor al objeto, es obvio pensar que la velocidad con la cual llegará al receptor está íntimamente relacionada con la distancia de transmisión y rebote.

Matemáticamente la distancia se puede obtener de la siguiente forma:

La velocidad de onda que emite el transmisor es de 340 m/s.

La fórmula lineal de la velocidad es: $v = d/t \dots (1)$

La distancia en este caso es de ida y vuelta, entonces sería $2d$, si lo reemplazamos en (1), la distancia del sensor hacia un objeto sería: $d = 170 \cdot t$, expresado en metros.

Tabla 11. *Parámetros eléctricos de operación del sensor HC – SR04*

Parámetros de operación	Min	Típ.	Máx	Unidad
Voltaje de operación	4.5	5	5.5	V
Corriente	10	15	20	mA

Fuente. Hoja de datos HC – SR04

El ángulo efectivo del sensor es de 15°, es decir tiene una visión total de 30°, a continuación, se muestra el diagrama de conexiones del sensor.

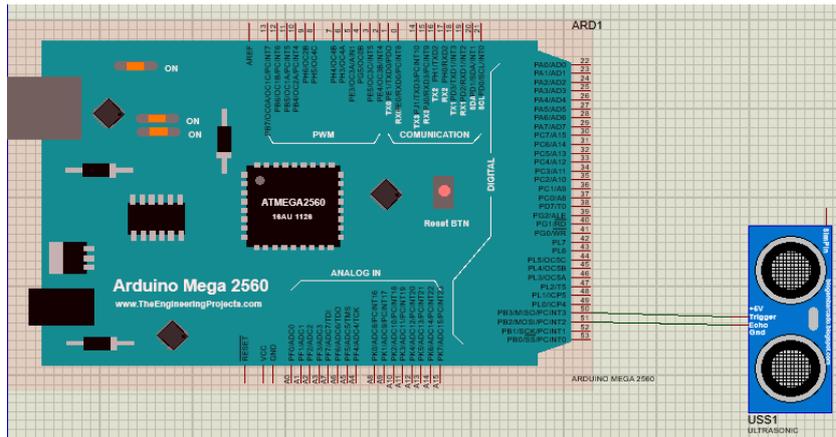


Figura 16. Esquema de conexiones del sensor HC – SR04
Fuente. Elaboración propia

4.1.2.2. Módulo Bluetooth HC 06

Se utilizó el módulo bluetooth HC 06, ya que este puede ser usado como maestro y esclavo, además de que su rango de trabajo es de 18 m, y el robot será controlado por la app, a una distancia máxima de 8 metros, por tal motivo este módulo es suficiente, y no es necesario hacer uso de otro tipo de comunicación.

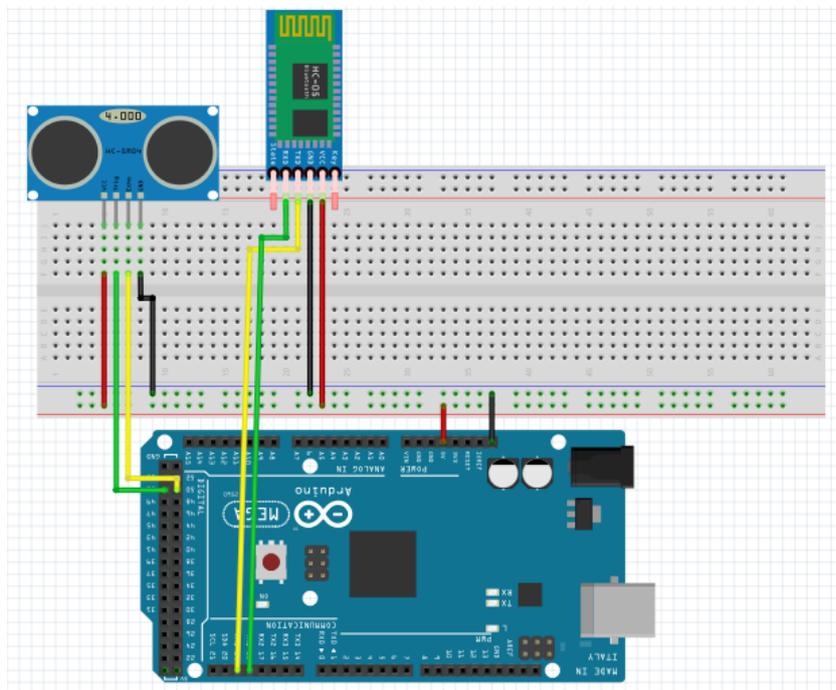


Figura 17. Esquema de conexiones del módulo Bluetooth HC 06, vista de protoboard.
Fuente. Elaboración propia.

4.1.2.3. Módulo DFPlayer mini MP3

El reproductor de MP3 DFPlayer Mini para Arduino es un módulo MP3 de bajo precio y con una salida simplificada directamente al altavoz. El módulo se puede usar como un módulo independiente con batería, altavoz y pulsadores adjuntos, o se puede usar en combinación con un Arduino UNO o cualquier otro con capacidades RX / TX.

Este módulo trabaja con una memoria SD, la cual puede ser de hasta 32GB, internamente al módulo se puede trabajar hasta con 100 carpetas, cada carpeta un máximo de 250 archivos, el módulo permite 30 niveles de audio y trabaja a un voltaje lógico de 5V.

Tabla 12. Especificaciones de los pines de entrada del módulo DFPlayer Mini.

Pin	Description	Note
VCC	Input Voltage	DC3.2~5.0V;Type: DC4.2V
RX	UART serial input	
TX	UART serial output	
DAC_R	Audio output right channel	Drive earphone and amplifier
DAC_L	Audio output left channel	Drive earphone and amplifier
SPK2	Speaker-	Drive speaker less than 3W
GND	Ground	Power GND
SPK1	Speaker+	Drive speaker less than 3W
IO1	Trigger port 1	Short press to play previous (long press to decrease volume)
GND	Ground	Power GND
IO2	Trigger port 2	Short press to play next (long press to increase volume)
ADKEY1	AD Port 1	Trigger play first segment
ADKEY2	AD Port 2	Trigger play fifth segment
USB+	USB+ DP	USB Port
USB-	USB- DM	USB Port
BUSY	Playing Status	Low means playing \High means no

Fuente. DFRobot.com

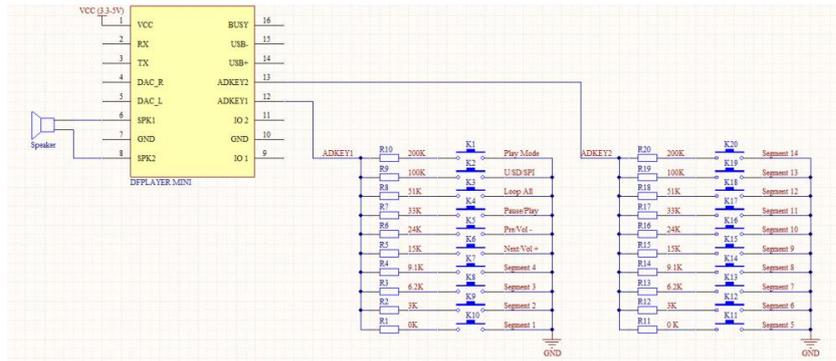


Figura 18. Conexiones internas del módulo DFPlayer mini MP3.

Fuente. DFRobot.com

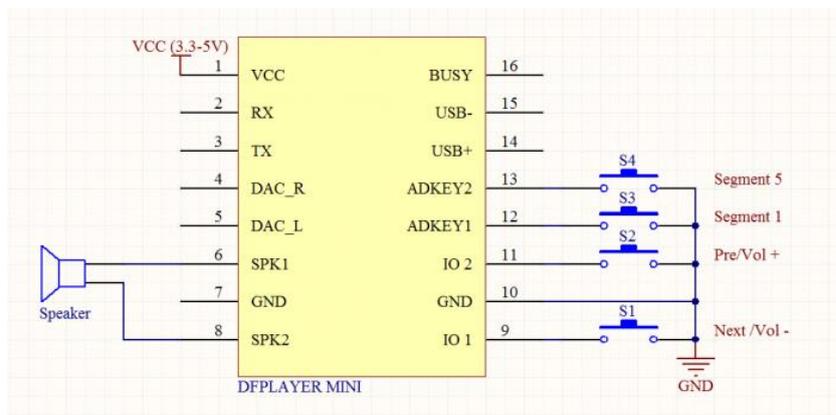


Figura 19. Conexión óptima de programación.

Fuente. DFRobot.com

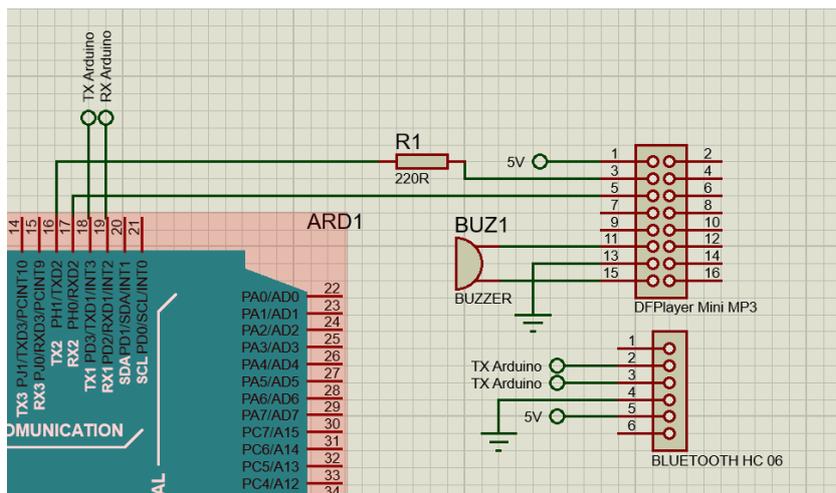


Figura 20. Esquema de conexiones para el módulo MP3.

Fuente. Elaboración propia.

4.1.2.4. Módulo ISD 1820

El módulo de grabación de voz se basa en ISD1820. el cual es un dispositivo de grabación / reproducción de múltiples mensajes. Este integrado puede ofrecer una verdadera grabación de voz de un solo chip, un almacenamiento no volátil y una capacidad de reproducción de aproximadamente 10 segundos. Este módulo es fácil de usar y puede controlarlo directamente con un botón a bordo o con un microcontrolador como Arduino, STM32, ChipKit, etc. Desde estos, puede controlar fácilmente la grabación, la reproducción y la repetición. Este módulo será utilizado para grabar y reproducir lo que el niño diga, para que este puede oírse, y sentir mayor confianza social con el robot.

Este módulo tiene un estándar de duración de 8 a 20 segundos, para manejar este tiempo de grabación se puede colocar una resistencia en relación a la tabla 13, sin embargo, por la frecuencia de trabajo es mejor trabajar en un promedio de 8 a 12 segundos.

Tabla 13. *Tabla de frecuencias del módulo ISD 1820, en relación al tiempo (seg) y la resistencia (ohmios).*

ROSC	Duration	Sample Rate	Bandwidth
80 K Ω	8 seg.	8.0 KHz	3.4 KHz
100 K Ω	10 seg.	6.4 KHz	2.6 kHz
120 K Ω	12 seg.	5.3 KHz	2.3 KHz
160 K Ω	16 seg.	4.0 KHz	1.7 KHz
200 K Ω	20 seg.	3.2 KHz	1.3 KHz

Fuente. Carrod Electrónica.

Este módulo trabaja a tensión nominal de 3 a 5 voltios en corriente continua, este módulo cuenta con un parlante para grabar la voz, este puede ser extendido dependiendo la distancia de grabación, para el modo de operación es mejor que sean distancias menores a 50 cm, caso contrario, existe mayor tiempo muerto, y mayor distorsión en el audio.

4.1.2.5. LM 386

El audio en los módulos anteriormente explicados, son de bajo audio, además de que su señal de audio no es una señal limpia, para sanear estos problemas utilizaremos un filtro Y amplificador de audio, para lo cual se utilizará el integrado LM 386, este CI es un pequeño amplificador de bajo consumo y aceptable calidad con el cual se puede conseguir potencia del orden de 700 mW bajo una tensión de 9 VDC, en cuanto al parlante uno de 8 ohmios, a 2 W de potencia mínima.

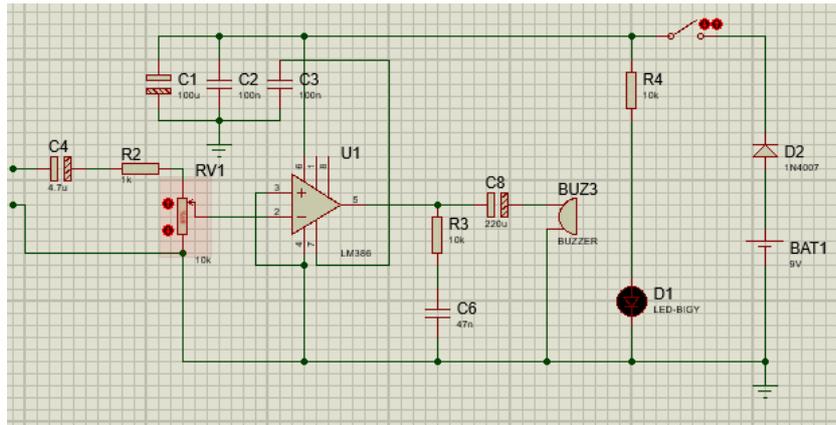


Figura 21: Filtro y amplificador de audio con CI LM 386.

Fuente. Elaboración propia.

Para el diseño de este circuito, primero se inició con un filtro pasivo pasa bajo el cual se conecta a la línea de alimentación y del condensador C3 lo unimos al pin 7 (Bypass) del integrado, seguido a ello conectamos los pines 2 y 3 del CI a la señal del Arduino y a negativo, esta señal primero se modula con un potenciómetro de 10k, en la salida del integrado conectamos un condensador a tierra, esto para evitar sobrecargas que puedan dañar al parlante, finalmente se tiene la alimentación del circuito, que es mediante una fuente de 9V, la cual tiene como protección un diodo rectificador acompañado de un switch para energizar el sistema.

