



**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E  
INDUSTRIAL  
CARRERA DE INGENIERÍA INDUSTRIAL EN PROCESOS DE  
AUTOMATIZACIÓN**

**TEMA**

---

**“CONTROL DE LA GESTICULACIÓN DE UN ROBOT SOCIAL  
INTERACTIVO CON ASPECTO HUMANOIDE”**

---

Proyecto de Trabajo de Graduación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero Industrial en Procesos de Automatización.

**SUBLÍNEA DE INVESTIGACIÓN:** Sistemas de control automatizados e instrumentación virtual para procesos industriales de baja y alta potencia.

**AUTOR:** Edison Mauricio Arias Infante

**TUTOR:** Ing. Mg. Patricio Encalada.

Ambato – Ecuador

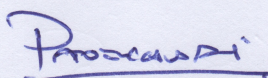
Junio – 2018

## CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor del Trabajo de Investigación sobre el tema: “CONTROL DE LA GESTICULACIÓN DE UN ROBOT SOCIAL INTERACTIVO CON ASPECTO HUMANOIDE”, elaborado por el señor Edison Mauricio Arias Infante, estudiante de la Carrera de Ingeniería Industrial en Procesos de Automatización, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, considero que el informe investigativo reúne los requisitos y méritos suficientes para que continúe con los trámites y consiguiente aprobación de conformidad con el numeral 7.2 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.

Ambato junio, 2018

EL TUTOR



---

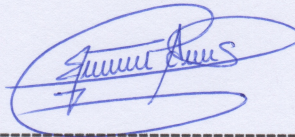
Ing. Mg. Patricio Encalada

## **AUTORÍA**

El presente Proyecto de Investigación titulado: “CONTROL DE LA GESTICULACIÓN DE UN ROBOT SOCIAL INTERACTIVO CON ASPECTO HUMANOIDE”, es de mi autoría, siendo el mismo personal, original y único, por tal razón el contenido, comentarios, conclusiones, recomendaciones y efectos legales son de exclusiva responsabilidad del autor.

Ambato junio, 2018

## **AUTOR**



---

Edisson Mauricio Arias Infante

CC: 180433464-5

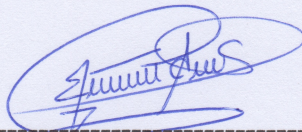
## **DERECHOS DE AUTOR**

Autorizó a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación, con fines de difusión pública, además autorizo su reproducción dentro de las regulaciones de la Universidad.

Ambato junio, 2018

**AUTOR**



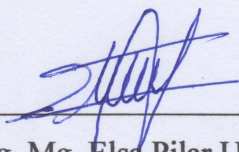
---

Edisson Mauricio Arias Infante

CC: 180433464-5

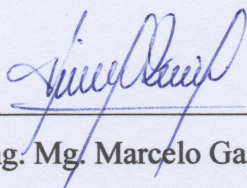
## APROBACIÓN DEL TRIBUNAL DE GRADO

La Comisión del Tribunal de Grado conformada por los señores docentes calificadores, revisó y aprobó el Informe Final del Proyecto de Investigación titulado “CONTROL DE LA GESTICULACIÓN DE UN ROBOT SOCIAL INTERACTIVO CON ASPECTO HUMANOIDE” presentado por el señor Edison Mauricio Arias Infante de acuerdo al numeral 9.1 de los Lineamientos Generales para la aplicación de Instructivos de las Modalidades de Titulación de las Facultades de la Universidad Técnica de Ambato.



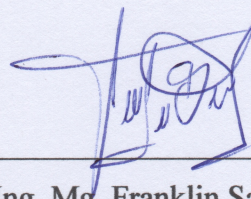
---

Ing. Mg. Elsa Pilar Urrutia Urrutia  
PRESIDENTE DEL TRIBUNAL



---

Ing. Mg. Marcelo García  
DOCENTE CALIFICADOR



---

Ing. Mg. Franklin Salazar  
DOCENTE CALIFICADOR

## **DEDICATORIA:**

*Dedico este trabajo principalmente a Dios por brindarme salud, fortaleza y sabiduría en el transcurso de esta etapa de mi vida y permitirme alcanzar tan anhelado objetivo para la satisfacción familiar y personal*

*A mi madre por demostrarme siempre su cariño, amor, comprensión y su apoyo incondicional que me impulsaron a seguir a delante y nunca rendirme; a mi padre que me ha apoyado en innumerables ocasiones; a mis queridos y amados abuelos por estar siempre a mi lado en los buenos y malos momentos, y ser partícipes de mi formación profesional, a mis hermanos que me han ofrecido amor y han permitido que seamos una familia unida.*

*A mi preciado y amado tío por sus consejos y su apoyo que fortalecieron mi corazón y me ayudaron a nunca rendirme ante las adversidades.*

*A todos mis demás familiares que de alguna manera u otra me han guiado para seguir siempre a delante.*

*Edisson Mauricio Arias Infante*

## **AGRADECIMIENTO:**

*A Dios por protegerme dándome salud, fortaleza y la oportunidad de formar parte de una familia tan maravillosa.*

*A mi madre y abuelos que con su esfuerzo diario permitieron que no me faltara nada durante mi ciclo universitario; a mi tío por su amistad y apoyo para seguir adelante; a mis hermanos por siempre darme fuerzas y el aliento para continuar.*

*Al Ing. Patricio Encalada por su amistad, apoyo y conocimientos que permitieron la culminación de este proyecto.*

*A la Universidad Técnica de Ambato por permitirme formar parte de tan prestigiosa institución y por ende adquirir las enseñanzas, valores y consejos de sus docentes.*

*Edisson Mauricio Arias Infante*

## ÍNDICE DE CONTENIDO

CERTIFICACIÓN DEL TUTOR.....	I
AUTORÍA.....	II
DERECHOS DE AUTOR.....	III
APROBACIÓN DEL TRIBUNAL DE GRADO.....	IV
DEDICATORIA:.....	V
AGRADECIMIENTO:.....	VI
RESUMEN.....	XIII
ABSTRACT.....	XIV
INTRODUCCIÓN.....	XV
CAPITULO I.....	1
EL PROBLEMA.....	1
1.1. Tema.....	1
1.2. Planteamiento del problema.....	1
1.3. Delimitación.....	2
1.3.1. Delimitación de contenido.....	2
1.3.2. Delimitación espacial:.....	3
1.3.3. Delimitación temporal:.....	3
1.4. Justificación.....	3
1.5. Objetivos.....	4
1.5.1. Objetivo General.....	4
1.5.2. Objetivos Específicos.....	4
CAPITULO II.....	5
MARCO TEÓRICO.....	5
2.1. Antecedentes Investigativos.....	5
2.2. Fundamentación Teórica.....	7
2.2.1. Robótica Social.....	7



2.2.2.	Valle Inquietante .....	8
2.2.3.	Sintetizadores de Audio .....	11
2.2.4.	Inteligencia Artificial .....	13
2.3.	Propuesta de Solución .....	22
CAPÍTULO III .....		23
METODOLOGÍA .....		23
3.1.	Modalidad de Investigación .....	23
3.1.1.	Investigación Bibliográfica-Documental .....	23
3.1.2.	Investigación experimental .....	23
3.2.	Población y Muestra .....	23
3.3.	Recolección de información .....	24
3.4.	Procesamiento y análisis de datos .....	24
3.5.	Desarrollo del Proyecto .....	24
CAPÍTULO IV .....		26
DESARROLLO DE LA PROPUESTA .....		26
4.1.	Análisis de factibilidad .....	27
4.1.1.	Factibilidad técnica .....	27
4.1.2.	Factibilidad económica .....	27
4.2.	Análisis de requerimientos .....	27
4.2.1.	Hardware, sensores y actuadores .....	27
4.2.2.	Software .....	30
4.3.	Diseño y construcción del robot interactivo social .....	34
4.3.1.	Diseño de piezas .....	34
4.3.2.	Sistema eléctrico y electrónico .....	40
4.4.	Gesticulación del robot interactivo social .....	41
4.4.1.	Visión .....	42
4.4.2.	Clasificación .....	47
4.4.3.	Movimiento del robot .....	55
4.4.4.	Pruebas y resultados .....	57
4.5.	Sistema de sincronización labial .....	63
4.5.1.	Convertidor de texto a voz .....	63
4.5.2.	Base de datos visemas .....	65

4.5.3.	Funciones del algoritmo de sincronización labial.....	65
4.5.4.	Pruebas y resultados.....	70
CAPITULO V.....		74
CONCLUSIONES Y RECOMENDACIONES .....		74
5.1.	Conclusiones .....	74
5.2.	Recomendaciones .....	75
BIBLIOGRAFÍA .....		77
ANEXOS .....		81
ANEXO A.....		82
Diagrama nodal de funcionamiento del robot social interactivo. ....		82
ANEXO B.....		83
Posición de servomotores – gestos faciales .....		83
ANEXO C.....		84
Codificación de Visemas y sílabas .....		84
ANEXO D.....		85
Posición de los servomotores – visemas.....		85
ANEXO E.....		86
Script de envío de datos a la tarjeta controladora. ....		86
ANEXO F .....		88
Script de reconocimiento de gestos faciales. ....		88
ANEXO G.....		90
Script de sincronización labial. ....		90
ANEXO H.....		92
Planos.....		92

## ÍNDICE DE TABLAS

Tabla 2.2.4.1. Principales campos de aplicación de la inteligencia artificial. ....	16
Tabla 2.2.4.2 Modelos de redes neuronales convolucionales desarrollados [29]. ....	20
Tabla 4.2.1.1 Características eléctricas de la placa controladora SCC-32.....	28
Tabla 4.2.1.2 Características de los actuadores tipo servomotor.....	29

Tabla 4.2.1.3 Características del sensor de imagen. ....	30
Tabla 4.3.1.1 Configuración de los músculos-grados de libertad.....	38
Tabla 4.4.1.1 Funciones de OpenCV - módulo de adquisición de imagen.....	43
Tabla 4.4.1.2 Funciones del módulo de pre-procesamiento. ....	43
Tabla 4.4.1.3 Funciones utilizadas en la etapa de reconocimiento de rostros. ....	47
Tabla 4.4.2.1. Función para compilación del modelo CNN .....	53
Tabla 4.4.4.1. Resultados del controlador de gesticulación del robot social. ....	60
Tabla 4.4.4.2. Movimientos para el gesto enojo. ....	62
Tabla 4.5.1.1 Característica de los conversores de texto a voz disponible. ....	64
Tabla 4.5.3.1. Movimientos de visemas obtenidos. ....	71

## ÍNDICE DE FIGURAS

Figura. 2.2.1.1 iCub y Nexi, Robots sociales [14]. ....	8
Figura. 2.2.1.2 Aplicaciones de los robots sociales. ....	8
Figura. 2.2.2.1 Telenoid R1 [18]. ....	9
Figura. 2.2.2.2 Ejemplos de imágenes del fenómeno Pareidolia [20]. ....	10
Figura. 2.2.2.3 Valle inquietante propuesto por Masahiro Mori [19]. ....	10
Figura. 2.2.3.1 Diagrama de bloques básico de un conversor de texto a voz.....	12
Figura. 2.2.3.2 Diferentes visema para pronunciar s. ....	13
Figura. 2.2.4.1 Robot humanoide ASIMO una aplicación de la AI. ....	13
Figura. 2.2.4.2 Robot SpotMini-Boston Dynamics. ....	15
Figura. 2.2.4.3 Características de las redes neuronales artificiales [28]. ....	17
Figura. 2.2.4.4 Modelo de neurona artificial [27]. ....	18
Figura. 2.2.4.5 Estructura de una red neuronal LeNet-5 [28]. ....	19
Figura. 4.2.1.1 Tarjeta controladora de servomotores SSC-32 [31]. ....	28
Figura. 4.3.1.1 Robot Humanoide INMOOV [37]. ....	34
Figura. 4.3.1.2 Estructura de la cabeza de INMOOV [37]. ....	35
Figura. 4.3.1.3 Estructura del cuello de INMOOV [37]. ....	35
Figura. 4.3.1.4 Mecanismo de ojo [19]. ....	36
Figura. 4.3.1.5. Mecanismo para el movimiento de ojos y parpados. ....	37
Figura. 4.3.1.6 Mecanismo para el movimiento de cejas. ....	37