4.1.3. Tercera etapa – Pantalla TFT con Arduino Uno

El robot tendrá una pantalla touch para que el niño pueda interactuar físicamente, en este proyecto se opta por una pantalla TFT táctil, debido a la compatibilidad, además de que el interfaz de usuario te permite mostrar una alta gamma de imágenes y mucha variedad en su operación, esta pantalla se alimenta a 3.3 V, las conexiones del Arduino a la pantalla son conexiones estándar ya que estos módulos fueron creados específicamente para trabajar inicialmente con Arduino Uno, en la figura XX se observa el esquema de

conexiones, a este se le adecuará las salidas digitales del Arduino para comunicarlo con el Arduino Mega.

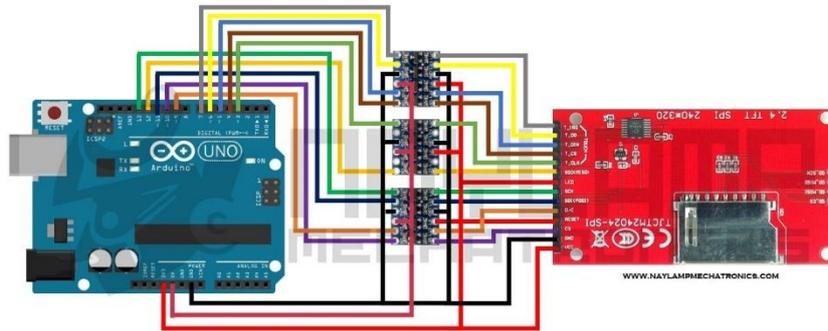


Figura 22. Esquema de conexiones Arduino – Pantalla TFT.
Fuente. Naylamp Mechatronics.



Figura 23. Conexión física Arduino – Pantalla TFT.
Fuente. Naylamp Mechatronics.

4.1.4. Cuarta etapa – Interfaz Arduino – Matlab

El procesamiento de imágenes, así como el reconocimiento facial, se hará en Matlab, el envío de señales de control se hará mediante un Arduino, este recibirá las señales de control y las procesará para que el robot ejecute las ordenes de control. La conexión de esta etapa es sencilla, la cámara se conecta al Arduino y este primero se carga con el código de Matlab.



Figura 24. Prueba del interfaz Arduino Matlab

Fuente. Foto propia.

4.1.5. Quinta etapa – Motores

En esta etapa se trabajará en las conexiones de los motores del robot, este tendrá tres motorreductores en los brazos, 1 un motor en la base y 1 servomotor en la base de la cámara, en el caso de los motorreductores se utilizará el driver L298N, para simplificar los circuitos se hará uso del módulo L298N, con este solo se tendría que unificar las tierras, alimentar un driver cada 2 motores y conectar los pines del Arduino a los del módulo.

4.1.5.1. Motorreductor con driver L298N

El driver L298N, es compuesto un módulo, el cual contiene 2 puentes H, esto da la opción de poder controlar hasta 2 motores de corriente discreta, o un motor paso a paso unipolar, mediante este modulo es posible controlar la velocidad de giro del motor, haciendo uso de PWM y señales TTL, este tipo de señales se obtienen en microcontroladores y tarjetas de desarrollo como la Raspberry Pi, el Arduino, ESPN, Launchpads, entre otro tipo de controladores.

El módulo esta integrado también por un regulador de 5V, el cual alimenta el sistema lógico del L298N, el uso de este regulador s por medio de un jumper, y puede ser utilizado para energizar la etapa de control del driver.

El módulo puede ser alimentado de dos formas, la primera es con el jumper conectado el cual permite al modulo un rango de alimentación de 6V a 12V, haciendo pruebas previas, se comprobó que el voltaje promedio que soporta el módulo es de hasta 15V DC, este

voltaje sirve para alimentar la parte de control, el módulo soporta hasta 500mA, cuando el jumper de selección no esta conectado el módulo permite un rango de tensión desde 12V hasta los 35V DC, como el pin de 5V no está conectado, se necesita alimentar la parte lógica del driver, a la cual si se utiliza un Arduino se puede conectar de los 5V del Arduino a los 5V del módulo.

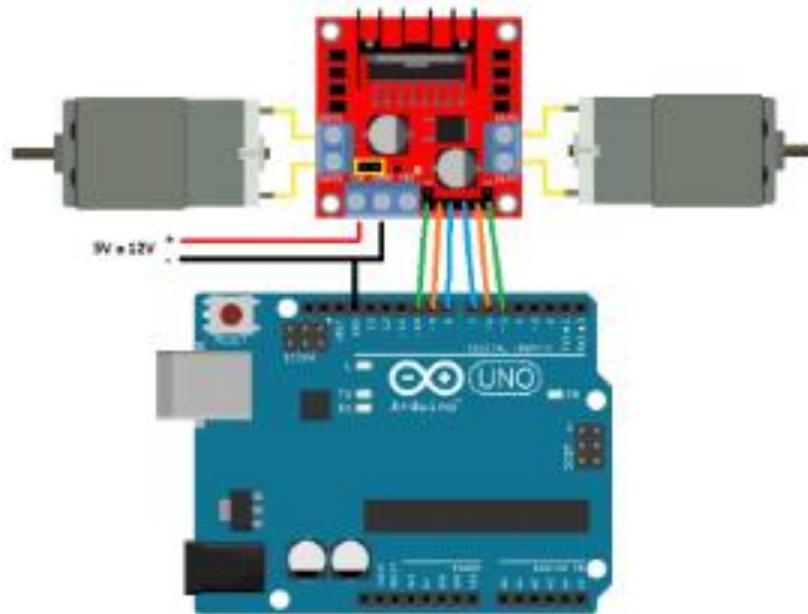


Figura 25. Esquema de conexiones del motor con L298n.

Fuente. Elaboración propia diseñado en Fritzing.

4.1.5.2. Servomotor MG

Este servo digital MG996R de alto par cuenta con engranajes de metal que dan como resultado 10kg extra altos, par de torsión en un paquete pequeño. El MG996R es esencialmente una versión mejorada del famoso servo MG995, y cuenta con mejoras a prueba de golpes y un PCB e IC rediseñados. Sistema de control que lo hacen mucho más preciso que su antecesor. El engranaje y el motor también se han actualizado para mejorar el ancho de banda y el centrado muertos. La unidad viene completa con cable de 30 cm y conector hembra tipo "S" de 3 pines que se adapta a la mayoría de los receptores, incluidos Futaba, JR, GWS, Cirrus, Blue Bird, Blue Arrow, Corona, Berg, Spektrum y Hitec.

Este servo estándar de alto par puede girar aproximadamente 120 grados (60 en cada dirección). La tensión de alimentación va de 4.8 a 7.2 V, la velocidad de operación de es de 0.17 segundos cada 60°, su torque de detención es de 9.4 Kgf cm, el ángulo de rotación es de 120°, y su banda muerta es de 5 microsegundos.

4.2. Dominio de control

En este dominio se tratará sobre la lógica de control del sistema, y los pasos a seguir en este proyecto, además de hacer pruebas de programación en cada etapa del sistema. Primero nos abocaremos en los diagramas de flujo que controlaran el robot.

4.2.1. Diagrama de flujo general

En la figura XX se muestra en un diagrama general la lógica que deberá seguir la programación, el sistema se dividirá en 6 etapas.

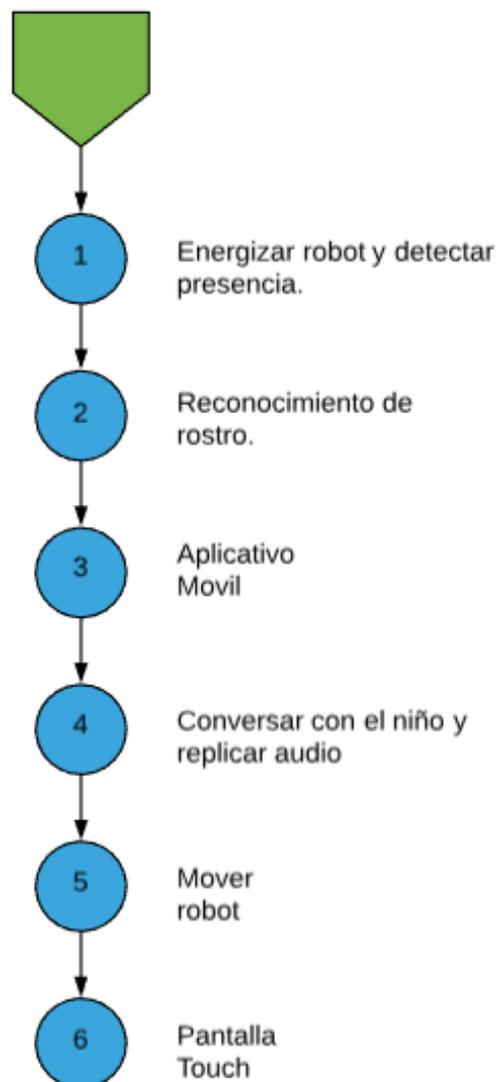


Figura 26. Diagrama de flujo por etapas.

Fuente. Elaboración propia.

4.2.2. Primera etapa – Energizar robot y detectar presencia

El nivel de lenguaje de diagrama de flujo será el nivel, ya que este es textual y puede ser entendido por todo tipo de lector.

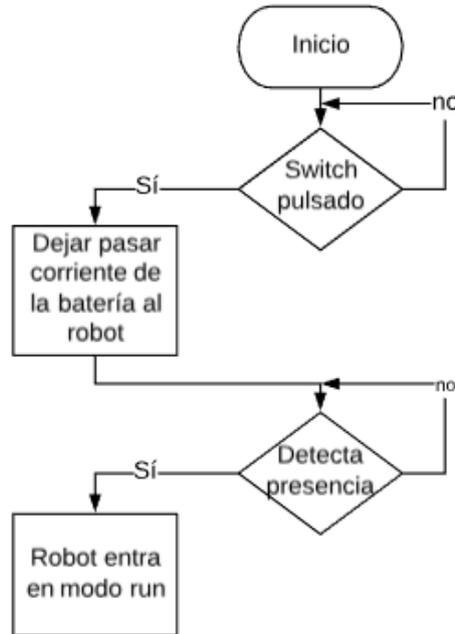


Figura 27. Diagrama de flujo de la primera etapa.

Fuente. Elaboración propia.

En esta primera etapa se designa un pulsador para energizar o des energizar el robot, este activará las baterías, después pasara a un condicional el cual esperará que el sensor de presencia ubicado en la parte frontal del robot se active, cuando este se active el robot entrará en modo run, este modo será el que inicie todo el funcionamiento, sin embargo, para que se inicie este modo y se mantenga en operación será necesario que el robot este detectando constantemente, cuando este deje de hacerlo, el robot tendrá que esperar 15 segundos para entrar en modo stop, los 15 segundos serán de espera, como esto es una etapa de finalización no se considera en esta primera etapa, ya que en esta mas importante es abocarnos en las funciones del sensor y en entrar en modo run, de forma correcta.

Para elaborar el código se debe de calibrar el sensor ultrasónico para así lograr mayor exactitud, las pruebas se realizaron con una pantalla LCD 1602.

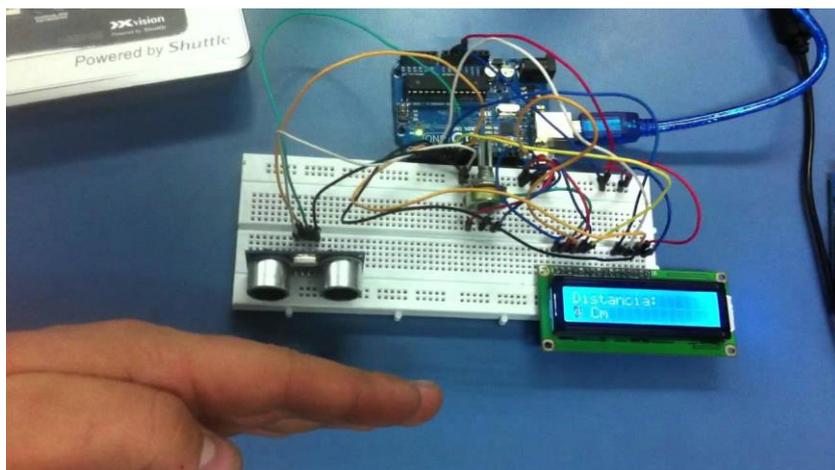


Figura 28. Prueba del sensor ultrasónico.

Fuente. Fotografía propia.

En respuesta a las pruebas realizadas del sensor, se obtuvo que el sensor detecta desde 1.75 cm hasta 2.53 metros, bajo este rango de trabajo, no habrá problemas en la interacción del robot con el niño, para asegurarnos que el robot no detecte a las paredes u otros objetos como si se tratase del niño, el intervalo de trabajo será como máximo de 1.70 metros, con esto se asegura que el robot trabaje a referencia única del niño.

4.2.3. Segunda etapa – Reconocimiento de rostro

La segunda etapa del dominio de control, se hará en Matlab con interfaz de Arduino, el reconocimiento facial es un resultado de la visión artificial.