Figura. 4.3.1.7 Estructura labial del robot social DORIS [21].	38
Figura. 4.3.1.8. Mecanismo para el movimiento de labios del robot social.	39
Figura. 4.3.1.9 Ensamble final del robot interactivo social.	39
Figura. 4.3.2.1 Sistema eléctrico y electrónico del robot.	40
Figura. 4.4.1 Sistema de gesticulación del robot social.	41
Figura. 4.4.1.1 Diagrama de flujo del módulo de Visión.	42
Figura. 4.4.1.2 Estructura del algoritmo de Viola-Jones [38].	44
Figura. 4.4.1.3 Ejemplos de características de dos, tres y cuatro rectángulos.	45
Figura. 4.4.1.4 Estructura de un clasificador tipo cascada.	46
Figura. 4.4.2.1 Imágenes de la base de datos Fer2013 [40].	48
Figura. 4.4.2.2 Descripción de Fer2013.	49
Figura. 4.4.2.3 Distribución de los datos de entrenamiento y test.	49
Figura. 4.4.2.4 Función para la creación de gráficos de barras.	49
Figura. 4.4.2.5 Distribución de datos modificados.	50
Figura. 4.4.2.6 Estructura de la red neuronal convolucional.	50
Figura. 4.4.2.7 Capas asignadas a la red neuronal convolucional (Keras-Python)...	52
Figura. 4.4.2.8 Descripción de la red neuronal convolucional (CCN) en Keras.	54
Figura. 4.4.2.9 Gráficas del comportamiento de la red neuronal.	55
Figura. 4.4.2.10 Resultados de la evaluación del modelo.	55
Figura. 4.4.3.1 Diagrama de flujo del módulo movimiento del robot.	56
Figura. 4.4.3.2 Código para escribir en el puerto serial.	56
Figura. 4.4.4.1 Resultados obtenidos de la predicción del modelo.	57
Figura. 4.4.4.2 Seudocódigo de matriz de confusión.	58
Figura. 4.4.4.3 Matriz de Confusión del modelo de red neuronal.	59
Figura. 4.4.4.4. Datos del controlador de la gesticulación del robot.	61
Figura. 4.5.1 Esquema secuencial de sincronización labial del robot social.	63
Figura. 4.5.1.1 Código del conversor de texto a voz Mac OS TTS.	65
Figura. 4.5.2.1 Estructura del visema [19].	65
Figura. 4.5.3.1 Diagrama de flujo del algoritmo de sincronización labial.	66
Figura. 4.5.3.2 Diagrama de flujo del conversor de voz a texto.	67
Figura. 4.5.3.3 Diagrama de flujo del subproceso Normalización del texto.	67
Figura. 4.5.3.4 Diagrama de flujo de la función división silábica.	68
Figura. 4.5.3.5 Diagrama de flujo del conversor de texto a voz.	69

Figura. 4.5.3.6 Diagrama de Flujo de la función Asignación de Visemas .....	69
Figura. 4.5.3.7 Diagrama de flujo de la función movimiento de servomotores. ....	70
Figura. 4.5.4.1 Movimientos de servomotores. ....	72
Figura. 4.5.4.2 Señal de salida de audio (voz robot) .....	72
Figura. 4.5.4.3 Comportamiento de los servomotores durante la sincronización.....	73

## RESUMEN

En la actualidad, los robots están cada vez más cerca de convertirse en nuestros compañeros de trabajo, sin embargo, todavía carecen de habilidades de interacción social similares a las humanas. El presente proyecto de investigación consiste en el control de gesticulación de un robot social interactivo con aspecto humanoide, comprendido de 3 segmentos como son; construcción del robot social, control gesticular del robot mediante el reconocimiento de los gestos faciales y control del habla por medio de un sistema de sincronización labial.

La estructura del robot social está basada en la estructura del robot humanoide InMoov, del cual se utilizaron las partes del torso y cabeza, y mediante el uso de herramientas como el diseño e impresión 3D se dotaron de varios mecanismos ausentes en la estructura original del robot para generar los movimientos de ojos, parpados, cejas y labios.

El control gesticular se basa en la identificación de los gestos faciales que genera el usuario, datos que son recopilados por medio de la visión artificial, luego estos datos son procesados por el submódulo de tratamiento de imágenes, posteriormente son enviados a un modelo de red neuronal convolucional, empleadas como clasificadores de patrones en el aprendizaje profundo (Deep Learning), para finalmente seleccionar y enviar la sentencia de movimientos al robot.

La sincronización labial radica en una coordinación simultánea entre el movimiento de la boca del robot y el audio del texto proporcionado por el usuario para que el robot pueda comunicarse con él de manera casi natural.

A través del desarrollo e implementación del controlador de gesticulación al robot social interactivo diseñado, se pretende infundir nuevos conceptos en cuanto a controladores basados en redes neuronales, así como también alentar a estudiantes, instructores, docentes e investigadores al estudio en el campo de la robótica social

## **ABSTRACT**

At the present time, robots are increasingly close to becoming in our co-workers, however, they still lack social interaction skills similar to human ones. The present research project consists in the gesticulation control of an interactive social robot with human aspect, comprised of 3 segments; construction of the robot, robot gestural control through of the facial gestures recognition and speech control through of a lip-sync system.

The social robot structure is based in the InMoov humanoid robot structure, of which is used the parts of the head and torso, and by means of use the tools like design and 3D printing were equipped with several mechanisms absent in the original robot structure to generate the movements of eyes, eyelids, eyebrows and lips.

The gestural control is based on the facial gestures identification generated by the user, data that are collected by the means of artificial vision, then these data are processed by the image processing sub-module, subsequently are sent to a convolutional neural network model, used as pattern classifiers in deep learning, to finally select and send the sentence of movements to the robot.

The lip-sync lies in a simultaneous coordination between the robot mouth move and the audio of text provided by the user so that the robot can communicate with him in almost natural way.

Through of the development and implementation of the gesticulation controller to the social interactive robot, it is aimed to infuse new concepts regarding controllers based in neural networks, as well as encourage to students, instructors, teachers and researchers to study the social robotics field.

## INTRODUCCIÓN

El presente proyecto se encuentra estructurado por cinco capítulos, detallados de la siguiente manera:

En el Capítulo I, se presentan las razones de importancia para el desarrollo de esta investigación partiendo de un análisis macro, meso y micro hasta el planteamiento del justificativo que sustente el desarrollo del mismo y los objetivos a satisfacer.

En el Capítulo II, se puntualiza una breve introducción acerca de los robots interactivos sociales, donde se describe rápidamente parte de su historia y el avance tecnológico que estos han tenido hasta la actualidad. Además, en este capítulo se plantea y describe de forma general todo lo concerniente a las tecnologías de control de robots sociales, así como también establecer un conocimiento base para el diseño, construcción y control del robot social interactivo a desarrollar.

El Capítulo III está constituido por la información de la metodología empleada para desarrollar la totalidad del proyecto.

En el Capítulo IV, se describe el proceso llevado a cabo para el diseño, construcción y control del robot social interactivo, partiendo desde la descripción del hardware y software empleados como base fundamental para el desarrollo, los métodos utilizados en el diseño y modelado de piezas para el robot, y además se presenta el proceso sistémico para la elaboración de los controladores de gesticulación y habla del robot social.

En el Capítulo V, se cita las conclusiones y recomendaciones que se derivan del desarrollo del trabajo propuesto. Finalmente, se presentan la lista de referencias consultadas y anexos



# **CAPITULO I**

## **EL PROBLEMA**

### **1.1. Tema**

“CONTROL DE LA GESTICULACIÓN DE UN ROBOT SOCIAL INTERACTIVO CON ASPECTO HUMANOIDE”

### **1.2. Planteamiento del problema**

El avance de la ingeniería informática, inteligencia artificial y redes neuronales artificiales, ha logrado ampliar el mundo de posibilidades para el diseño y desarrollo de robots. Hoy en día existe un sin número de robots destinados a realizar tareas específicas tanto en el campo industrial como en el de servicio. Entre las ramas de la robótica se halla la robótica social, que tiene por objetivo el estudio del comportamiento y diseño de los robots para ayudar a las personas en tareas o actividades que son costosas, repetitivas y peligrosas para los humanos.

Muchas de las aplicaciones que presentan los robots sociales requieren interacciones frecuentes e íntimas con los usuarios humanos principalmente a través de comunicaciones verbales asistidas por elementos no verbales. La interacción humano-robot es el ingrediente tecnológico más importante para tales aplicaciones. Como ejemplos de aplicación se pueden incluir orientación al cliente, atención cognitiva y emocional, educación y servicios a domicilio [1].

Lograr que los seres humanos interactúen con los robots es una tarea compleja que hasta la actualidad no se ha logrado, pero que es necesaria para compensar el crecimiento de las aplicaciones que tienen los robots sociales en el mundo. Requiere

una comprensión de la conducta social humana, y requiere que los robots utilicen comportamientos apropiados, adaptativos y contingentes para formar y mantener estas interacciones sociales, a más de ser capaces de superar el valle inquietante [2].

En el mundo, la robótica social ha dado un gran avance en la creación de robots sociales semejantes a los seres humanos, los cuales pueden efectuar varias aplicaciones, como expresar gestos y conversar de forma casi forma natural como lo hacen los humanos, un ejemplo de estos es Sophia un robot social de naturaleza femenina desarrollada por Hanson Robotics [3], otro aportación es la robot Jia – Jia desarrollado en la Universidad de Ciencia y Tecnología de China (USTC), el cual es capaz de comunicarse de manera básica, interactuar con personas cercanas y usar expresiones faciales naturales [4].

En el Ecuador, existe una cantidad no muy extensa de información en cuanto al diseño y construcción de robots sociales interactivos, entre los que podemos mencionar: K-Lu-K, un rostro robótica que posee apariencia humana y ejecuta gestos y emociones controladas de manera inalámbrica [5], HENSAR un robot con representación lo más cercana a las expresiones humanas desarrollado en la Escuela Politécnica Nacional [6]; los cuales presentan algoritmos pre-programados para interactuar con los usuarios.

Al examinar en el repositorio, laboratorios y el club de robótica pertenecientes a la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, temas relevantes al proyecto de investigación propuesto, de los cuales no existen registros ni publicaciones relacionadas al mismo.

### **1.3. Delimitación**

#### **1.3.1. Delimitación de contenido**

**Área académica:** Electrónica

**Línea de investigación:** Automatización

**Sublínea de investigación:** Sistemas de control automatizados e instrumentación virtual para procesos industriales de baja y alta potencia.

### **1.3.2. Delimitación espacial:**

El presente proyecto de investigación se desarrollará como un prototipo en las instalaciones de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

### **1.3.3. Delimitación temporal:**

El presente proyecto de investigación se realizará previo a la aprobación del perfil por parte del Honorable Consejo Directivo de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial hasta la culminación de un periodo académico.

## **1.4. Justificación**

El siguiente trabajo de investigación es de **interés** e innovación en el campo tecnológico, para el desarrollo de aplicaciones didácticas en las que se pueda aprovechar el diseño mecánico y electrónico.

El diseño del robot social interactivo tiene una gran **importancia** en el desarrollo de mecanismos para el movimiento, la generación de algoritmos avanzados de control y toma de decisiones con el uso de inteligencia artificial y redes neuronales artificiales en el campo de la robótica social de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, en virtud que permita el estudio estructural del robot aplicando nuevos conocimientos para el aumento de la autonomía del mismo.

Ante la temática de rostro robótico social interactivo con aspecto humanoide, el diseño y construcción de un robot con dichas características **beneficiará** a estudiantes, docentes e investigadores que conforman la Carrera de Ingeniería Industrial en Procesos de Automatización y áreas afines, puedan tomarlo como un modelo para futuras investigaciones formativas y generativas de nuevas aplicaciones del mismo.

El robot social interactivo causará **impacto**, al ser el primer prototipo acoplado a su forma irregular de diseño, aportará el desarrollo de nuevos criterios para el diseño y construcción de prototipos inicialmente no considerados.

La necesidad de interacción hombre-robot y con el avance tecnológico de los robots sociales en tareas domésticas y de educación, como ejemplo se puede mencionar a RobotKind, un robot interactivo social que el objetivo de apoyar y promover la enseñanza-aprendizaje en los niños con trastornos del espectro autista (TEA) [7], para lo cual el hombre debe estar preparado psicológicamente para dicha interacción. El presente trabajo de investigación es **factible**, puesto que se cuenta con los conocimientos necesarios para su desarrollo, además de poseer una amplia bibliografía física y electrónica, recursos humanos y tecnológicos; los cuales son económicamente asequibles.

## **1.5. Objetivos**

### **1.5.1. Objetivo General**

Diseñar y construir un robot social interactivo que ejecute gestos y emociones acordes a las generadas por los seres humanos.

### **1.5.2. Objetivos Específicos**

- ✓ Diseñar y construir los mecanismos para movimiento del robot, además implementar los sistemas eléctricos y electrónicos necesarios para el control del mismo.
- ✓ Diseñar el controlador de gestos y expresiones mediante el uso de redes neuronales artificiales.
- ✓ Diseñar el controlador del habla y movimientos labiales bajo la utilización de visemas para el idioma español.
- ✓ Realizar las pruebas necesarias de interacción humano-robot que ayuden a la detección de ajustes requeridos en el sistema.

## CAPITULO II

### MARCO TEÓRICO

#### 2.1. Antecedentes Investigativos

En los repositorios de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial y en la biblioteca de la misma, después de revisar distintas fuentes bibliográficas como: Tesis, artículos científicos, revistas científicas no se encontraron trabajos de investigación similares. Sin embargo, indagando en el repositorio del departamento de Ciencias de la Energía y Mecánica de la Universidad de las Fuerzas Armadas, se encontró un trabajo desarrollado en el año 2015 titulado: “Diseño y Construcción de un rostro robótico de apariencia humana, con capacidad de ejecutar gestos y emociones”. Elaborado por el Sr. Carlos David Ruales Cortés, el cual indica el siguiente resumen [8]:

“El rostro robótico es una entidad mecánica que posee una apariencia semejante al ser humano, y es considerado como un humanoide, dado a la característica esencial de imitar los gestos y emociones que reproduce el hombre común.”

Del mismo modo, se tomó como referencia el siguiente artículo científico: “Desing of a Human-like Robot with Emotions”. Que traducido al español significa: “Diseño de una Cabeza Humana Robot con Emociones”. El cual presenta una revisión bibliográfica sobre el diseño del mecanismo de movimiento de la cabeza del robot, además incluye las características del mecanismo para el desplazamiento del ojo, mecanismo para el movimiento del cuello además de su apariencia humana [9].

Otro artículo científico publicado en el año 2015 bajo el nombre: “Design of a humanoid robot head for studying human-robot interaction”. Traducido al idioma español: “Diseño de una cabeza robot humanoide para estudiar la interacción humano robot”. El cual detalla los sistemas y subsistemas empleados para la reproducción de las expresiones humanas, a más de la identificación de objetos humanos presentes en su rango de visión [10].

El artículo científico con la temática: “Mirroring and recognizing emotions through facial expressions for a RoboKind platform”. Que traducido al español significa: “Reflejo y reconocimiento de emociones a través de expresiones faciales para la plataforma RoboKind”. El cual presenta información bibliográfica de una metodología para la interacción humano-robot con el desarrollo y aplicación de tecnologías interactivas para apoyar y promover nuevos enfoques de enseñanza-aprendizaje adaptativo para niños con trastornos del espectro autista (TEA) [7].

En el año 2016 Michelle Carolina Mejía Silva y Jennifer Alexandra Núñez Arroba de la Universidad de las Fuerzas Armadas realizaron el proyecto denominado: “Diseño y Construcción de un Robot Social para la Interacción Hombre-Máquina”. El cual trata sobre el diseño y construcción de los mecanismos del robot social para la interacción hombre-robot que responda a los principios de la robótica social, además de dotarle la capacidad de interactuar, comunicarse mediante expresiones, movimientos y sonidos [11].

Haciendo referencia a una aportación clásica, la cual ha sido hincapié para la publicación de un sin número de artículos científicos dedicados al estudio de los robots sociales; esta reseña habla del muy conocido robot Kismet, un robot social diseñado para apreciar cambio en las señales sociales naturales de los canales visuales y auditivos, y para enviar señales de carácter social al cuidador humano a través de la dirección de la mirada, la expresión facial, la postura corporal y las vocalizaciones. Por lo tanto cada aspecto del diseño está encaminado a que el robot experimente interpretar y enviar señales sociales comprensibles al cuidador humano [12].

## **2.2. Fundamentación Teórica**

### **2.2.1. Robótica Social**

La robótica social, una importante rama de la robótica, ha captado la atención de muchas disciplinas, como la visión por ordenador, la inteligencia artificial y la mecatrónica. Con el paso del tiempo se han desarrollado una serie de robots sociales, sin tener aún una clara definición de robot social y los diferentes autores la han definido desde distintas perspectivas. Por ejemplo, Bartneck and Forlizzi et al. [13] manifiestan que un robot social es un robot autónomo o semiautónomo que interactúa con los humanos siguiendo algunos comportamientos sociales; Hegel et al. [14] señala que un robot social es una combinación de un robot y una interfaz social. Existen algunas diferencias entre estas definiciones, pero se las considera mínimas puesto que no se alejan de la idea principal. Por consiguiente y tomando en consideración las aportaciones mencionadas se define a un robot social de la siguiente manera [15]:

"Un robot social es un robot que puede realizar tareas designadas, y la condición necesaria para convertir un robot en un robot social es la capacidad de interactuar con los seres humanos mediante la adhesión a ciertas señales y reglas sociales".

Otra aportación a la definición de robot social, menciona que los robots sociales son robots que interactúan con las personas y no entre ellos. Se asigna esta característica a los robots, puesto que son construidos con aplicaciones que incluyen en la interacción humano-robot en actividades de carácter cotidiano como: el cuidado de personas adultas, terapias, entretenimiento, publicidad, entre otras [11]. En la Figura 2.2.1.1 se puede observar ejemplares de estas aportaciones.

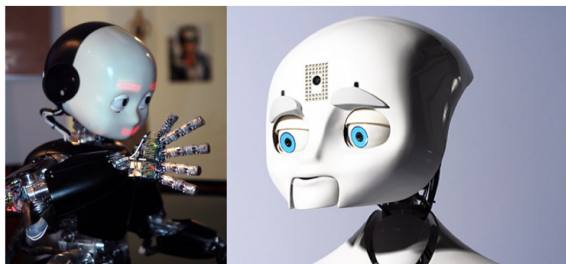


Figura. 2.2.1.1 iCub y Nexi, Robots sociales [16].

Con el progreso de la robótica y las técnicas de computación relacionadas al desarrollo de controladores, los robots sociales se están integrando con mayor frecuencia en la vida cotidiana de las personas. Se los puede emplear en diferentes áreas como: investigación, terapia para el autismo, educación y aplicaciones domésticas, actividades en las que se requiere una correcta interacción con los seres humanos compartiendo la atención y respondiendo a las señales sociales humanas conscientes e inconscientes tales como expresiones faciales, tono de voz, habla y gestos [17]. Varias de las aplicaciones en las que se ven inmersos los robots sociales se detallan en la Figura 2.2.1.2.



Figura. 2.2.1.2 Aplicaciones de los robots sociales.

### 2.2.2. Valle Inquietante

El valle inquietante o inexplicable, también conocido como Uncanny Valley es conocido como una hipótesis sobre las respuestas emocionales que presentan los seres humanos hacia los robots o cualquier otra entidad que se asemeje de manera casi realista a un ser humano o parecido a este. En 1970, el experto japonés en robótica Masahiro Mori quien ha estudiado este fenómeno, percibió que cuanto más los robots son parecidos a los seres humanos, el número de



personas que se sentían atraídas por ellos aumentaba, mientras que si un robot se presentaba con una caracterización alta de realismo y naturalidad referente a un humano, de repente la gente mostraba un alto índice de repulsión y disgusto hacia él [18].

Para obtener una idea más clara del concepto de valle inquietante podemos observar a Telenoid R1 en la Figura 2.2.2.1 un robot cuyo diseño es minimalista, de un tamaño cercano al de un niño pequeño. Con éste diseño, el profesor Hiroshi Ishiguro vence el miedo a explorar las profundidades de Valle Inquietante [19].



Figura. 2.2.2.1 Telenoid R1 [20].

#### **A. Interpretación del mecanismo emocional que estimula el valle inquietante.**

El ser humano a temprana edad empieza a integrarse en su hábitat, es capaz de diferenciar rostros de su especie con una alta eficiencia, al mismo tiempo de distinguir los gestos y relacionarlos con los estados de ánimo. Gracias a este instinto el ser humano es caracterizado como un individuo social y colaborativo con sus semejantes. En nuestro cerebro se produce un fenómeno psicológico denominado pareidolia, en el cual un estímulo vago y aleatorio percibe imágenes de rostros humanos donde no existen [21], la Figura 2.2.2.2 muestra ejemplos de imágenes del fenómeno pareidolia.



Figura. 2.2.2.2 Ejemplos de imágenes del fenómeno Pareidolia [22].

La Figura 2.2.2.3 hace referencia al grado de aceptación o repulsión que tiene el ser humano hacia las máquinas, partiendo de un ejemplo, en el cual hace énfasis en la interacción desde una perspectiva a nivel industrial (robot industrial) hasta el propio ser humano en excelentes condiciones de salud, no existe linealidad en el desplazamiento que tiene el nivel de aceptación, por lo que se produce una caída significativa cuando el objeto empieza a tomar característica más realistas del ser humano, en esta sección denominada valle se encuentran robots o animaciones los cuales a primera vista son espeluznantes [21].