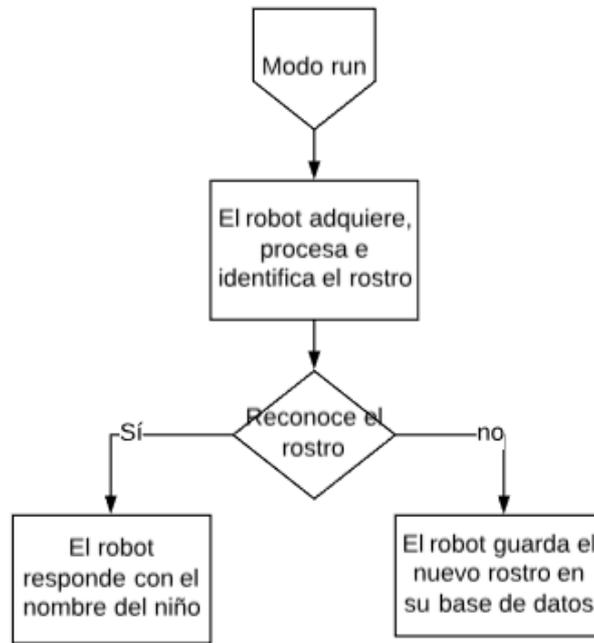


Figura 29. Diagrama de flujo de la segunda etapa.

Fuente. Elaboración propia.

Para las pruebas de la segunda etapa se hizo todo bajo simulación en el software Matlab, la visión artificial se trabajó bajo redes neuronales, el programa elaborado se trabajó en 4 neuronas, antes del entrenamiento de RNA se procesó las imágenes pixeleando cada una a un formato estándar, y poniendo todos en escala de grises.

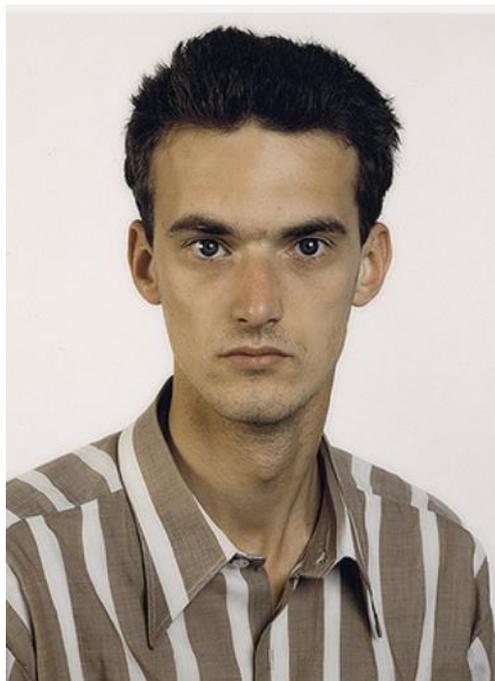


Figura 30. Imagen 1 de prueba.

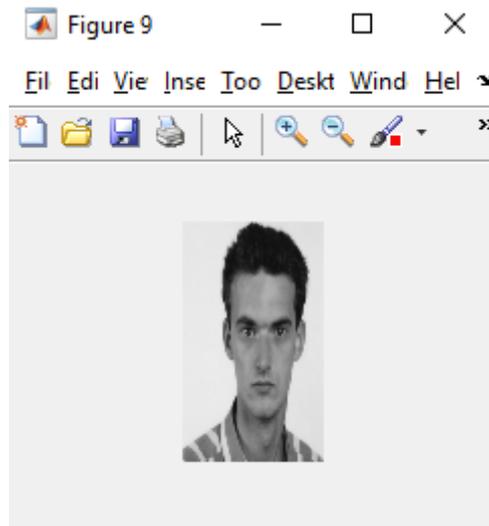


Figura 31. Imagen 1 de prueba en escala de grises y a 20*120 pixeles.

Fuente. Elaboración propia.

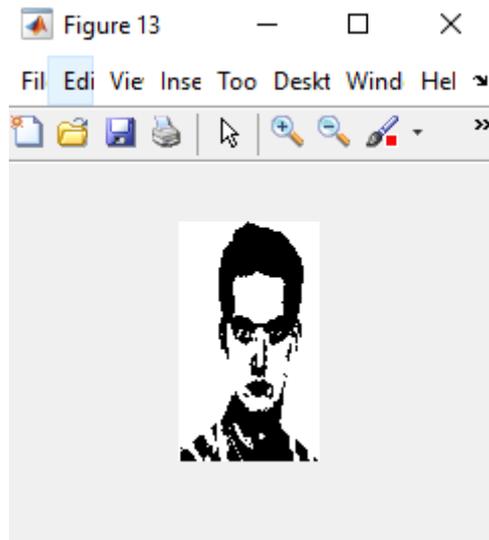


Figura 32. Imagen procesada.

Fuente. Elaboración propia.

Las pruebas se hicieron en un total de 10 imágenes, el segundo paso es entrenar las imágenes en una red neuronal artificial, estas redes se desarrolló en 4 partes, en las cuales se les da un porcentaje de ruido a las imágenes, ya que de esa forma es de las mas adecuadas, debido a que las imágenes adquiridas de la cámara generalmente no son imágenes limpias, por eso es necesario trabajar con imágenes que tengan ruido, esto da mayor garantía al reconocimiento facial.

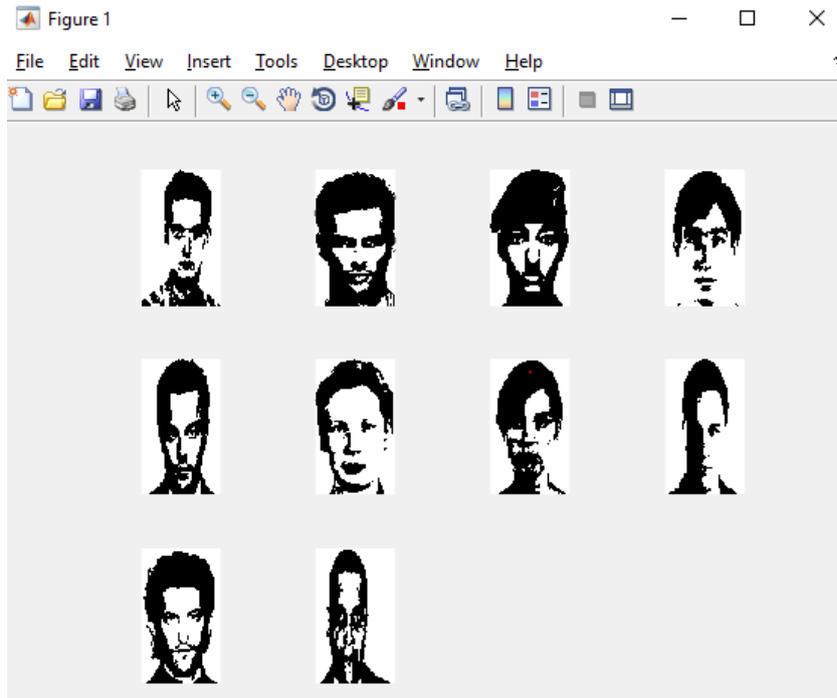


Figura 33. Imágenes procesadas sin ruido.

Fuente. Elaboración propia.



Figura 34. Imágenes con ruido del 5%.

Fuente. Elaboración propia.

Como se observa en la figura XX las imágenes no son limpias, tiene un ruido aproximado del 5%, y además el sistema pudo identificar todos los rostros sin ningún error, cuando el sistema de redes neuronales se le ingresa imágenes con un ruido mayor al 15 % recién comete errores de 2 a 3 imágenes,

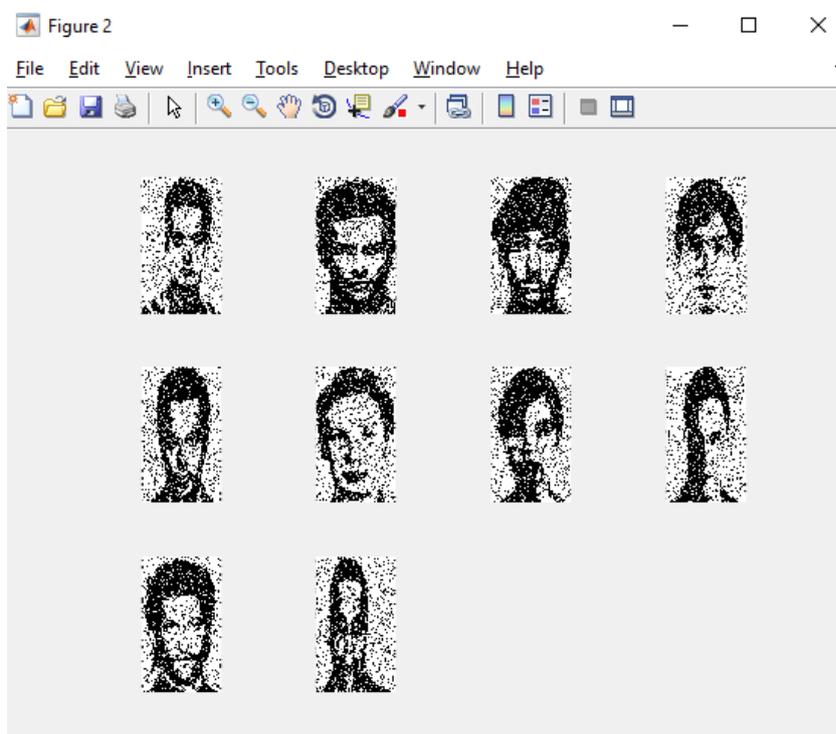


Figura 35. Imágenes procesadas con ruido de 15%.

Fuente. Elaboración propia.

En este punto el sistema empieza a confundir la imagen 3 con la imagen 10, debido a que la forma del rostro empieza a asemejarse.

En cuanto a la segunda neurona el procesamiento es diferente, lo que se hace es un comparación múltiple para una sola imagen, es decir lo que se busca es tratar cada imagen por separado y no como se hacía en la neurona anterior de buscar muchas similitudes para mas de una imagen, la ganancia de esta en comparación de la anterior es que podemos denominar esta neurona como una segunda parte ya que en esta se busca la validación pese al bajo porcentaje de parecido con otras imágenes. Es decir, pese al ruido que pueda tener la imagen pasa esta etapa donde la validación es mas rigurosa y si toma en cuenta cada punto específico.

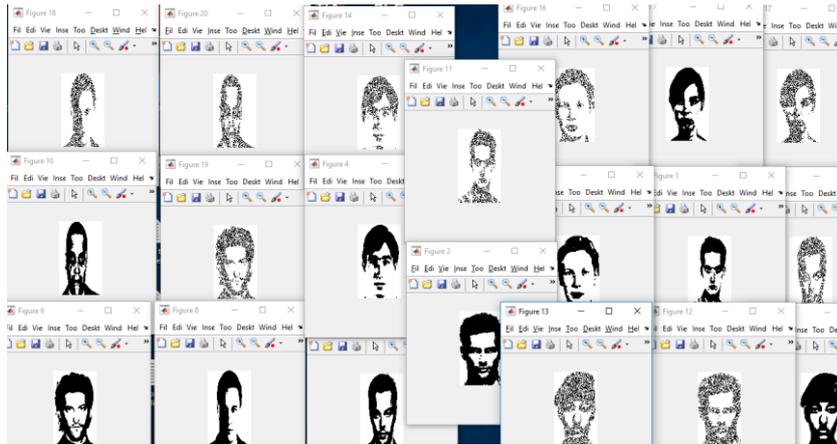


Figura 36. Prueba de segunda neurona de una imagen con todas las posibilidades de la base de datos.

Fuente. Elaboración propia.

Pero estas dos neuronas son de validación, ¿Cómo el sistema llega a obtener estos resultados?, para responder esta incógnita se debe de explicar las dos neuronas de pre validación, las cuales serían las de reconocimiento, las redes neuronales artificiales o RNA tienen base fundamental en la matemática.

Tabla 14. Fórmulas de la función de activación de una red neuronal.

<p>FUNCIÓN PASO</p>	$f(u_i) = \begin{cases} 1 & \text{si } u_i > \theta_i \\ 0 & \text{si } u_i \leq \theta_i \end{cases}$
<p>FUNCIÓN RAMPA</p>	$f(u_i) = \begin{cases} u_i & \text{si } 0 \leq u_i \leq 1 \\ 0 & \text{si en otro caso} \end{cases}$
<p>FUNCIÓN SIGMOIDAL</p>	$f(u_i) = \frac{1}{1 + e^{-\frac{u_i}{\sigma}}}$
<p>FUNCIÓN GAUSSIANA</p>	$f(u_i) = ce^{-\frac{u_i}{\sigma}}$

Fuente. Redes neuronales y su aplicación en la lógica difusa.

Con respecto a la tabla 4, una red neuronal tiene diferentes activaciones, a estas se les conoce como la función de activación, desde una función bastante simple como es una función paso o una función escalón hasta una función gaussiana o sigmoidea, la más común en sistemas estables es la función paso, debido a que esta solo cuenta con dos estados de forma similar a una señal digital, para este proyecto se usa una función sigmoidea, debido a la activación será solo de subida dependiendo en el rango en el que se encuentre.

En segundo lugar, tenemos los pesos, los cuales son actualizados para mejorar la precisión de la red, en Matlab estos pesos se manejaron bajo las siguientes líneas de código:

```
v = 0.1*(rand(ne,nm) - 0.5)*2;
```

```
w = 0.1*(rand(nm,ns) - 0.5)*2;
```

La obtención y actualización de pesos se rigen bajo la siguiente fórmula:

$$W_j = W_{j-1} + e * X_j * \alpha$$

Donde:

W_j = Peso actualizado

W_{j-1} = Peso actual

e = error

α = coeficiente de aprendizaje

X_j = entrada de la neurona.

Estos pesos se actualizan por cada iteración, es decir por cada ciclo que de la red, generalmente para sistemas robustos se suele dar más de 4000 iteraciones, en sistemas complejos mientras mas iteraciones se realicen, se obtiene un mayor ratio de aprendizaje, sin embargo esto toma tiempo, en caso de este robot se aplican 3000 iteraciones, este número se obtuvo mediante la forma de una onda sigmoidea, ya que al aumentar más iteraciones los resultados suelen ser similares, por tal motivo, afirmo que para el sistema desarrollado las 3000 iteraciones son suficientes.

Los pesos óptimos obtenidos a un factor de aprendizaje de 0.01, es de $5.003e-15$

Tal como se observa en la siguiente imagen, la precisión de imágenes, son bastantes similares, lo cual indica que el sistema está trabajando adecuadamente.

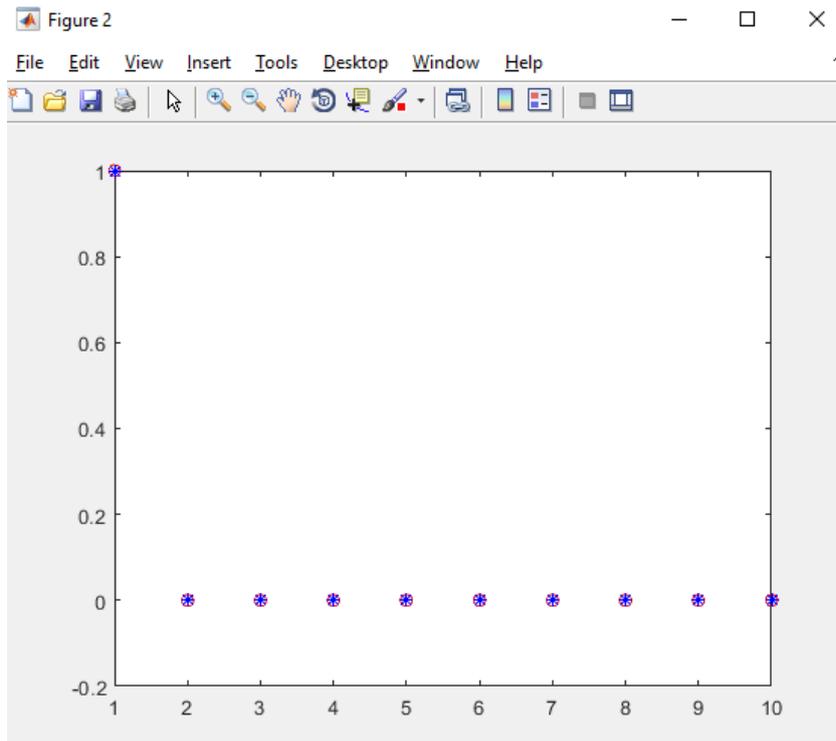


Figura 37. Simulación en Matlab de la proximidad de entrenamiento.

Fuente. Elaboración propia.

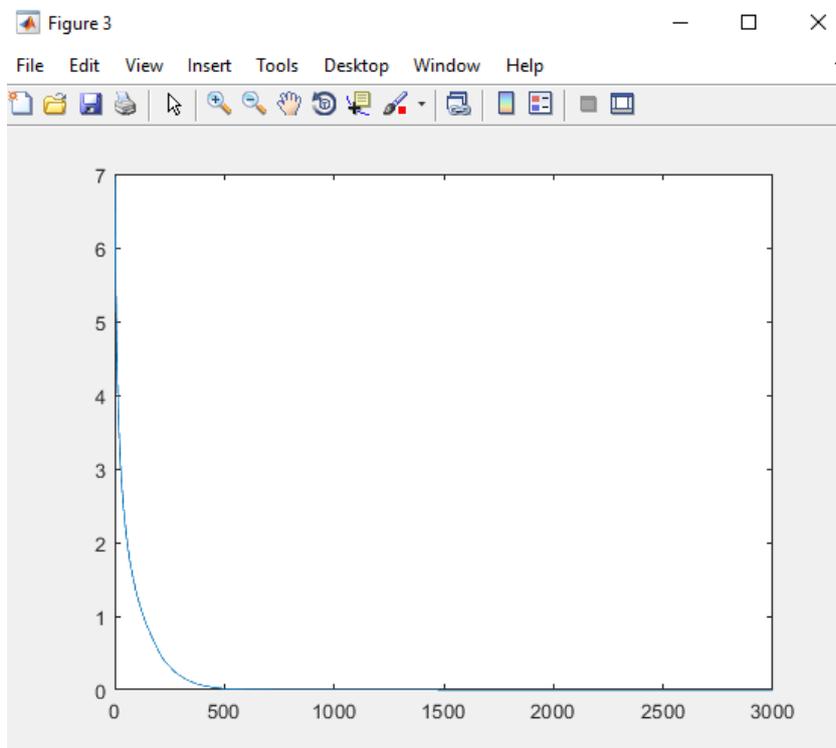


Figura 38. Curva de entrenamiento de la neurona.

Fuente. Elaboración propia.

En el tratamiento de la otra neurona de entrenamiento, se hace por partes de 4 en 4, mientras que la otra tomaba el grupo completo para el entrenamiento, en este caso se hizo con 2000 iteraciones.

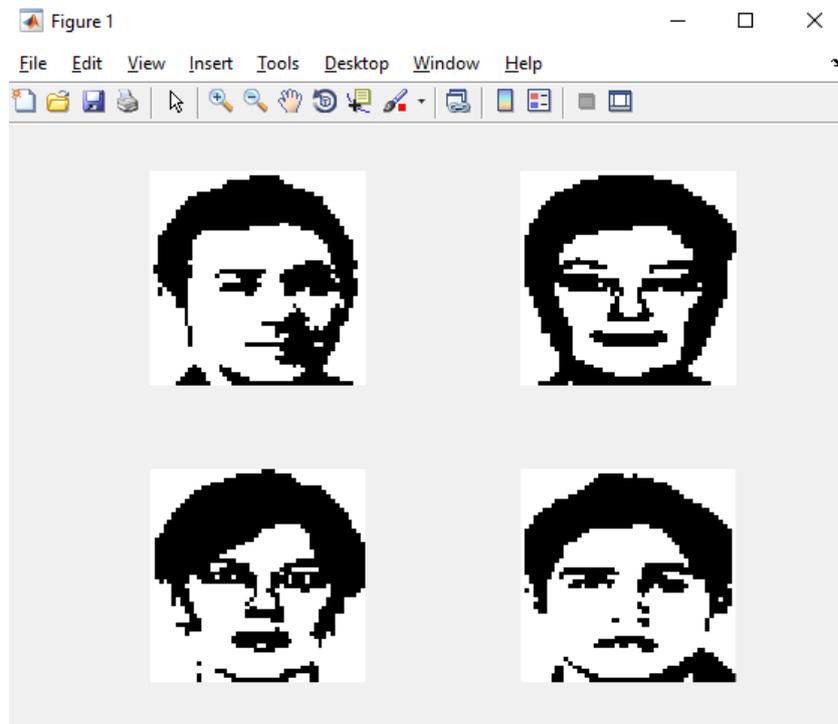


Figura 39. Entrenamiento por grupos.

Fuente. Elaboración propia.

Como se observa en la anterior imagen, primero procesa las imágenes, luego en las figuras XX XX, se observa el patrón de entrenamiento, para 2000 iteraciones con un ratio de aprendizaje de 0.02, se obtiene buenos resultados, aumentando el ratio de aprendizaje a 1, no se puede observar un aprendizaje debido a que existe alta variación, y esto ocasiona que el sistema no sea muy efectivo, es por eso que se prueba con ratios bajos para que no alteren el funcionamiento de la neurona.

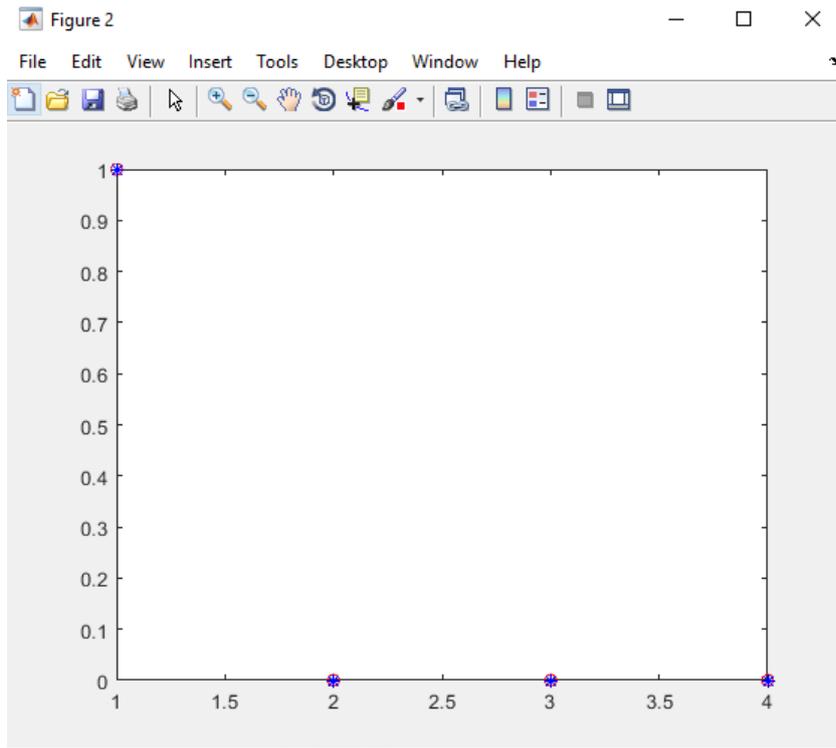


Figura 40. Simulación de precisión de entrenamiento.

Fuente. Elaboración propia.

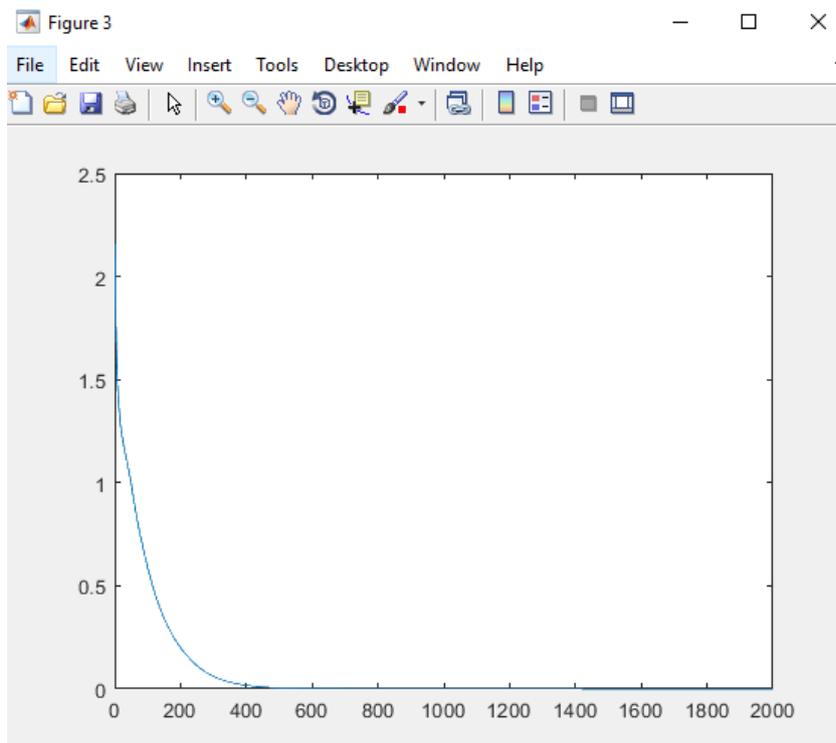


Figura 41. Curva de aprendizaje de los pesos con alfa = 0.02

Fuente. Elaboración propia.

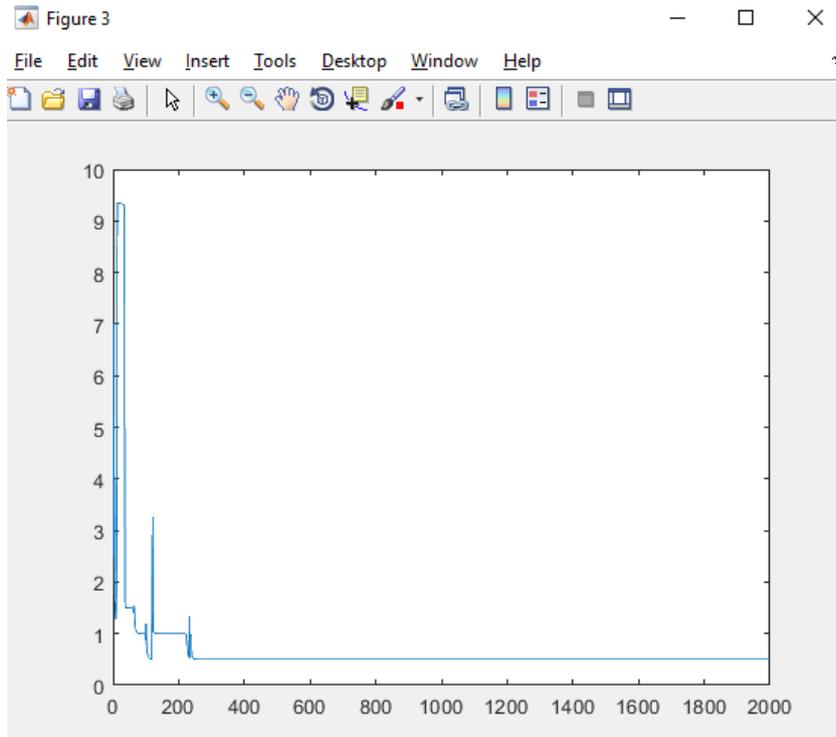
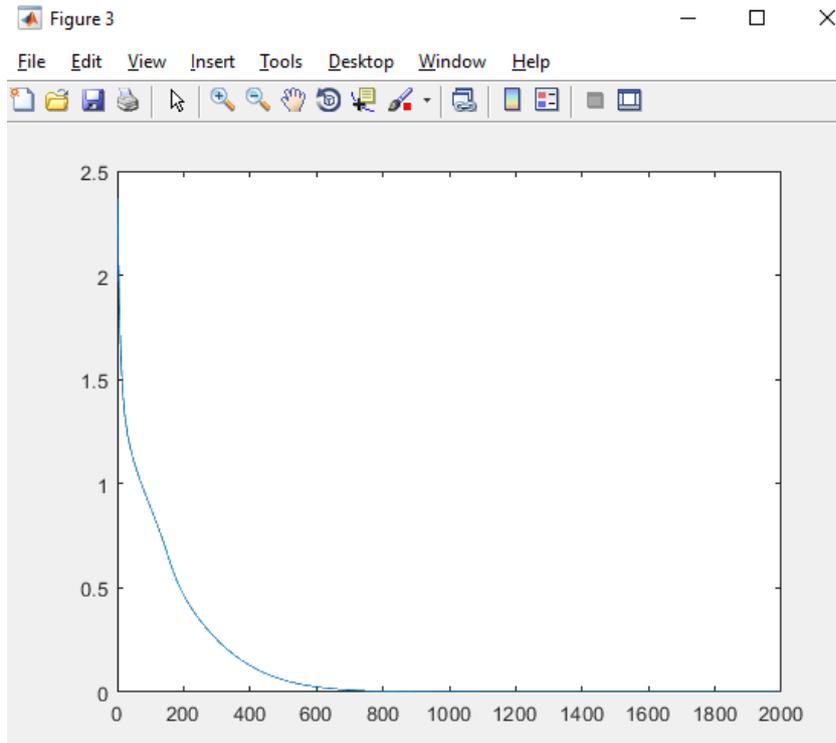


Figura 42. Entrenamiento de la segunda neurona con coeficiente de aprendizaje = 1.