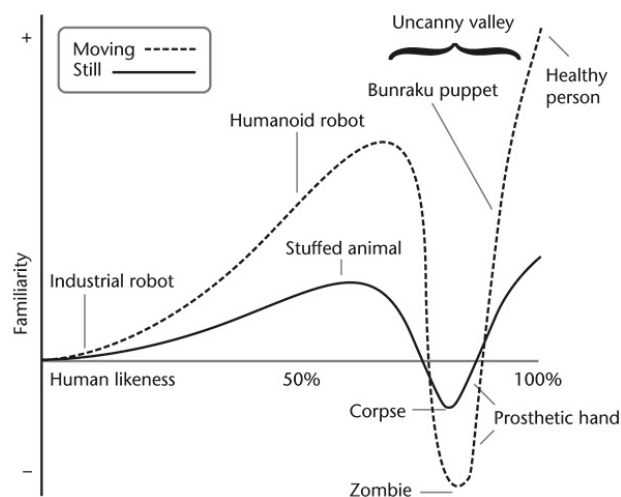


Figura. 2.2.2.3 Valle inquietante propuesto por Masahiro Mori [21].

### 2.2.3. Sintetizadores de Audio

#### A. Conversor Texto a Voz (CTV o TTS)

La tecnología de TTS es la capacidad de un sistema para reproducir un texto impreso en lenguaje hablado. Es decir, son sistemas que realizan la conversión de textos escritos a una voz sintética. Esta nueva forma de comunicación para las personas con capacidades visuales disminuidas o nulas está dada por la accesibilidad fehaciente que posee el utilizar este tipo de herramientas; su practicidad en la conversión de libros en audios o el uso de libros digitales con TTS por sobre los libros en braille; la difusión que adquirió esta herramienta en el ámbito web ha dado como resultado un acceso mucho mayor para los usuarios, aunque todavía estas herramientas no se han incorporado a los repositorios en forma masiva. Otra característica a destacar es que la lectura de los libros digitales utilizando TTS es fácil y rápida, pues sólo se debe manejar requerimientos técnicos básicos de las aplicaciones para poder utilizarlas [23]. Entre los principales conversores tenemos:

- ✓ Loquendo
- ✓ Festival
- ✓ DECTalk
- ✓ BabTTS
- ✓ GoogleTTS
- ✓ Parallax EMIC 2, posee dos analizadores embebidos (DECTalk y EPSON) en un solo hardware.

Breazeal Cynthia en su trabajo llamado Design Sociable Robots resume el uso del sintetizador DECTalk en Kismet [12], en el cual asume la importancia de ciertos parámetros que debe tener el sintetizador para expresar emociones creíbles por medio del sonido hablado, entre ellos se tiene:

- ✓ Velocidad de hablado (palabras por minuto)
- ✓ Cambio de velocidad Elección de tono Intensidad
- ✓ Calidad de voz
- ✓ Control de parada y pausas

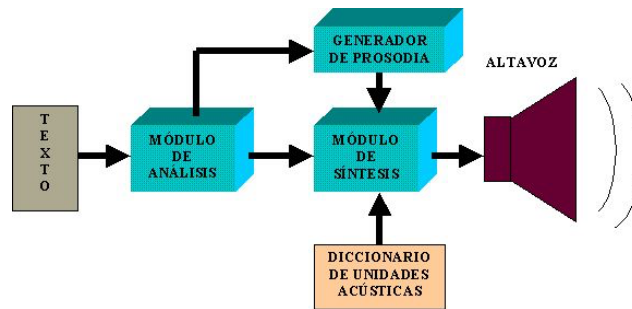


Figura. 2.2.3.1 Diagrama de bloques básico de un convertor de texto a voz

## B. Visemas

Se han propuesto varias aplicaciones en el campo del procesamiento de señales audio y visual a lo largo de los últimos cuarenta años aproximadamente, entre las que se encuentran la cabeza parlante, la lectura de labios, sincronización de los labios, formación de la pronunciación asistida por computadora, y agrupación de visemas. Las investigaciones que se realizan en el área de agrupamiento de visajes son pocas, aunque este fenómeno puede producir aplicaciones audio – visuales apropiadas.

Los visemas se pueden analizar de dos maneras: En primer lugar, tomar datos de las formaciones de los labios de los seres humanos, los cuales cuentan con diferentes habilidades lectoras. A continuación, los resultados globales pueden tomarse como base para un análisis de los visemas. Sin embargo, por sí sola no puede demostrar una fiabilidad y exactitud. El segundo es el procesamiento algorítmico computarizado. Se considera que un modelo de contorno activo para la boca interior es exacto, que puede ser perseguido en varios enunciados donde se pueden visualizar puntos de interés en diferentes enunciados para detectar la variación del labio [24].

El visema es la forma visual de un fonema [9] como se puede apreciar en la Figura 2.2.3.2. En otras palabras, visemas de algunos fonemas son iguales, como “y” o “p” que son fonéticamente diferentes, pero lo mismo en forma visual. Sin embargo, cabe señalar que los visemas de un solo fonema no son necesariamente los mismos, ya que un fonema adquiere diversas formas gracias a las influencias ejercidas por sus fonemas anteriores y posteriores, denominados en conjunto coarticulación.



Figura. 2.2.3.2 Diferentes visema para pronunciar s.

#### 2.2.4. Inteligencia Artificial

La inteligencia Artificial (AI) está orientada a una perspectiva interdisciplinaria para comprender, modelar y replicar la inteligencia y procesos cognitivos humanos, a través de varios dispositivos computacionales, matemáticos, lógicos, mecánicos e incluso biológicos. A menudo tiene un enfoque abstracto y teórico, puesto que los investigadores dedicados a su estudio tratan de desarrollar teorías que faciliten la comprensión. Por otro lado, es puramente pragmático, debido a que otros investigadores se centran en la ingeniería de máquinas inteligentes y aplicaciones de la misma. Desde su creación a mediados del siglo XX, la Inteligencia Artificial ha sido una de las más nuevas y productivas áreas de investigación [25]. La Figura 2.2.4.1 muestra al robot ASIMO como aplicación de la inteligencia artificial.

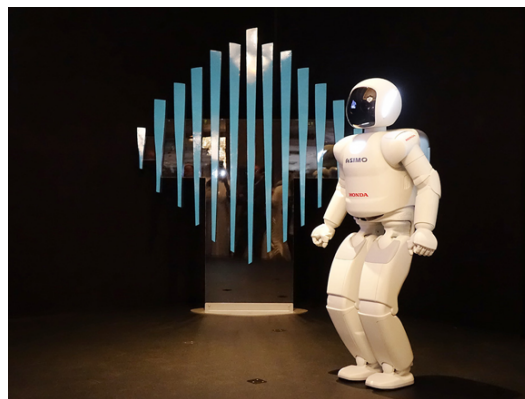


Figura. 2.2.4.1 Robot humanoide ASIMO una aplicación de la AI.

El hombre desde sus inicios se representó en su entorno mediante el uso de símbolos o caracteres que contribuyeron al desarrollo del lenguaje humano. Con esta premisa se podría decir que la inteligencia artificial (AI) es un rama

de las ciencias computacionales que tiene por objetivo la resolución de problemas mediante métodos no algorítmicos [26].

### **A. Enfoques de la Inteligencia Artificial (AI)**

Algunos de los enfoques de la Inteligencia Artificial (AI) son los siguientes:

- **Inteligencia Artificial (AI) fuerte:** Está enfocado a la programación de máquinas capaces de mantener estados cognitivos mentales para razonar y resolver problemas. Estas máquinas deben tener un nivel alto de comprensión similar a la inteligencia humana [27].
- **Inteligencia Artificial (AI) débil:** Algunas máquinas son incapaces de realizar y resolver problemas, pero tratan de actuar como agentes inteligentes. Para este caso la AI débil, las máquinas programadas de forma correcta pueden simular la conciencia humana.
- **Inteligencia Artificial (AI) aplicada:** Está enfocado al desarrollo de aplicaciones para el ámbito comercial. Ha logrado exitosas aplicaciones como: tarjetas inteligentes, reconocimiento facial y sistemas de seguridad [27].
- **Inteligencia Artificial (AI) cognitiva:** Tiene por objetivo el desarrollo de máquinas que estudien el comportamiento cognitivo del cerebro humano. Además del estudio de teorías sobre el reconocimiento los objetos (en el caso de la robótica), o la solución a los problemas abstractos.

### **B. Aplicaciones de la Inteligencia Artificial**

La inteligencia Artificial (AI) tiene una gran variedad de campos de aplicación entre ellos: la robótica, el análisis de imágenes o el tratamiento automático de textos. En el campo de la robótica, una de las temáticas que ha despertado interés en las investigaciones es el aprendizaje adaptivo, donde un sistema robotizado analiza las diferentes configuraciones con el objetivo de realizar movimientos complejos, por ejemplo: que un robot cuadrúpedo se desplace de manera autónoma, debe realizar un proceso de exploración y aprendizaje

similar al que realiza un recién nacido [28]. En la Figura 2.2.4.2 se presenta SpotMini un robot desarrollado por la compañía Boston Dynamics como ejemplo de aplicación de la Inteligencia Artificial.



Figura. 2.2.4.2 Robot SpotMini-Boston Dynamics.

Otro enfoque de la inteligencia artificial en la robótica son las interfaces cerebro-computador, sistemas artificiales que interactúan con el sistema nervioso de las personas mediante señales neurofisiológicas con la finalidad de asistir a personas que presentan alguna discapacidad motora.

Una de las ramas con mayor influencia en la inteligencia artificial se la denomina como sistemas expertos, que tienen por objetivo desarrollar sistemas capaces de analizar un conjunto de datos y realizar tareas asociadas a un perfil profesional como: el diagnóstico, detección de fallos, planificación y toma de decisiones [28].

En la Tabla 2.2.4.1 se muestran de forma resumida los campos de aplicación de la Inteligencia Artificial.

Tabla 2.2.4.1. Principales campos de aplicación de la inteligencia artificial.

Área	Aplicaciones
<b>Medicina</b>	Ayuda al diagnóstico Análisis de imágenes biomédicas Procesado de señales fisiológicas
<b>Economía</b>	Análisis financiero y bursátil Análisis de riesgos Estimación de precios en productos derivados Minería de datos Marketing y fidelización de clientes [27].
<b>Biología</b>	Análisis de estructuras biológicas Genética médica y molecular
<b>Informática</b>	Procesado de lenguaje natural Criptografía Teoría de juegos Lingüística computacional
<b>Robótica y Automática</b>	Sistemas adaptativos de rehabilitación Interfaces cerebro-computadora Sistemas de visión artificial Sistemas de navegación automática
<b>Física y Matemáticas</b>	Demostración automática de teoremas Análisis cualitativo sistemas no-lineales Caracterización de sistemas complejos

### C. Redes neuronales artificiales

Las redes neuronales artificiales se inspiran en el funcionamiento de la corteza cerebral en los mamíferos (biología). No obstante, hay que señalar que estos modelos no se comparan al funcionamiento, la escala y la complejidad del cerebro humano. Los modelos de redes neuronales artificiales se pueden deducir como un conjunto de unidades de procesamiento básicas, que están estrechamente interconectadas y operan en función a las entradas dadas, para procesar la información y generar las salidas deseadas [29]. En la Figura 2.2.4.3 se puede observar las principales características de las redes neuronales artificiales.



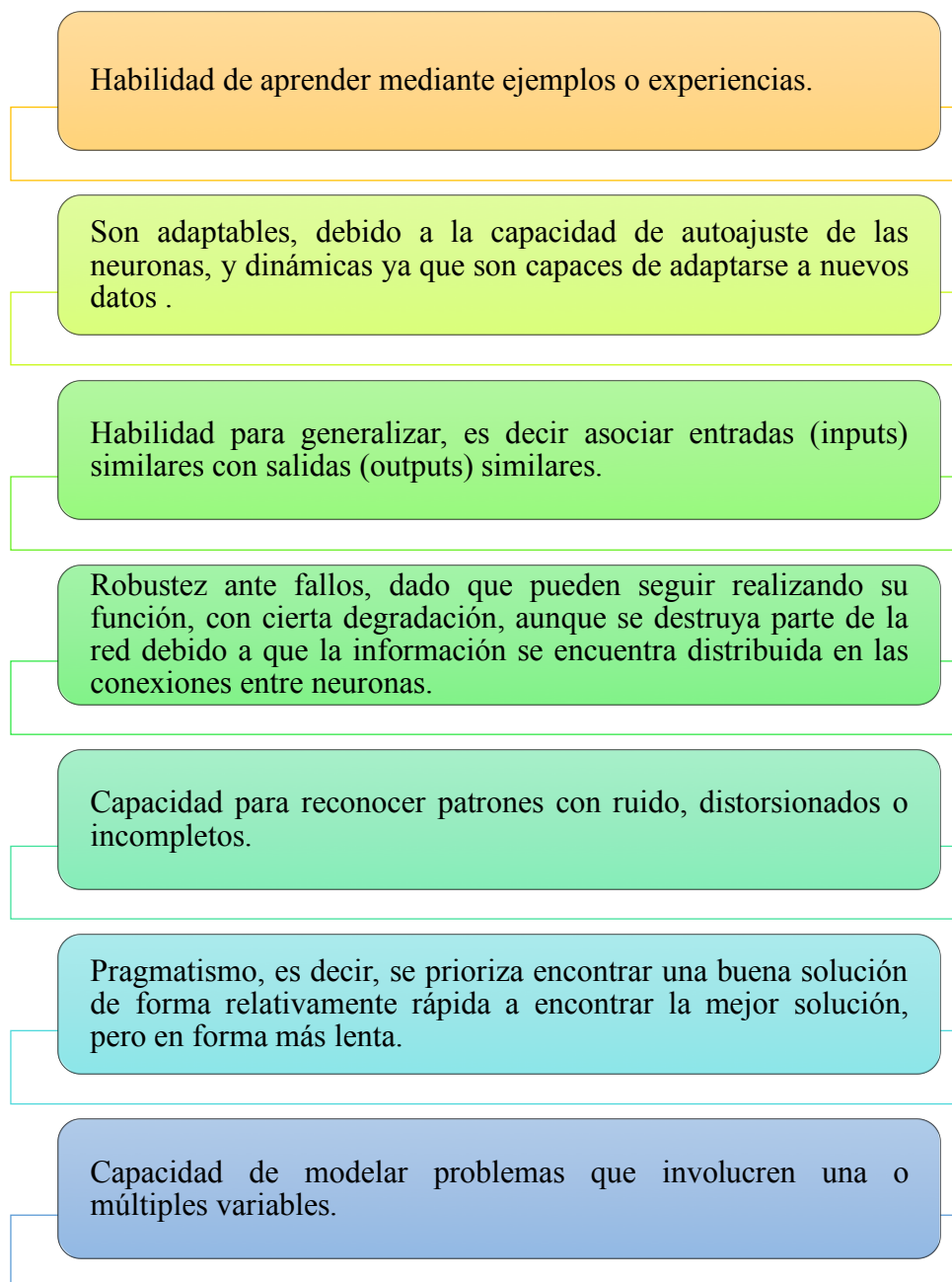


Figura. 2.2.4.3 Características de las redes neuronales artificiales [30].

Una red neuronal artificial está compuesta por un bloque o conjunto de neuronas conectadas entre sí imitando las conexiones sinápticas, cada neurona está compuesta por: conexiones de entrada, núcleo central de proceso y una salida como se puede observar en la Figura 2.2.4.4.

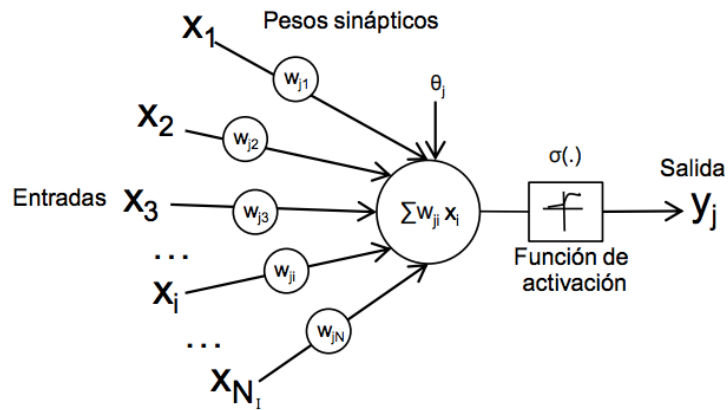


Figura. 2.2.4.4 Modelo de neurona artificial [29].

La red neuronal consta de una serie de datos de entrada  $x_i$  ( $i = [1, \dots, n]$ ), ponderadas por los  $w_{ji}$  que son los pesos de las conexiones a la neurona  $j$ ; combinados con el umbral  $\theta_j$ , dará como resultado el nivel de potencial de la neurona. El resultado de esta sumatoria es evaluado por la función de activación  $\sigma(x)$  que da lugar a la salida proceso  $y_j$ . La descripción matemática de la red se detalla a continuación [30].

$$y_j = \sigma\left(\sum_{i=1}^n w_{ij} * x_i \pm \theta_j\right) \quad (1)$$

Algunas de las funciones de activación más utilizadas en la actualidad para un sin número de aplicaciones de las redes neuronales artificiales desarrolladas son las siguientes:

✓ Escalón: 
$$\sigma(x) = \begin{cases} 0 & \text{si } x < 0 \\ 1 & \text{si } x \geq 0 \end{cases} \quad (2)$$

✓ Lineal: 
$$\sigma(x) = x \quad (3)$$

✓ Logística: 
$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (4)$$

✓ Tangente Hiperbólica: 
$$\sigma(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \quad (5)$$

Las redes neuronales se pueden catalogar en dos grupos genéricos según la forma en que se propaga la información en la red.

- **Redes de conexiones hacia delante (Feed-forward networks):** El flujo de información ocurre solo en una dirección. Si la red se considera como un gráfico con neuronas como sus nodos, las conexiones entre los nodos son tales que no hay bucles o ciclos en el gráfico.
- **Redes de conexiones hacia atrás (Feed-back networks):** Son redes de retroalimentación tienen conexiones que forman ciclos dirigidos. La arquitectura les permite operar y generar secuencias de tamaños arbitrarios. Las redes de retroalimentación exhiben capacidad de memorización y pueden almacenar información y relaciones de secuencia en su memoria interna [29].

#### D. Redes Neuronales Convolucionales

Las redes neuronales convolucionales son un ejemplo de redes de conexiones hacia delante (Feed-forward networks), su funcionamiento es muy similar a las redes neuronales estándar. Sin embargo, la diferencia radica en que cada unidad en una capa de CNN es un filtro de dos (o más) dimensiones que se convoluciona con la entrada de esa capa lo que lo hace esencial para casos en los que queremos aprender patrones de entradas de alta dimensión, por ejemplo, imágenes o videos. Los filtros CNN incorporan el contexto espacial al tener una forma espacial similar (pero más pequeña) que los medios de entrada, y usan el uso compartido de parámetros para reducir significativamente el número de variables aptas para el aprendizaje [30]. La Figura 2.2.4.5 muestra la estructura de una red neuronal Le-Net-5.

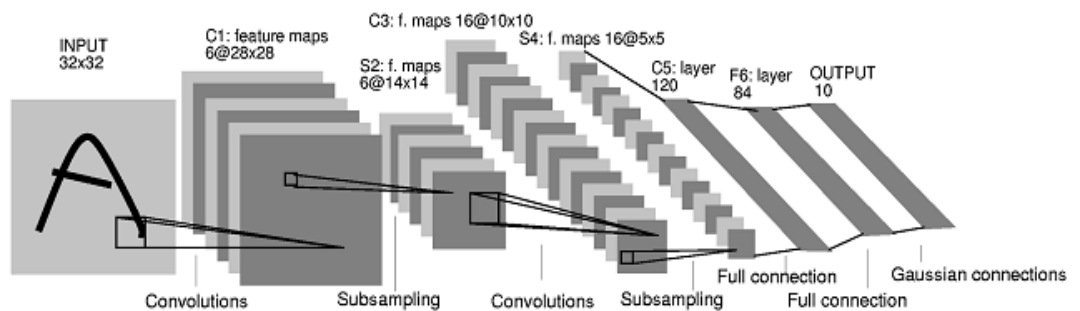


Figura. 2.2.4.5 Estructura de una red neuronal LeNet-5 [30].

## E. Casos de estudio

Existen varias arquitecturas en el campo de las Redes Convolucionales que hay sido de gran utilidad en el desarrollo de este tipo de modelos y gracias a varias contribuciones de muchos se van puliendo los modelos CNNs, en la siguiente tabla se visualiza varias aportaciones de estos modelos.

Tabla 2.2.4.2 Modelos de redes neuronales convolucionales desarrollados [31].

<b>LeNet</b>	Las primeras aplicaciones exitosas de Convolutional Networks fueron desarrolladas por Yann LeCun en la década de 1990. De estos, el más conocido es la arquitectura LeNet que se utilizó para leer códigos postales, dígitos, etc.
<b>AlexNet</b>	AlexNet. La red tenía una arquitectura muy similar a LeNet, pero eran capas convolucionales más profundas, más grandes y más destacadas apiladas una encima de la otra [31].
<b>ZF Net</b>	Fue una mejora en AlexNet al ajustar los hiperparámetros de la arquitectura, en particular, al expandir el tamaño de las capas convolucionales del medio y hacer que el tamaño del paso y del filtro en la primera capa sea más pequeño.
<b>GoogLeNet</b>	La principal contribución fue el desarrollo de un módulo de inicio que redujo drásticamente la cantidad de parámetros en la red (4M, en comparación con AlexNet con 60M).
<b>VGGNet.</b>	Su principal contribución fue mostrar que la profundidad de la red es un componente crítico para un buen desempeño. Su mejor red final contiene 16 capas CONV / FC y, sorprendentemente, presenta una arquitectura extremadamente homogénea que solo realiza convoluciones de 3x3 y agrupación 2x2 desde el principio hasta el final. Una desventaja de VGGNet es que es más costoso de evaluar y usa mucha más memoria y parámetros (140M) [31].
<b>ResNet.</b>	Cuenta con conexiones de omisión especiales y un uso intensivo de la normalización de lotes. A la arquitectura también le faltan capas totalmente conectadas al final de la red.

## F. Estructura de una red neuronal convolucional

La estructura básica de una red convolucional está formada por las siguientes capas:

- **Capa convolucional:** Es el núcleo de las redes convolucionales y consiste en una serie de filtros de aprendizaje, estos filtros tienen un campo receptivo pequeño tamaño. Siguiendo el flujo de aprendizaje hacia delante cada uno de estos filtros se convoluciona con todo el campo de visión produciendo así un mapa de características.
- **Capa de submuestreo:** Es una parte importante de las redes neuronales, estas capas se aplican después de las capas convolucionales. El objetivo principal es reducir el tamaño de la entrada, por otro lado, esto genera pérdidas de información, pero también beneficia a la red al reducir la carga de cálculo en las siguientes capas y además de reducir el sobreajuste de la red.
- **Capa completamente conectada:** Por lo general su ubicación está al final de las combinaciones de las capas convolucionales y de submuestreo. Cada elemento corresponde a una neurona y el número total de neuronas en estas capas corresponderá al número de clases que se desean predecir. La última se emplea en tareas de clasificación [30].