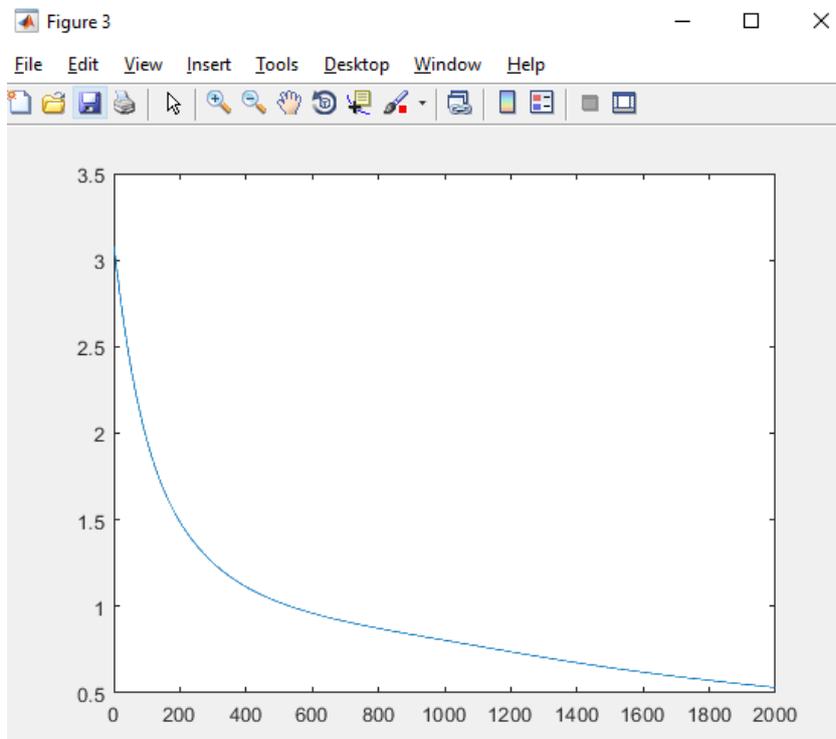
Fuente. Elaboración propia.

Se observa que esta curva, a diferencia de la otra, en menos de 100 iteraciones sufre un pico alto, y luego se linealiza, este quiere decir que el coeficiente de aprendizaje propuesto no es el adecuado, ya que es absurdo utilizar 2000 iteraciones, cuando el sistema solo en 100 empieza a tener fallas.

Para subsanar este inconveniente el coeficiente de aprendizaje se redujo aún más, utilizado $\alpha = 0.01$, se obtuvo resultados óptimos, sin embargo no es aconsejable utilizar menos coeficiente de aprendizaje de 0.01, ya que al igual que con $\alpha = 1$, el sistema se vuelve lineal con pendiente negativa, como se sabe estaríamos aproximando a función sin ganancia, es decir sería como utilizar un control netamente proporcional, y lo que se busca es un curva que pueda amoldarse a la situación, algo que es imposible con una función lineal.



*Figura 43. Curva de pesos, alfa = 0.08.
Fuente. Elaboración propia.*



*Figura 44. Curva de pesos, alfa = 0.05.
Fuente. Elaboración propia.*

4.2.4. Tercera etapa – Aplicativo Móvil

El aplicativo móvil se desarrolló en MIT App Inventor, debido a que se utilizará el Google voice, para el reconocimiento de las palabras emitidas por el niño.

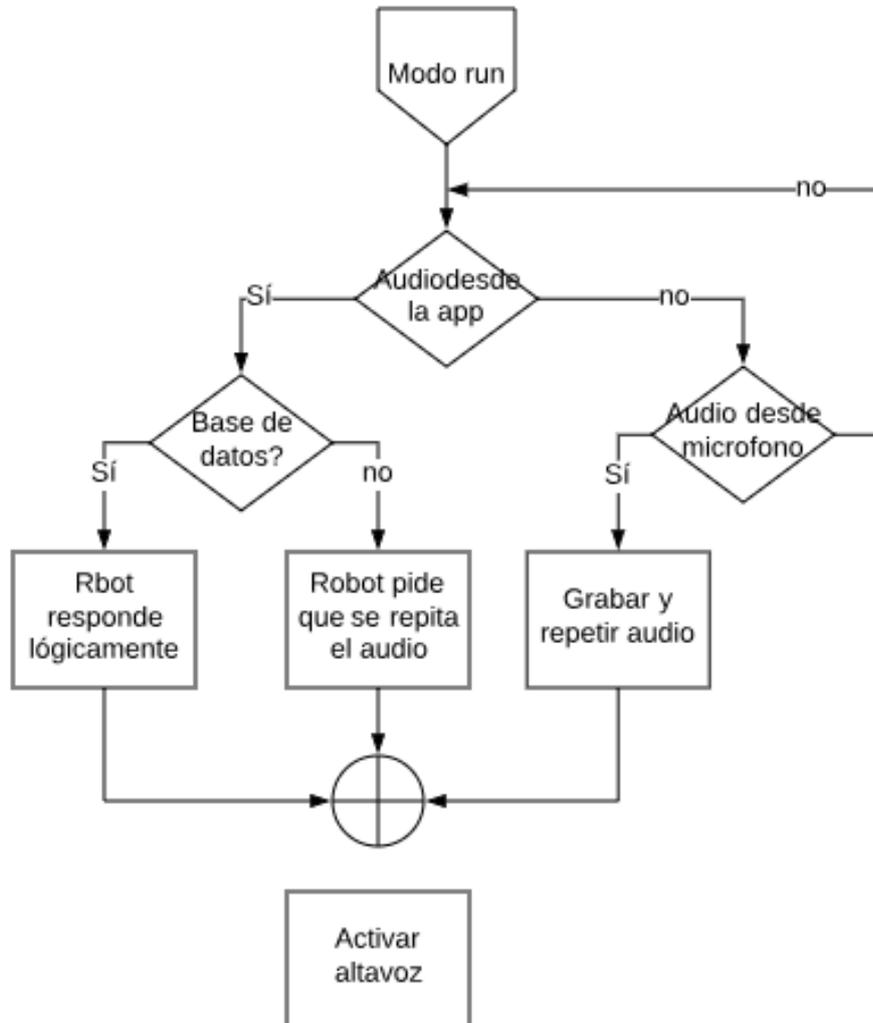


Figura 45. Diagrama de flujo de la tercera etapa.

Fuente Elaboración propia.

Como se observa en el diagrama, el robot dará prioridad a la app, debido a que esto permite interacción, niño – robot, y el otro su función básica es de grabar y guardar y reproducir lo escuchado.

La comunicación del Bluetooth con la app, será mediante un módulo HC 06, la configuración del BT será mediante comando AT (ver figura XX), y la comunicación será

mediante el monitor serie, ya que la comunicación es por ambos lados, donde ambos emiten y reciben.

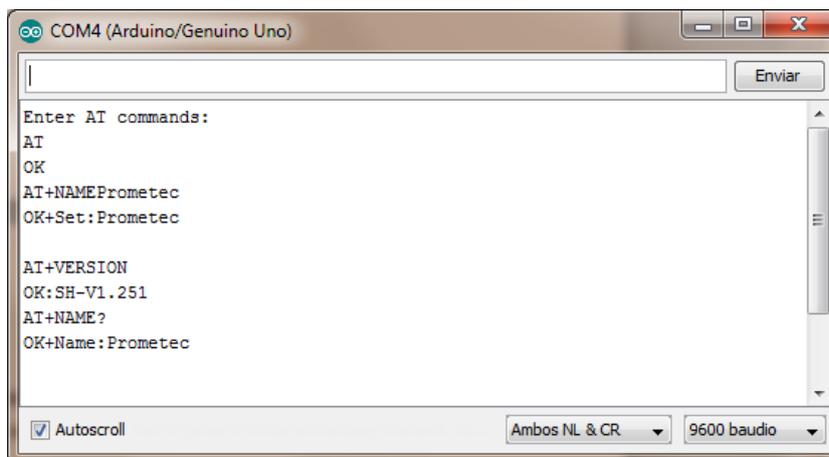


Figura 46. Comandos AT.

Fuente. Prometec.

La comunicación, niño – robot se hará con el módulo DFPlayer mini MP3, este módulo nos permite almacenar muchos audios en diferentes carpetas, y tener en base a ello una abse de datos bastante amplia, para el uso de este módulo se necesita la librería DFRobot mini, el formato para almacenar los audios será en mp3, y se usará un total de 70 reproducciones por carpeta, haciendo un total de 700 audios.

Lo referente al interfaz se desarrolló con sencillez para que el usuario no tenga mayor complicación, así mismo se necesitará desarrollar la lista de bloques en la cual se hará la lógica de control de la app.

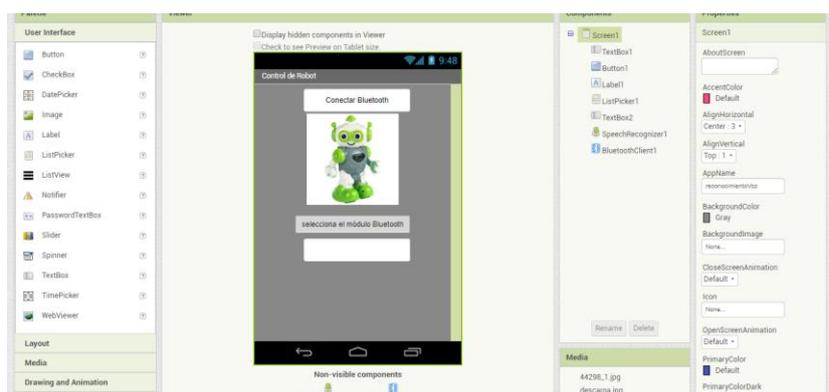


Figura 47. Interfaz de la app.

Fuente. Elaboración propia.

4.2.5. Quinta etapa – Mover robot

En esta etapa se desarrollará la lógica de programación de los motores, en cuanto a la regulación del PWM, estará en el rango de 100 – 150 cuando la alimentación sea de 9V, para asegurar que el robot sea lento en sus movimientos, esto con el fin de cuidar la integridad del niño, ya que, a velocidades altas del robot, puede dañar al menor.

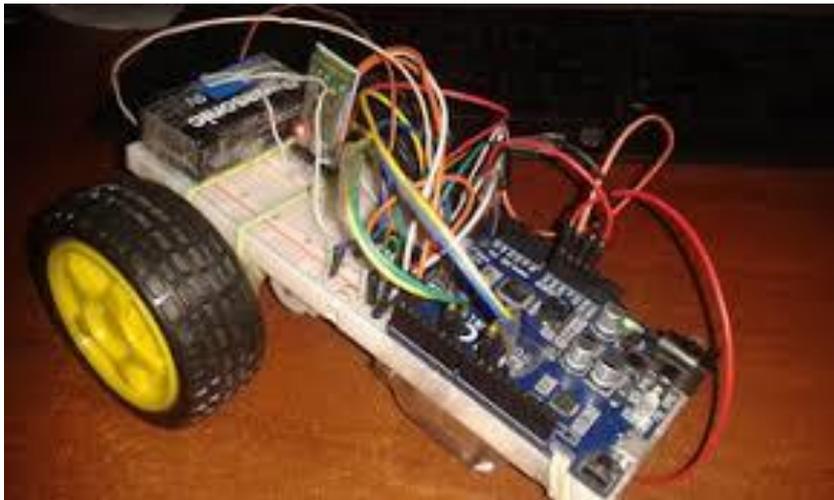


Figura 48. Pruebas de motorreductor.

Fuente. Foto propia.

Para las pruebas del servomotor se midió el ángulo y el tiempo de respuesta dando resultados esperados en cuanto a su operatividad.

4.3. Dominio Mecánico

4.3.1. Diseño y planos

El diseño se elaboró en SolidWorks 2017, el robot se hizo en partes, ya que de esta forma facilitará el montaje y desmontaje del robot, los planos mostrados a continuación son para un robot de 10 cm de alto, y 7 cm de ancho, el diseño deberá ser infantil y amigable ante la visión del niño.

4.3.1.1. Diseño y planos del brazo

Para realizar las extremidades inferiores del robot, se consideró la situación de que el robot deberá mover su brazo, por tal motivo esta parte del cuerpo de robot se particionó en brazo y antebrazo, las medidas del brazo se visualizan en la figura XX.