## G. Diseño mecánico

El diseño mediante la aplicación de software de que permiten el diseño en 3d tales como: AutoCAD, Inventor, Maya. SolidWorks, Catia, entre otros, facilitan la tarea de diseño, a más de ser capaces de visualizar el acabado final del modelo con respectivas herramientas de renderización e incluso permiten el simulado de mecanismos si se da el caso para el modelo.

Un potente software para el diseño y modelado en 3D que emplean muchas empresas hoy en día es el software SolidWorks desarrollado por la compañía Dessault Systemes quienes herramientas de software 3D completas para crear, simular, publicar y administrar los datos. Los productos de SolidWorks son fáciles de aprender y utilizar; obtendrá mejores diseños, más rentables y de

forma más rápida. La facilidad de uso de los productos de SolidWorks permite a más ingenieros, diseñadores y profesionales de la tecnología centrarse más que nunca en las ventajas del 3D al darles vida a los diseños [32].

### **2.3. Propuesta de Solución**

Mediante la construcción de un robot social interactivo con aspecto humanoide, se pretende estimular la ejecución de proyectos de investigación innovadores y generativos de nuevo conocimiento para los estudiante, docentes e investigadores de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial. A demás, con la presentación de un modelo robótico controlado por redes neuronales se pretende impulsar el desarrollo de aplicaciones basadas en inteligencia artificial para el control de robots.

## **CAPÍTULO III**

### **METODOLOGÍA**

#### **3.1. Modalidad de Investigación**

La modalidad de investigación del presente trabajo se enmarcó en la investigación aplicada, debido a que se emplean los conocimientos científicos y técnicos para brindar mecanismos que faciliten la resolución de un problema y contribuyan al desarrollo de ideas para conseguir mejoras e innovaciones en el campo de la robótica social.

Las modalidades de investigación son las siguientes:

##### **3.1.1. Investigación Bibliográfica-Documental**

Es de carácter bibliográfica-documental en vista de que se ha requerido ampliar y profundizar conceptos, temas y criterios de los diversos autores, en base a información de libros, revistas, tesis artículos para el desarrollo del marco teórico.

##### **3.1.2. Investigación experimental**

Es de carácter experimental debido a que se llevará a cabo una serie de pruebas con diferentes equipos y dispositivos, hasta lograr los resultados deseados.

#### **3.2. Población y Muestra**

El presente proyecto no se contará con población y muestra, puesto que se realizará una investigación aplicada, la información requerida fue tomada de fuentes bibliográficas.

### **3.3. Recolección de información**

Para el presente proyecto la recolección de información se realizará mediante la revisión de información pertinente del proyecto en recursos como: fuentes online, libros y artículos que aporten al desarrollo de la investigación; además el uso de guías que clarifican la idea del diseño y control del robot social interactivo.

### **3.4. Procesamiento y análisis de datos**

La información recolectada será inducida a un análisis crítico, para poder condensar información defectuosa o incompleta. Para ello se llevará a cabo los siguientes pasos:

- ✓ Recolección de información mediante Internet, investigaciones, librerías, artículos científicos, tesis, libros.
- ✓ Análisis y clasificación de la información recolectada para obtener conceptos de importancia referentes al tema.
- ✓ Interpretación de resultados con el apoyo de fundamentos teóricos para cada aspecto inmerso en el desarrollo del proyecto.

### **3.5. Desarrollo del Proyecto**

Los aspectos básicos al momento del desarrollo del proyecto se detallan a continuación:

- ✓ Analizar los antecedentes sobre los robots sociales interactivos, y clasificar la información más relevante acerca del diseño, construcción y aplicabilidad de los mismos.
- ✓ Determinar el material adecuado del cual estarán constituido, tanto las piezas internas del prototipo, como también la parte externa del mismo.
- ✓ Diseñar y construir las piezas que servirán de soporte para todos los componentes que se alojaran en el interior del prototipo.
- ✓ Construir y recubrir la parte externa del robot.
- ✓ Establecer una configuración inicial de la tarjeta controladora, sensores, actuadores y dispositivos afines al sistema de movilización y mando.



- ✓ Acondicionar los componentes electrónicos y mecánicos que formarán parte del mecanismo de movilización.
- ✓ Desarrollo del controlador de gesticulación basado en redes neuronales convolucionales
- ✓ Desarrollo del controlador de habla mediante la implementación de visemas.
- ✓ Evaluación del rendimiento tanto del prototipo y la comunicación entre sí.

## **CAPÍTULO IV**

### **DESARROLLO DE LA PROPUESTA**

El control de gesticulación y habla de un robot interactivo social con apariencia humana es una tarea compleja que exige eventualmente una serie de pasos necesarios para su correcto desarrollo, para lo cual en la propuesta se explica de manera minuciosa el diseño y construcción del prototipo con un alto nivel de parentesco hacia los seres humanos, capaz de realizar correctamente las tareas asignadas, de igual manera se menciona el estudio y desarrollo de los controladores para cada canal del robot (visual, auditivo).

El canal visual del robot estará basado en algoritmos empleados actualmente en el llamado aprendizaje profundo (Deep Learning) utilizados en el desarrollo de aplicaciones de inteligencia artificial, los cuales permitirán la incursión a nuevas metodologías de control de robots tanto en el campo investigativo como en el industrial, debido a que los algoritmos desarrollados bajo este concepto pueden adaptarse a cualquier necesidad.

En una sección 4.2.1 del presente capítulo se describe el hardware y software empleados, considerando y aprovechando las características y propiedades que presentan cada una de estas herramientas, estos a su vez tienen una gran importancia en la interacción entre el usuario y el robot. También se hace hincapié en el uso de nuevas tecnologías como son: la tecnología de impresión 3D para la construcción de robot, por las diversas ventajas que este presenta, así como también el uso de librerías open-source para el procesamiento de imágenes.

## **4.1. Análisis de factibilidad**

En los siguientes apartados del proyecto de investigación se presentan las instancias de factibilidad para el desarrollo de la propuesta.

### **4.1.1. Factibilidad técnica**

El trabajo de investigación desde el punto de vista técnico es factible desarrollado, debido a que el investigador tiene a su disposición materiales, equipos electrónicos e informáticos.

### **4.1.2. Factibilidad económica**

La propuesta de un robot interactivo social con aspecto humano, tiene una apertura positiva en la factibilidad económica, debido a que los gastos inmersos en este serán amortizados por el investigador.

## **4.2. Análisis de requerimientos**

Para el desarrollo del presente proyecto de investigación se tiene en cuenta los siguientes requerimientos.

### **4.2.1. Hardware, sensores y actuadores**

#### **A. Tarjeta controladora Lynxmotion SSC-32**

Dispositivo capaz de controlar hasta 32 servomotores mediante el puerto serial a través de la modulación por ancho de pulso (PWM), ofreciendo una respuesta inmediata, mediana, o sincronizada, con una resolución de 1useg para posiciones exactas en un rango de 0.5mseg hasta 2.5mseg que cubren los 180° [33]. En la Tabla 4.2.1.1 se describen las características eléctricas de la tarjeta controladora de servomotores, mientras que la Figura 4.2.1.1 muestra la tarjeta controladora.

Tabla 4.2.1.1 Características eléctricas de la placa controladora SCC-32

<b>Parámetro</b>	<b>Descripción</b>
Microcontrolador	Atmel ATMEGA168-20PU
Velocidad	14.75 MHz
Entrada serial	RS-232 o TTL, 2400, 9600, 38.4k, 115.2k, N81
Salidas	32 (Servo o TTL)
Entradas	4 (Análoga o Digital)
Requerimientos de corriente	31mA
Conectividad	DB9F
Tipos de Servo que soporta	Futaba o Hitec
Rango de movimiento del servo	180°
Resolución del servo	1uS, .09°
Velocidad de resolución del servo	1uS / Segundo
Control de movimiento del servo	Inmediato, Medido, o sincronizado
Corriente máxima VS	máximo 15 amperios por lado

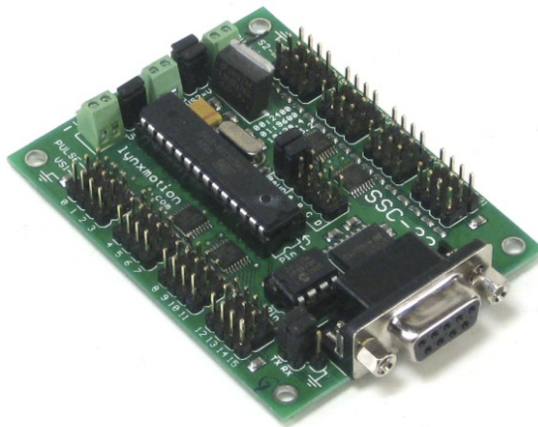


Figura. 4.2.1.1 Tarjeta controladora de servomotores SSC-32 [33].

## B. Servomotores

Son actuadores mecánicos que ofrecen un gran torque para manipulación de objetos de distinto peso, con rango de movimiento establecido de 0 a 180 ideales para la ejecución de movimientos con gran precisión. La empresa HITEC ofrece una gran variedad de estos actuadores de acuerdo a los requerimientos de los usuarios. La Tabla 4.2.1.2 se observan las características de los servomotores empleados en el prototipo de robot social.

Tabla 4.2.1.2 Características de los actuadores tipo servomotor.

Parámetros	Servomotores		
	Hitec HS-625MG	Hitec HS-815BB	Tower Pro SG-90
Rango de Voltaje	4.8V – 6.0V	4.8V – 6.0V	4.8 V
Velocidad sin carga (4.8V)	0.24seg / 60°	0.48sec / 140°	0.1 s/60°
Velocidad sin carga (6.0V)	0.20seg / 60°	0.38seg / 140°	-
Par de torsión (4.8V)	106.93 oz / in (7.7kg.cm)	274.96 oz / in (19.8kg.cm)	1.8kg / cm
Par de torsión (6.0V)	133.31 oz / in (9.6kg.cm)	343.01 oz / in (24.7kg.cm)	-
Rango de señal de PWM máximo (estándar)	553-2520 $\mu$ sec	920-2105 $\mu$ sec	-
Amplitud de pulso	3-5V	3-5V	-
Temperatura de funcionamiento	-20°C a + 60°C	-20°C a + 60°C	0°C - 55°C
Retroalimentación	Potenciómetro de 5K $\Omega$	Potenciometr o de 5K $\Omega$	
Material del engranaje	3 engranajes metálicos, 1 metálico de resina	Nylon	Nylon

### C. Cámara web

Es necesario el uso de sensores para lo cual la cámara incorporada en prototipo actuará como un sensor, con la finalidad de capturar imágenes en tiempo real para su posterior análisis y procesamiento con herramientas informáticas.

Existen una variedad de cámaras web en el mercado desde baja resolución hasta una resolución full HD que proporcionan las cámaras Logitech, ideales en el uso de aplicaciones que requieren imágenes con gran cantidad de información, como por ejemplo en la detección de tumores cerebrales. Para el proyecto se empleó una cámara web USB con una resolución HD.

Tabla 4.2.1.3 Características del sensor de imagen.

<b>Especificaciones WEBCAM GENIUS</b>	
Sensor de imagen	CMOS HD 720p píxeles
Tipo de lente	Lente de enfoque manual
Conectividad	USB 2.0
Formato de archivo	MJPEG/WMV
Imágenes	1MP, 1280x720, 640x480 píxeles
Video	CIF/VGA, 720p HD con hasta 30 cuadros / segundo

#### 4.2.2. Software

Existe una infinidad de lenguajes de programación adaptables al desarrollo de los algoritmos para el control de todas las acciones del robot interactivo social, desde aquellos lenguajes que su uso no exige un completo dominio teórico-práctico sobre programación hasta los lenguajes que presentan complejidad en la ejecución de módulos, métodos y librerías necesarios en el desarrollo de pseudocódigos. El lenguaje considerado como idóneo para la codificación de los controladores es Python, el cual que ofrece muchas ventajas y facilidades frente a otras alternativas como por ejemplo C++.