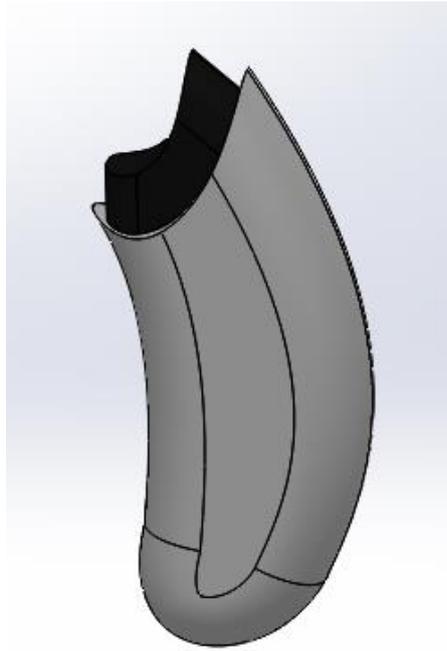


Figura 49. Vista frontal del brazo del robot.
Fuente. Elaboración propia.

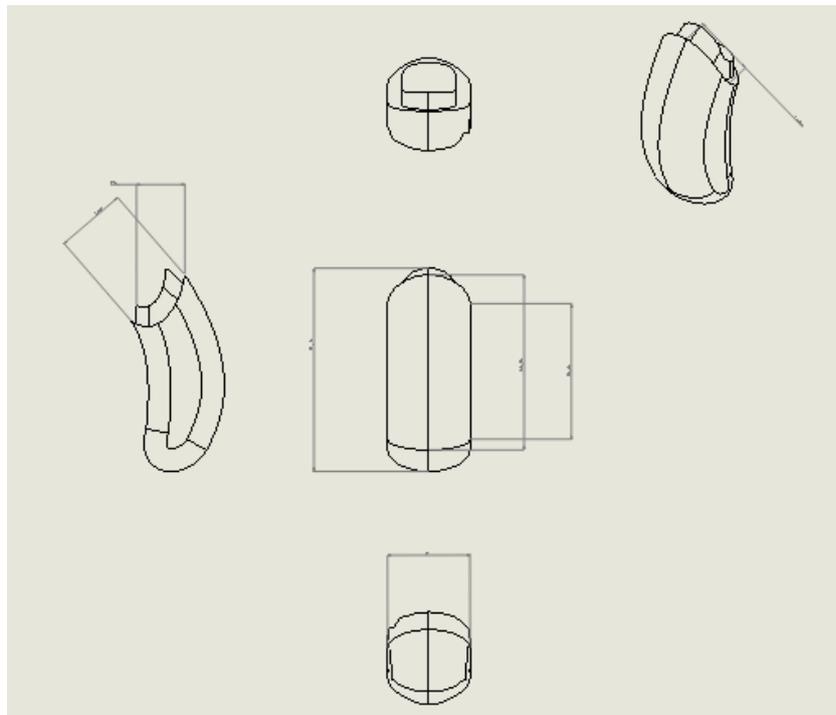


Figura 50. Vista en 5 planos diferentes del brazo acotado.
Fuente. Elaboración propia.

4.3.1.2. Diseño y planos del antebrazo

Esta parte será el conector entre el pecho y el brazo, su movimiento será por inercia, es decir será proporcional al giro del brazo.

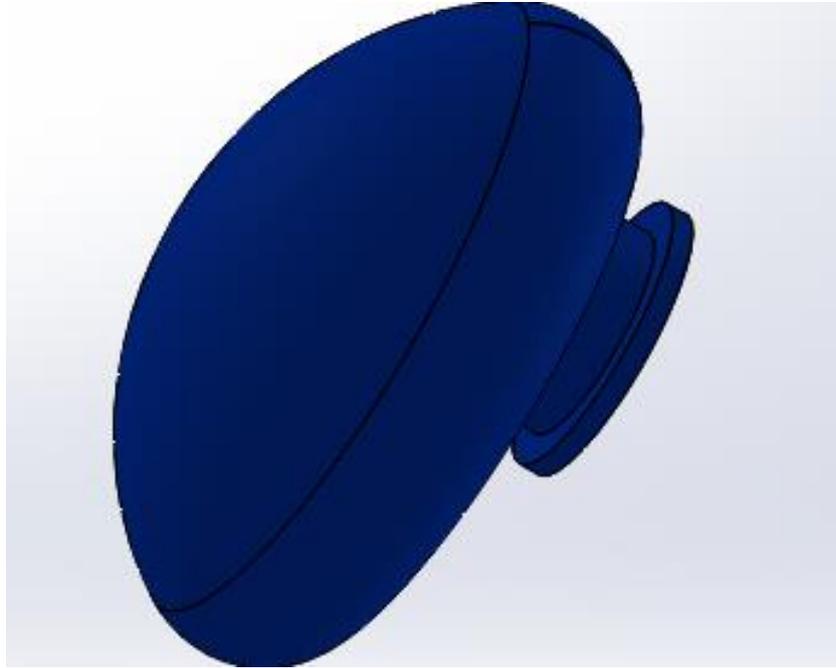


Figura 51. Vista isométrica de antebrazo.

Fuente. Elaboración propia.

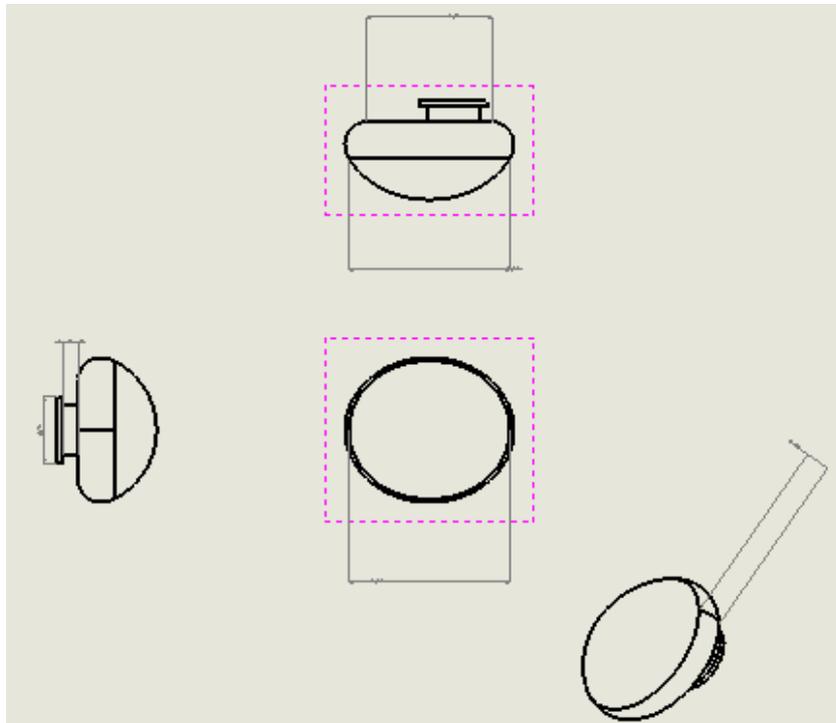


Figura 52. Planos del antebrazo del robot.

Fuente. Elaboración propia.

4.3.1.3. Diseño y planos del pecho

Las consideraciones tomadas para diseñar la pechera, fue de que en ahí se colocará la pantalla TFT de 2.4 pulgadas.

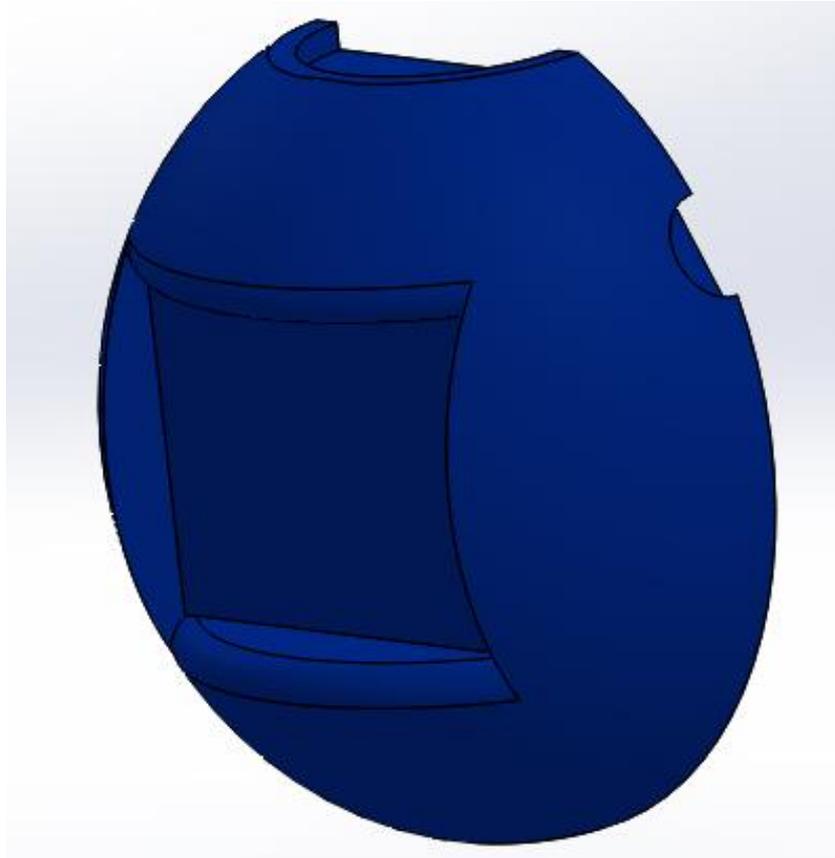


Figura 53. Vista isométrica de la pechera del robot.

Fuente. Elaboración propia.

De igual forma se diseñó la parte posterior de la pechera, la única diferencia es el espacio de la pantalla TFT, además de los sujetadores para el antebrazo y la cabeza del robot, el diseño de la pechera deberá ser hueco en el interior, ya que es en esa zona en la que se colocará el sistema de control, la cual no debe ser de dimensiones muy estrechas, por el contrario deberá tener la mayor amplitud posible para evitar desconexiones o fallos electrónicos ocasionados por la mucha presión cableada.



Figura 54. Vista isométrica de la espalda del robot.

Fuente. Elaboración propia.

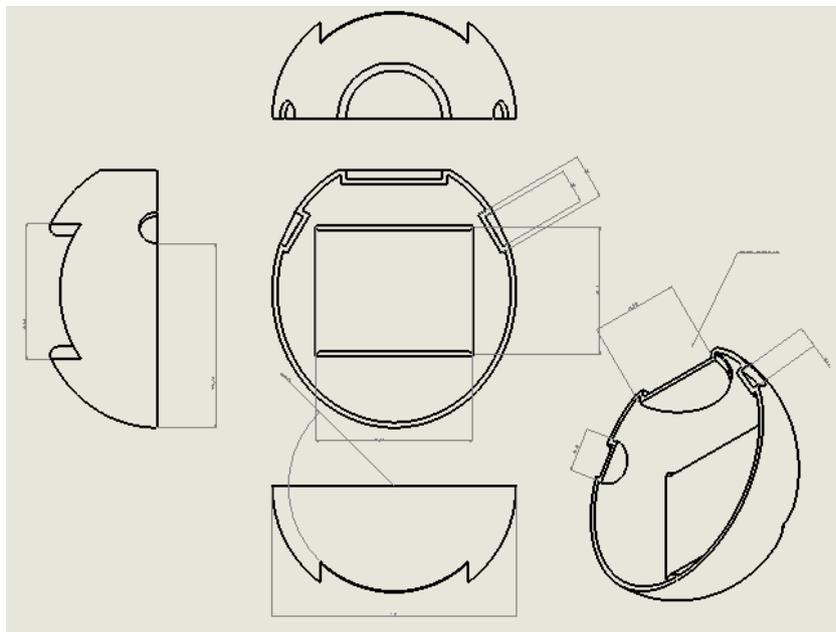


Figura 55. Vista en 5 planos de pechera del robot.

Fuente. Elaboración propia.

4.3.1.4. Diseño y planos de la cabeza y ensamblaje del robot

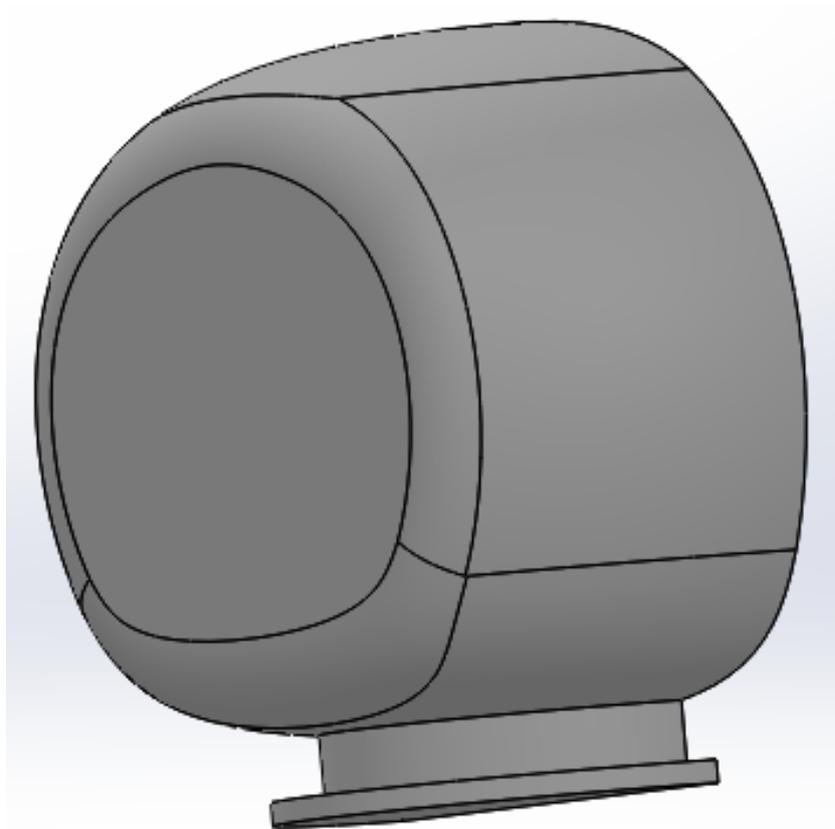


Figura 56. Cabeza del robot.

Fuente. Elaboración propia.