## A. Python

Python es un lenguaje de programación de alto nivel, su selección viene dada por una serie de ventajas sobre el lenguaje de programación C o C++, se describen a continuación:

- Es un lenguaje multiplataforma es decir que puede correr bajo, Windows, OS X o Linux.
- Tiene un entorno de desarrollo relativamente intuitivo y de fácil aprendizaje, además de ofrecer la posibilidad de integrar una gran cantidad de aplicaciones.
- Está enfocado al procesamiento de datos, inteligencia artificial, aprendizaje de máquina (Machine Learning) entre otros.
- Existe una gran cantidad de información sobre su uso y aplicación, así como también códigos desarrollados por una enorme comunidad que respalda la aplicación de este lenguaje.
- Facilidad en la instalación de módulos, librerías, etc.

## B. Anaconda

Anaconda es un gestor que facilita la instalación de varios módulos, librerías, paquetes, etc. Cuenta con varias alternativas en entornos de desarrollo, haciendo énfasis a los principales y con mayor reconocimiento, estos son:

- **Spyder:** Es un potente entorno de desarrollo con funciones avanzadas de edición, pruebas interactivas, depuración e introspección y un entorno informático numérico gracias al soporte de IPython (intérprete interactivo mejorado de Python) y bibliotecas populares de Python como NumPy (álgebra lineal), SciPy (procesamiento de señal e imagen) o Matplotlib (trazado interactivo 2D/3D) [34].
- **Jupyter Notebook:** Es una aplicación web de código abierto que permite crear y compartir documentos que contengan códigos, ecuaciones, visualizaciones y textos narrativos en vivo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático, entre otros [35].

### C. Librerías y Paquetes

Las librerías necesarias compatibles con Python y empleadas en la codificación de los controladores del robot social interactivo son:

- **Keras (2.0.8):** Es una biblioteca de red neuronal de código abierto escrita en Python. Es capaz de ejecutarse sobre los backends de TensorFlow, Microsoft Cognitive Toolkit, Theano o MXNet. Su enfoque global es permitir una rápida experimentación con redes neuronales profundas, debido a que es fácil de usar, modular y extensible [36].
- **TensorFlow (1.3.0):** Es una biblioteca de código abierto para el cálculo numérico de alto rendimiento. Su arquitectura flexible permite una fácil implementación de computación en una variedad de plataformas (CPU, GPU, TPU) y desde escritorios hasta clústeres de servidores y dispositivos móviles y periféricos.
- **PySerial (3.4):** Paquete que permite la manipulación de los puertos seriales del hardware.
- **SpeechRecognition (3.8.1):** Interfaz de programación de aplicaciones (APIs) de Google que permite convertir el audio en texto aplicando potentes modelos de redes neuronales.
- **Lxml (4.1.0):** Es una biblioteca que presenta una gran variedad de funciones, fácil de usar para procesar archivos bajo lenguaje XML y HTML en el lenguaje Python.
- **NumPy (1.13.3):** Es un paquete que permite el procesamiento de matrices y otras operaciones matemáticas, está diseñado para trabajar con matrices de N-dimensiones, funciones sofisticadas (difusión), herramientas para integrar códigos en C/C++ y álgebra lineal útil, transformada de Fourier y capacidades de números aleatorios, otra característica que lo destaca es la manipulación de base de datos [37].
- **Matplotlib (2.1.0):** Proporciona herramientas para generar gráficos, histogramas, espectros de potencia, gráficos de barras, diagramas de errores, diagramas de dispersión, etc.



- **Pandas (0.20.3):** Biblioteca de código abierto con licencia de BSD que proporciona estructuras de datos y herramientas de análisis de datos de alto rendimiento y fácil de usar para el lenguaje de programación Python [37].
- **Brewer2mpl (1.4.1):** Es un paquete que ofrece un rápido acceso a los mapas de colores para gráficos, y no necesita conectarse a internet para el acceso.
- **Scikit-learn (0.19.1):** Biblioteca de software libre dedicado al aprendizaje de máquina, el cual trabaja con Python. Cuenta con varios algoritmos de clasificación, regresión y agrupación, incluyendo máquinas de vectores de soporte (SVM), bosques aleatorios, aumento de gradiente, k-means y DBSCAN, y está diseñado para interoperar con las bibliotecas numéricas y científicas NumPy y SciPy [38].

#### **D. OpenCV**

Es una librería de código abierto, está escrita en C y C++ y se puede ejecutar en los sistemas operativos Linux, Windows y Mac OSX, además existen paquetes que se pueden ejecutar en Python, Java, MATLAB y otros lenguajes, incluyendo el desarrollo de aplicaciones móviles bajo Android e iOS. OpenCV está diseñada para mejorar la eficiencia computacional en el procesamiento de imágenes, videos, etc., aprovechando los procesadores multinúcleo.

Proporciona una infraestructura de visión por computador fácil de usar y que ayude a los usuarios a desarrollar aplicaciones de visión mucho más sofisticadas en menor tiempo. OpenCV cuenta con más de 500 funciones que cubren muchas áreas de visión, incluida la inspección de productos de fábrica, imágenes médicas, seguridad, interfaz de usuario, calibración de cámaras, visión estéreo y robótica. También contiene una biblioteca completa de aprendizaje de máquina de uso general (módulo ML). Esta sub-biblioteca se centra en el reconocimiento estadístico de patrones y la agrupación.

Los paquetes **opencv-python** y **opencv-contrib-python** son empleados en el desarrollo del controlador de gesticulación del robot social interactivo, los cuales ofrecen en su mayoría las funciones para el desarrollo de aplicaciones de visión artificial en Python.

### 4.3. Diseño y construcción del robot interactivo social

#### 4.3.1. Diseño de piezas

El diseño y modelado de un rostro humano no es una tarea trivial, debido que requiere de un profundo conocimiento de varias técnicas de la escultura, además de contar con una gran habilidad en el uso de herramientas informáticas para el modelado de piezas irregulares o complejas, razón por la cual se optó utilizar el modelo estructura del robot humanoide INMOOV desarrollado por el escultor francés Gaël Langevin en septiembre del 2011, quien generosamente coloca toda la información, documentos, diagramas y archivos al alcance de todo el mundo en su página web: <http://inmoov.fr/> [39].

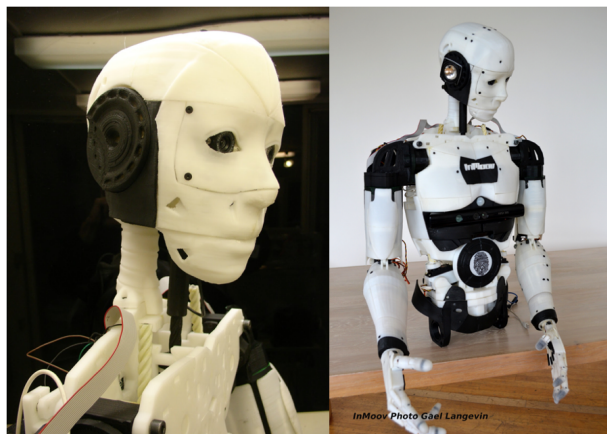


Figura. 4.3.1.1 Robot Humanoide INMOOV [39].

Como se puede observar en la Figura 4.3.1.1, INMOOV es un robot humanoide impreso con la tecnología 3D lo que hace factible su uso, reduciendo enormemente el trabajo de diseño desde cero.

Los archivos empleados en la construcción del robot en su mayoría del humanoide INMOOV son: las piezas que forma la parte exterior de la cabeza mostrados en la Figura 4.3.1.2, así como también las piezas internas que forman el mecanismo para movimiento de cuello (Figura 4.3.1.3), así como también parte del torso para la unión de la cabeza al cuello proporcionando el movimiento vertical.

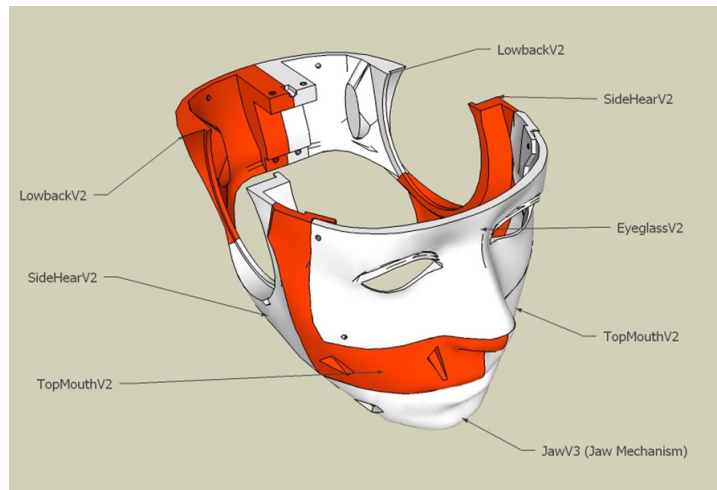


Figura. 4.3.1.2 Estructura de la cabeza de INMOOV [39].

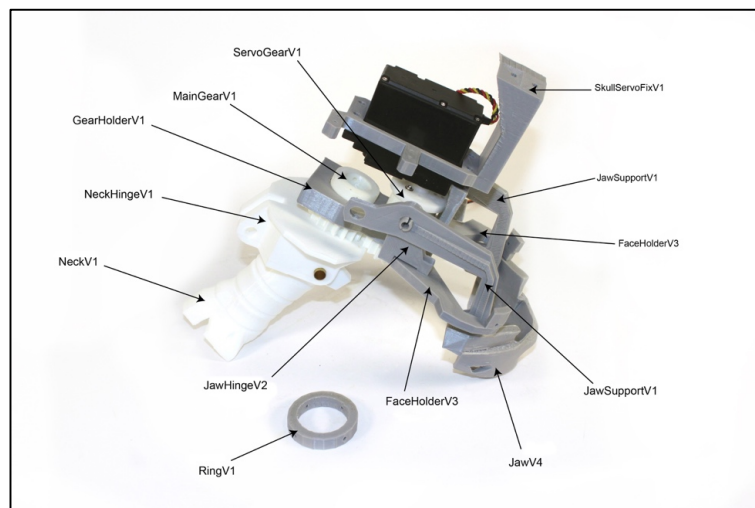


Figura. 4.3.1.3 Estructura del cuello de INMOOV [39].

La desventaja que presenta INMOOV es que no puede expresar gestos faciales por lo que se recurrió al diseño y construcción de otros mecanismos que cumplan este condicionante.

Con la ayuda del software de diseño y modelado 3D se realizó bosquejos de los mecanismos necesarios para el movimiento de ojos, cejas y labios.

### **A. Impresión 3D**

Una de las ventajas que tiene la tecnología 3D y por la cuál fue la mejor opción para la fabricación de todas las piezas del robot social, es su bajo costo y además de ofrecer una amplia gama de posibilidades en la impresión de piezas u objetos de geometría compleja.

Los archivos previamente diseñados en el software 3D se exportan a archivos con la extensión STL, los cuales son utilizados por el programa de impresión 3D Cura 15.0.1 de la empresa Ultimaker para generar el código G que utiliza la impresora para construir la pieza.