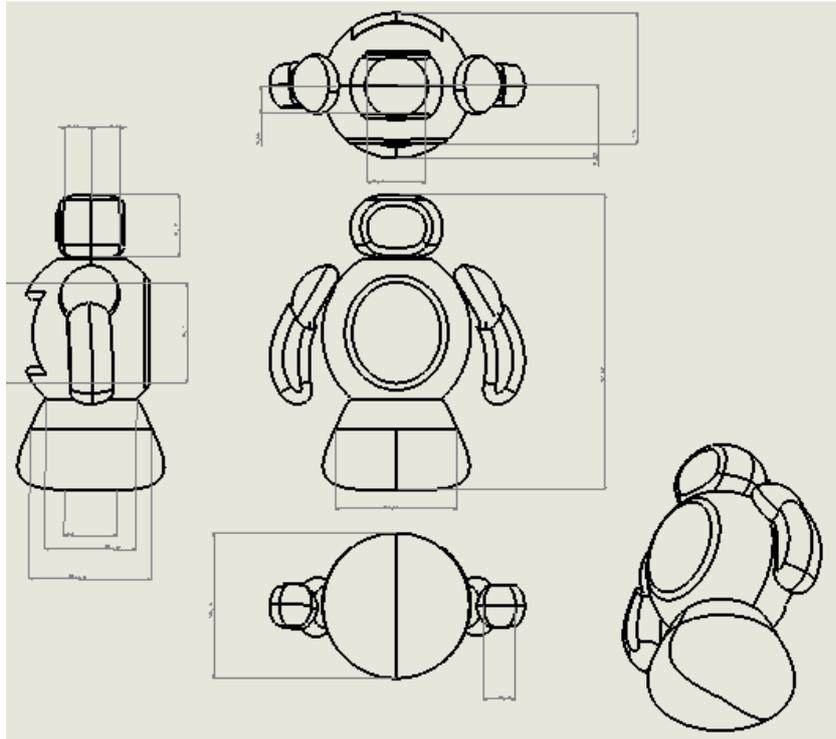


Figura 57. Vista en planos diferentes del robot.

Fuente. Elaboración propia.

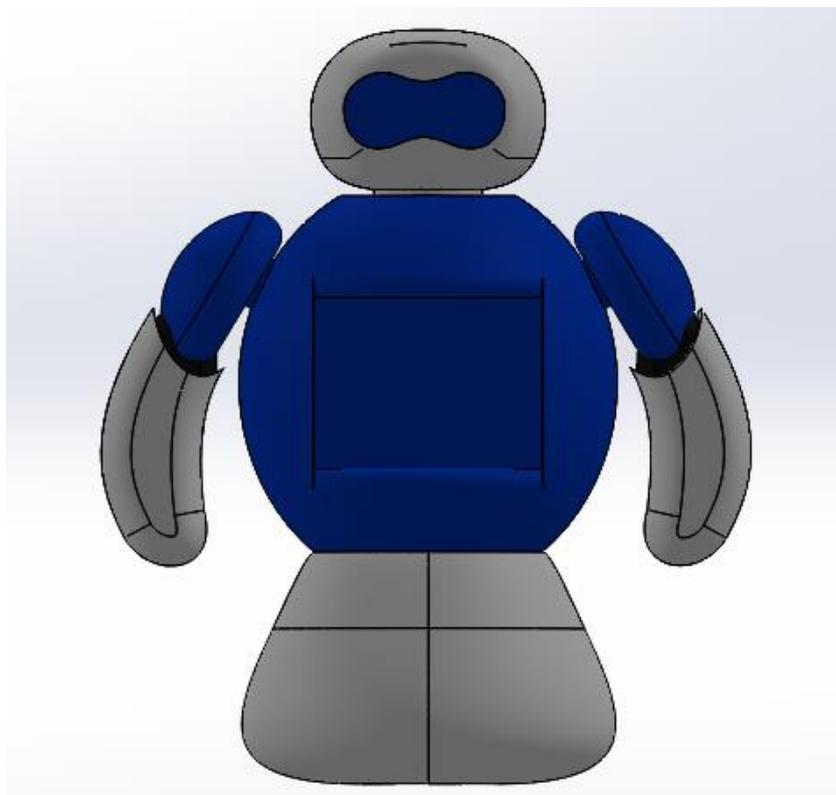


Figura 58. Vista frontal del robot.

Fuente. Elaboración propia.

BIBLIOGRAFÍA

1. ACHIC, Fred. Diseño del sistema de control de un brazo robótico de asistencia a personas discapacitadas. Tesis (para optar el grado de magister en ingeniería mecatrónica). Lima: Pontificia Universidad Católica del Perú, 2016.
2. CHAVEZ, Paul. diseño e implementación de un robot humanoide asistencial controlado por computador para aplicaciones en pacientes parapléjicos. Tesis (Para optar el título en ingeniero mecatrónico). Lima: Universidad Ricardo Palma, 2013.
3. VALENCIA, David y GONZALEZ, Luis. Diseño e implementación de un prototipo de robot asistente para personas con discapacidad motriz y adultos mayores, basado en inteligencia artificial. Tesis (para optar el título de ingeniero eléctrico). Ecuador: Universidad Politécnica Salesiana, 2014.
4. MARTIN, Fernando. Sistema de Interacción Humano-Robot basado en Diálogos Multimodales y Adaptables. Tesis (doctorado). España: Universidad Carlos III de Madrid, 2014.
5. Handleman, J.S., Harris, S., eds. *Preschool Education Programs for Children with Autism* (2nd ed). Austin, TX: Pro-Ed. 2000.
6. National Research Council. *Educating Children with Autism*. Washington, DC: National Academy Press, 2001.
7. Gupta, Vidya Bhushan. *Complementary and Alternative Medicine*. New York Medical College and Columbia University, 2004. *Pediatric Habilitation*, volume 12.
8. Levy, S. *Complementary and Alternative Medicine Among Children Recently Diagnosed with Autistic Spectrum Disorder*; *Journal of Developmental and Behavioral Pediatrics*, December 2003; vol 24: pp 418-423. News release, Health Behavior News Service.

ANEXOS

1. Procesamiento de imágenes 1

```
clear all;
close all;
clc;

I1 = imread('cara_2.jpg','jpg');
figure(1);
imshow(uint8(I1));
I2 = I1;
Iz = I1(:,:,2);
[nf nc] = size(Iz);
figure(2);
imshow(uint8(Iz));

for f = 1:nf
    Ifila = Iz(f,:);
    if(any(Ifila < 190))
        break;
    end
end
nfc = 2400;
Iz1 = Iz(f:nfc,:);
figure(3);
imshow(uint8(Iz1));

for c1 = 1:nc
    Icol = Iz1(:,c1);
    if(any(Icol < 190))
        break;
    end
end
Iz2 = Iz1(:,c1:nc);
figure(4);
imshow(uint8(Iz2));

[nc2 nf2] = size(Iz2);
%for c2 = nc2:-1:1
for c2=nf2:-1:1
    Icol = Iz2(:,c2);
    if(any(Icol < 190))
        break;
    end
end
Iz3 = Iz2(:,1:c2);
figure(5);
imshow(uint8(Iz3));

% Pixelamiento en cuadrados de 10x10
%Izz = Iz3(1:1650,1:1390);
Izz=Iz3(1:2400,1:1460);
ii = 1;
for i = 1:10:2400
    jj = 1;
    for j = 1:10:1460
        im = Iz3(i:i+9,j:j+9);
```

```

        im = sum(sum(im))/100;
        Inueva10(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
figure(8);
imshow(uint8(Inueva10));

% Pixelamiento en cuadrados de 20x20
Izz = Iz3(1:2400,1:1460);
ii = 1;
for i = 1:20:2400
    jj = 1;
    for j = 1:20:1460
        im = Iz3(i:i+19,j:j+19);
        im = sum(sum(im))/400;
        Inueva20(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
figure(9);
imshow(uint8(Inueva20));

% Pixelamiento en cuadrados de 30x30
Izz = Iz3(1:1650,1:1380);
ii = 1;
for i = 1:30:1650
    jj = 1;
    for j = 1:30:1380
        im = Iz3(i:i+29,j:j+29);
        im = sum(sum(im))/900;
        Inueva30(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
%figure(10);
%imshow(uint8(Inueva30));

% Pixelamiento en cuadrados de 40x40
Izz = Iz3(1:1640,1:1360);
ii = 1;
for i = 1:40:1640
    jj = 1;
    for j = 1:40:1360
        im = Iz3(i:i+39,j:j+39);
        im = sum(sum(im))/1600;
        Inueva40(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
%figure(11);
%imshow(uint8(Inueva40));

% Pixelamiento en cuadrados de 40x30

```

```

Izz = Iz3(1:1640,1:1380);
ii = 1;
for i = 1:40:1640
    jj = 1;
    for j = 1:30:1380
        im = Iz3(i:i+39,j:j+29);
        im = sum(sum(im))/1200;
        Inueva43(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
%figure(11);
%imshow(uint8(Inueva43));

% Pixelamiento en cuadrados de 30x40
Izz = Iz3(1:1650,1:1360);
ii = 1;
for i = 1:30:1650
    jj = 1;
    for j = 1:40:1360
        im = Iz3(i:i+29,j:j+39);
        im = sum(sum(im))/1200;
        Inueva34(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
%figure(12);
%imshow(uint8(Inueva34));

I20=Inueva20;
I20bn= 255*round(I20/255*0.999);
figure(13);
imshow(uint8(I20bn));
%I40 = Inueva40;
%I40bn = 255*round(I40/255*0.75);
%figure(13);
%imshow(uint8(I40bn));
%I41bn = 255*round(I40/255*0.90);
%figure(14);
%imshow(uint8(I41bn));
%I42bn = 255*round(I40/255*0.98);
%figure(15);
%imshow(uint8(I42bn));

```

2. Procesamiento de imágenes 2

```
% Lee imagenes en jpg, pixelea y baja la resolución
clear all;
close all;
clc;

I1 = imread('cara_10.jpg','jpg');
figure(1);
imshow(uint8(I1));
I2 = I1;
Iz = I1(:,:,2);
[nf nc] = size(Iz);
figure(2);
imshow(uint8(Iz));
for f = 1:nf
    Ifila = Iz(f,:);
    if(any(Ifila < 190))
        break;
    end
end
nfc = 1700;
Iz1 = Iz(f:nfc,:);
figure(3);
imshow(uint8(Iz1));

for c1 = 1:nc
    Icol = Iz1(:,c1);
    if(any(Icol < 190))
        break;
    end
end
Iz2 = Iz1(:,c1:nc);
figure(4);
imshow(uint8(Iz2));

[nc2 nf2] = size(Iz2);
%for c2 = nc2:-1:1
for c2=nf2:-1:1
    Icol = Iz2(:,c2);
    if(any(Icol < 190))
        break;
    end
end
Iz3 = Iz2(:,1:c2);
figure(5);
imshow(uint8(Iz3));
% Pixelamiento en cuadrados de 20x20
[cc ff]=size(Iz3);
cc20=fix(cc/120);
cc_20=cc20*120;
ff20=fix(ff/70);
ff_20=ff20*70;
Izz=Iz3(1:cc_20,1:ff_20);
%Izz = Iz3(1:2400,1:1460);
ii = 1;
for i = 1:cc20:cc_20
    jj = 1;
```

```

    for j = 1:ff20:ff_20
        im = Iz3(i:i+19,j:j+19);
        im = sum(sum(im))/400;
        Inueva20(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
figure(9);
imshow(uint8(Inueva20));
I20=Inueva20;

I20bn= 255*round(I20/255*1.1);
figure(13);
imshow(uint8(I20bn));
save ('carapix_10.mat', 'I20bn');

```

3. Procesamiento de imágenes 3

```

clear all;
close all;
clc;
I1 = imread('cara_1.jpg','jpg');
figure(1);
imshow(uint8(I1));
I2 = I1;
Iz = I1(:,:,2);
[nf nc] = size(Iz);
figure(2);
imshow(uint8(Iz));
for f = 1:nf
    Ifila = Iz(f,:);
    if(any(Ifila < 190)) %190 para no capturar tonos claros de fondo
        break;
    end
end
nfc = 2400;
Iz1 = Iz(f:nfc,:);
figure(3);
imshow(uint8(Iz1));
for c1 = 1:nc
    Icol = Iz1(:,c1);
    if(any(Icol < 190)) %190 para no capturar tonos claros de fondo
        break;
    end
end
Iz2 = Iz1(:,c1:nc);
figure(4);
imshow(uint8(Iz2));
[nc2 nf2] = size(Iz2);
%for c2 = nc2:-1:1
for c2=nf2:-1:1
    Icol = Iz2(:,c2);
    if(any(Icol < 190))
        break;
    end
end
Iz3 = Iz2(:,1:c2);

```

```

figure(5);
imshow(uint8(Iz3));
% Pixelamiento en cuadrados de ixj
[cc ff]=size(Iz3);
cc20=fix(cc/120);
cc_20=cc20*120;
ff20=fix(ff/70);
ff_20=ff20*70;
Izz=Iz3(1:cc_20,1:ff_20);
%Izz = Iz3(1:2400,1:1460);
ii = 1;
for i = 1:cc20:cc_20
    jj = 1;
    for j = 1:ff20:ff_20
        im = Iz3(i:i+19,j:j+19);
        im = sum(sum(im))/400;
        Inueva20(ii,jj) = round(im);
        jj = jj + 1;
    end
    ii = ii + 1;
end
figure(9);
imshow(uint8(Inueva20));
I20=Inueva20;
I20bn= 255*round(I20/255*0.99);%umbral
figure(13);
imshow(uint8(I20bn));
save ('carapix_1.mat', 'I20bn');

```

4. Etapa de prueba

```

clc;
clear all;
close all;

for j=1:4
    for k=1:4
        a = rand;
        if a>=0.1
            A(j,k)=1;

        end
        if a<=0.1
            A(j,k)=0;
        end
    end
end

```

5. Primera neurona – Reconocimiento

```
clear; clc;
close all;
load carapix_1; carapix_1 = I20bn; load carapix_2; carapix_2 = I20bn;
load carapix_3; carapix_3 = I20bn; load carapix_4; carapix_4 = I20bn;
load carapix_5; carapix_5 = I20bn; load carapix_6; carapix_6 = I20bn;
load carapix_7; carapix_7 = I20bn; load carapix_8; carapix_8 = I20bn;
load carapix_9; carapix_9 = I20bn; load carapix_10; carapix_10 = I20bn;
figure(1);
subplot(3,4,1); imshow(carapix_1);
subplot(3,4,2); imshow(carapix_2);
subplot(3,4,3); imshow(carapix_3);
subplot(3,4,4); imshow(carapix_4);
subplot(3,4,5); imshow(carapix_5);
subplot(3,4,6); imshow(carapix_6);
subplot(3,4,7); imshow(carapix_7);
subplot(3,4,8); imshow(carapix_8);
subplot(3,4,10); imshow(carapix_9);
subplot(3,4,11); imshow(carapix_10);
disp('Press ENTER');
pause;
caral = abs((carapix_1-255)/255);
cara2 = abs((carapix_2-255)/255);
cara3 = abs((carapix_3-255)/255);
cara4 = abs((carapix_4-255)/255);
cara5 = abs((carapix_5-255)/255);
cara6 = abs((carapix_6-255)/255);
cara7 = abs((carapix_7-255)/255);
cara8 = abs((carapix_8-255)/255);
cara9 = abs((carapix_9-255)/255);
cara10 = abs((carapix_10-255)/255);
[nf nc] = size(caral);
x(1,:) = caral(1,:);
x(2,:) = cara2(1,:);
x(3,:) = cara3(1,:);
x(4,:) = cara4(1,:);
x(5,:) = cara5(1,:);
x(6,:) = cara6(1,:);
x(7,:) = cara7(1,:);
x(8,:) = cara8(1,:);
x(9,:) = cara9(1,:);
x(10,:) = cara10(1,:);
for k = 2:nf
    caras = [ caral(k,:)
              cara2(k,:)
              cara3(k,:)
              cara4(k,:)
              cara5(k,:)
              cara6(k,:)
              cara7(k,:)
              cara8(k,:)
              cara9(k,:)
              cara10(k,:) ];
    x = [ x caras ];
end
[ nxf nxc ] = size(x);
nx = nxf;
yb(1,:) = [ 1 0 0 0 0 0 0 0 0 0 ];
```

```

yb(2,:) = [ 0 1 0 0 0 0 0 0 0 0 ];
yb(3,:) = [ 0 0 1 0 0 0 0 0 0 0 ];
yb(4,:) = [ 0 0 0 1 0 0 0 0 0 0 ];
yb(5,:) = [ 0 0 0 0 1 0 0 0 0 0 ];
yb(6,:) = [ 0 0 0 0 0 1 0 0 0 0 ];
yb(7,:) = [ 0 0 0 0 0 0 1 0 0 0 ];
yb(8,:) = [ 0 0 0 0 0 0 0 1 0 0 ];
yb(9,:) = [ 0 0 0 0 0 0 0 0 1 0 ];
yb(10,:) = [ 0 0 0 0 0 0 0 0 0 1 ];
[ nyf nyc ] = size(yb);
ny = nyf;
ne = nxc;
nm = 20;
ns = nyc;
bias = input('Bias: SI = 1 : ');
if(bias == 1)
    ne = ne +1;
    x = [ x ones (nx,1) ];
end
v = 0.1*(rand(ne,nm) - 0.5)*2;
w = 0.1*(rand(nm,ns) - 0.5)*2;
% load pesoscaras;
eta = input('eta pesos : ');
for iter = 1:3000
count(iter,1) = iter;
dJdw = 0;
dJdv = 0;
for k = 1:nx
in = (x(k,:))';
m = v'*in;
%n = 1.0./(1+exp(-m)); % Sigmoidea 1
n = 2.0./(1+exp(-m)) - 1; % sigmoidea 2
%n = exp(-m.^2); % Gaussiana
out = w'*n;
y(k,:) = out';
er = out - (yb(k,:))';
error(k,:) = er;
%dndm = n.*(1 - n); % Sigmoidea 1
dndm = (1 - n.*n)/2; % Sigmoidea 2
%dndm = -2.0*(n.*m); % Gaussiana
dydw = n;
dJdw = 1*dJdw + dydw*er';
dJdv = 1*dJdv + in*((w*er).*dndm)';
%w = w - eta*dJdw/nx;
%v = v - eta*dJdv/nx;
end
w = w - eta*dJdw/nx;
v = v - eta*dJdv/nx;
JJ = 0.5*sum(sum(error.*error))
J(iter,1) = JJ;
end
save pesoscarasfotos ns nm v w bias;
figure(2);
plot(y(1,:), 'or'); % Se grafica sin redondeo
hold on;
plot(yb(1,:), '*b');
figure(3);
plot(count,J);

```

6. Segunda neurona – Reconocimiento

```
clear;
clc;
close all;

load cara1pix; load cara2pix; load cara3pix; load cara4pix;
figure(1);
subplot(2,2,1); imshow(cara1pix);
subplot(2,2,2); imshow(cara2pix);
subplot(2,2,3); imshow(cara3pix);
subplot(2,2,4); imshow(cara4pix);
disp('Press ENTER');
pause;
cara1 = abs((cara1pix-255)/255);
cara2 = abs((cara2pix-255)/255);
cara3 = abs((cara3pix-255)/255);
cara4 = abs((cara4pix-255)/255);
% clear cara1pix cara2pix cara3pix cara4pix;

[nf nc] = size(cara1);
x(1,:) = cara1(1,:);
x(2,:) = cara2(1,:);
x(3,:) = cara3(1,:);
x(4,:) = cara4(1,:);
for k = 2:nf
    caras = [ cara1(k,:)
              cara2(k,:)
              cara3(k,:)
              cara4(k,:) ];
    x = [ x caras ];
end
[nxf nxc] = size(x);
nx = nxf;

yb(1,:) = [ 1 0 0 0 ];
yb(2,:) = [ 0 1 0 0 ];
yb(3,:) = [ 0 0 1 0 ];
yb(4,:) = [ 0 0 0 1 ];

[nyf nyc] = size(yb);
ny = nyf;

ne = nxc;
nm = 10;
ns = nyc;
bias = input('Bias: SI = 1 : ');
if(bias == 1)
    ne = ne +1;
    x = [ x ones(nx,1) ];
end
v = 0.2*(rand(ne,nm) - 0.5)*2;
w = 0.2*(rand(nm,ns) - 0.5)*2;
% load pesoscaras;

eta = input('eta pesos : ');
```

```

for iter = 1:2000
count(iter,1) = iter;
dJdw = 0;
dJdv = 0;
for k = 1:nx
in = (x(k,:))';
m = v'*in;
n = 1.0./(1+exp(-m)); % Sigmoidea 1
%n = 2.0./(1+exp(-m)) - 1; % sigmoidea 2
%n = exp(-m.^2); % Gaussiana
out = w'*n;
y(k,:) = out';
er = out - (yb(k,:))';
error(k,:) = er;
dndm = n.*(1 - n); % Sigmoidea 1
%dndm = (1 - n.*n)/2; % Sigmoidea 2
%dndm = -2.0*(n.*m); % Gaussiana
dydw = n;
dJdw = 1*dJdw + dydw*er';
dJdv = 1*dJdv + in*((w*er).*dndm)';
%w = w - eta*dJdw/nx;
%v = v - eta*dJdv/nx;
end
w = w - eta*dJdw/nx;
v = v - eta*dJdv/nx;
JJ = 0.5*sum(sum(error.*error))
J(iter,1) = JJ;
end

save pesoscarasfotos ns nm v w bias;

figure(2);
plot(y(1,:), 'or'); % Se grafica sin redondeo
hold on;
plot(yb(1,:), '*b');

figure(3);
plot(count,J);

```

7. Tercera Neurona – Validación

```

clear;
clc;
close all;
for i=1:10
%i=1+fix(10*rand);
if i==1
load carapix_1;
carapix = I20bn;
elseif i==2
load carapix_2;
carapix = I20bn;
elseif i==3
load carapix_3;
carapix = I20bn;
elseif i==4
load carapix_4;

```

```

        carapix = I20bn;
elseif i==5
    load carapix_5;
    carapix = I20bn;
elseif i==6
    load carapix_6;
    carapix = I20bn;
elseif i==7
    load carapix_7;
    carapix = I20bn;
elseif i==8
    load carapix_8;
    carapix = I20bn;
elseif i==9
    load carapix_9;
    carapix = I20bn;
elseif i==10
    load carapix_10;
    carapix = I20bn;
end

figure(i);
imshow(carapix);
%carapix(35:37,15:17) = zeros(3,3);           % Negro
%carapix(40:41,20:21) = zeros(2,2);         % Negro
%carapix(15,20:35) = zeros(1,16);           % Negro
%carapix(6:7,26:27) = 255*ones(2,2);        % Blanco
%carapix(9:10,18:19) = 255*ones(2,2);      % Blanco
%carapix(:,47:50) = 255*ones(50,4);
%carapix(1:4,:) = 255*ones(4,50);
%carapix(12:14,15:35) = 255*ones(3,21);
carapix = carapix + 255*round(1*rand(120,70));
figure(10+i);
imshow(carapix);
cara = abs((carapix-255)/255);
%
clear carapix;
[nf nc] = size(cara);
x(1,:) = cara(1,:);
for k = 2:nf
    x = [ x cara(k,:) ];
end
[ nxf nxc ] = size(x);
nx = nxf;
ne = nxc;
load pesoscarasfotos; % Carga nm v w bias
if(bias == 1)
    ne = ne +1;
    x = [ x ones(nx,1) ];
end

in = x';
m = v'*in;
n = 1.0./(1+exp(-m)); % Sigmoidea 1
%n = 2.0./(1+exp(-m)) - 1; % sigmoidea 2
%n = exp(-m.^2); % Gaussiana
out = w'*n;
y = out;

```

```

[maxy k] = max(y);
if(k == 1)
    disp('La Cara es 1');
elseif(k == 2)
    disp('La Cara es 2');
elseif(k == 3)
    disp('La Cara es 3');
elseif(k == 4)
    disp('La Cara es 4');
elseif(k == 5)
    disp('La Cara es 5');
elseif(k == 6)
    disp('La Cara es 6');
elseif(k == 7)
    disp('La Cara es 7');
elseif(k == 8)
    disp('La Cara es 8');
elseif(k == 9)
    disp('La Cara es 9');
elseif(k == 10)
    disp('La Cara es 10');
end

%figure(20+i);
%axis([ 0 nf 0 nc ])
%hold on;
%for i = 1:nf
    % for j = 1:nc
        % if(cara(i,j) == 1)
            % plot(j, (nf-i+1), '*b');
        % end

    %end
%end
grid;
clear;
end

```

8. Cuarta neurona – Validación

```

clear;
clc;
close all;
noise= input('porcentaje de ruido(6%) (0-100%): '); % porcentaje de ruido
noise= 1-(noise/100);
for i=1:10
% i=1+fix(10*rand);
if i==1
    load carapix_1;
    carapix = I20bn;
elseif i==2
    load carapix_2;
    carapix = I20bn;
elseif i==3
    load carapix_3;
    carapix = I20bn;
elseif i==4
    load carapix_4;

```

```

        carapix = I20bn;
elseif i==5
    load carapix_5;
    carapix = I20bn;
elseif i==6
    load carapix_6;
    carapix = I20bn;
elseif i==7
    load carapix_7;
    carapix = I20bn;
elseif i==8
    load carapix_8;
    carapix = I20bn;
elseif i==9
    load carapix_9;
    carapix = I20bn;
elseif i==10
    load carapix_10;
    carapix = I20bn;
end

figure(1);
subplot(3,4,i); imshow(carapix);

for j=1:120
    for k=1:70
a = rand;
b = rand;
if a>=noise
A(j,k)=1;
end
if a<=noise
A(j,k)=0;

end
if b>=noise
B(j,k)=0;
end
if b<=noise
    B(j,k)=1;
end
end
end
carapix = carapix + 255.*A; %ajustar ruido
carapix1 = carapix *255.*B;
figure(2);
subplot(3,4,i); imshow(carapix1);
cara = abs((carapix-255)/255);
cara= cara.*B;
%clear carapix;
[nf nc] = size(cara);
x(1,:) = cara(1,:);
for k = 2:nf
    x = [ x cara(k,:) ];
end
[ nxf nxc ] = size(x);
nx = nxf;

```

```

ne = nxc;
load pesoscarasfotos; % Carga nm v w bias
if(bias == 1)
    ne = ne +1;
    x = [ x ones(nx,1) ];
end

in = x';
m = v'*in;
%n = 1.0./(1+exp(-m)); % Sigmoidea 1
n = 2.0./(1+exp(-m)) - 1; % sigmoidea 2
%n = exp(-m.^2); % Gaussiana
out = w'*n;
y = out;
[maxy k] = max(abs(y));
if(k == 1)
    disp('La Cara es 1');
elseif(k == 2)
    disp('La Cara es 2');
elseif(k == 3)
    disp('La Cara es 3');
elseif(k == 4)
    disp('La Cara es 4');
elseif(k == 5)
    disp('La Cara es 5');
elseif(k == 6)
    disp('La Cara es 6');
elseif(k == 7)
    disp('La Cara es 7');
elseif(k == 8)
    disp('La Cara es 8');
elseif(k == 9)
    disp('La Cara es 9');
elseif(k == 10)
    disp('La Cara es 10');
end

%figure(20+i);
%axis([ 0 nf 0 nc ])
%hold on;
%for i = 1:nf
    % for j = 1:nc
        % if(cara(i,j) == 1)
            % plot(j, (nf-i+1), '*b');
        % end

    %end
%end
grid;
clear I20bn;
clear cara;
clear x;
end

```