### **B. Mecanismo movimiento de ojos y parpados.**

El mecanismo que cumplirá con el objetivo de reproducir el movimiento de ojos y cejas está basado en la anatomía propia del ojo humano como se apreciar en la Figura 4.3.1.4, el literal (a) muestra los músculos presentes en el ojo y en el literal (b) el mecanismo del ojo.

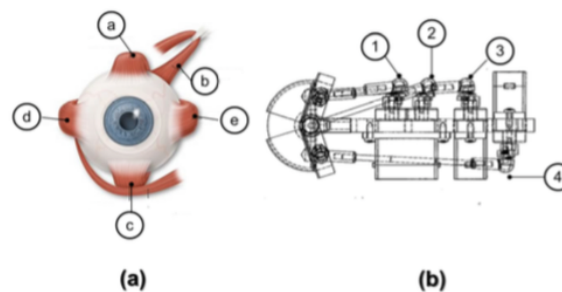


Figura. 4.3.1.4 Mecanismo de ojo [21].

El elemento propio que dispone INMOOV para el desplazamiento de ojos solo permite el movimiento de izquierda – derecha, además carece del movimiento de parpados, haciéndolo inadecuado para su uso, por esta razón se diseñó un nuevo mecanismo que no solo permita el desplazamiento descrito, sino que

también permita el desplazamiento de arriba-abajo, además de implementar el movimiento de párpados. En la Figura 4.3.1.5 se observa el mecanismo implementado. En el Anexo H se muestra plano del mecanismo mencionado.



Figura. 4.3.1.5. Mecanismo para el movimiento de ojos y párpados.

### **C. Mecanismo movimiento de cejas**

El mecanismo está constituido por 2 servomotores colocados en la estructura frontal del robot que simulen el movimiento de cejas. En la Figura 4.3.1.6 se aprecia el mecanismo.

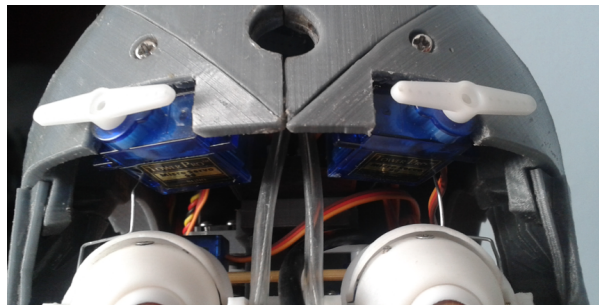


Figura. 4.3.1.6 Mecanismo para el movimiento de cejas.

### **D. Mecanismo para movimiento de boca y labios**

Para diseñar el mecanismo de movimiento de boca y labios se analizó la estructura labial del robot social interactivo DORIS [23], como se puede apreciar en la Figura 4.3.1.7, el cual consta de 6 grados de libertad, cada GDL controlado por su respectivo actuador (Servomotor).

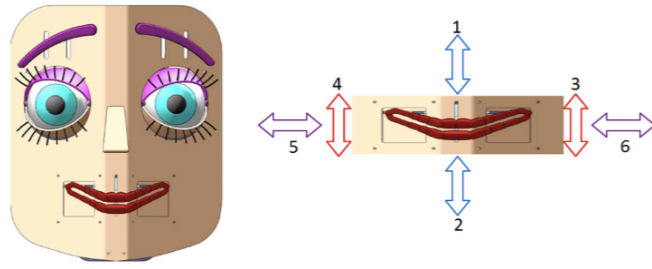


Figura. 4.3.1.7 Estructura labial del robot social DORIS [23].

Los labios de DORIS están inspirados en la anatomía de la biología humana, en donde cada grado de libertad corresponde a un músculo o conjunto de músculos. En la Tabla 4.3.1.1 se visualiza la representación de cada músculo vinculado a los 6 grados de libertad que dispone el robot.

Tabla 4.3.1.1 Configuración de los músculos-grados de libertad

N°	Músculo	Grado de libertad
1	Elevador de labio superior	1
2	Obicular	1, 2, 3, 4, 5, 6
3	Depresor de labio superior	4
4	Mentoniano	4
5	Risorio	2, 6
6	Cigomático mayor	6
7	Cigomático menor	1

Tomando en consideración las configuraciones presentado en la Tabla 4.3.1.1, se modificó las piezas exteriores del robot, el cual solo contaba con el desplazamiento vertical de la mandíbula inferior, para implementar los siguientes elementos: dentadura plástica que le otorga un aspecto más realista al robot, canales de plástico con filamentos metálicos que actúan como los

músculos de los labios y estos a su vez conectados respectivamente cada uno a un servomotor dando un total de 6GDL. La Figura 4.3.1.8 muestra las partes que conforman el mecanismo de movimiento de boca y labios.

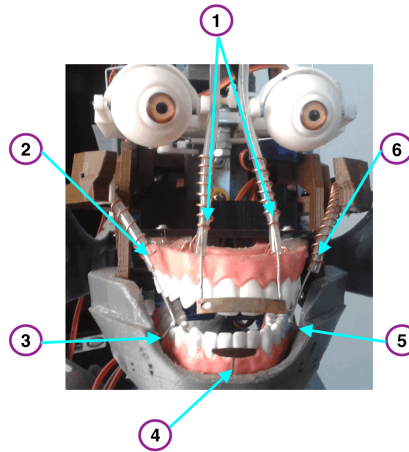


Figura. 4.3.1.8. Mecanismo para el movimiento de labios del robot social.

#### **E. Ensamble final de robot interactivo social**

Una vez presentado y elaborado los diferentes mecanismos a incluir en el robot interactivo social, y mediante el uso de la guía didáctica del robot INMOOV disponible en su página web, se ensamblaron todas las partes del robot a través del uso del uso de elementos mecánicos como juntas, tonillos, tuercas, etc. En la Figura 4.3.1.9 se presenta el ensamble final del robot.

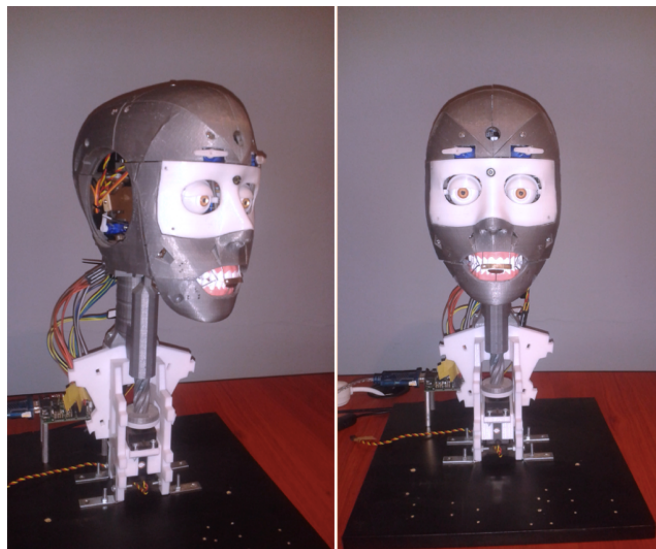


Figura. 4.3.1.9 Ensamble final del robot interactivo social.

### 4.3.2. Sistema eléctrico y electrónico

El sistema eléctrico y electrónico del robot social interactivo mostrado en la Figura 4.3.2.1 está constituido por los siguientes puntos:

- Los 15 actuadores para cada GDL, se conectan a la placa controladora de servomotores.
- La controladora de servomotores se energiza con una fuente de poder de 5V a 2,5A.
- La interacción entre la placa de servomotores y el ordenador se realiza mediante comunicación serial, empleando un cable convertor de USB–Serial.
- La cámara web se conecta al ordenador mediante USB.
- El ordenador controlará al robot social mediante el uso de Scripts desarrollados en Python y ejecutados bajo consola.

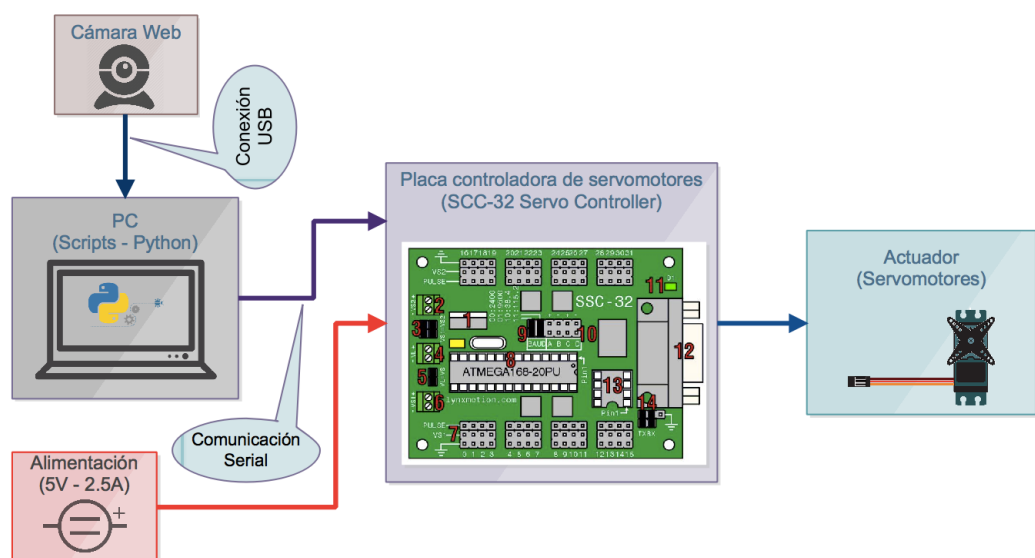


Figura. 4.3.2.1 Sistema eléctrico y electrónico del robot.



#### 4.4. Gesticulación del robot interactivo social

La gesticulación del robot interactivo social está basada en la forma habitual en que los seres humanos expresan sus emociones, estas expresiones se realizan mediante el uso del lenguaje corporal, para el presente proyecto se asume como lenguaje corporal los gestos faciales, estos son: el gesto de felicidad, tristeza, enojo, miedo, etc. Partiendo de esta premisa el controlador a desarrollar estará constituido de las siguientes partes descriptas en la Figura 4.4.1.

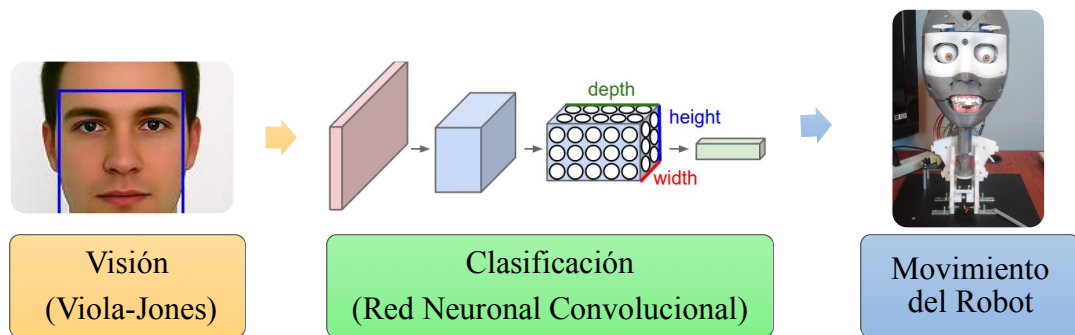


Figura. 4.4.1 Sistema de gesticulación del robot social.

El control de gesticulación para el robot interactivo social está conformado por los módulos de, visión, clasificación y movimiento, cada uno de ellos realizan tareas específicas tales como:

- **Visión:** Tiene por objetivo el procesamiento de la imagen de entrada al sistema de gesticulación, y la vez es el encargado de proveer al módulo de clasificación solo la información relevante de cada imagen.
- **Clasificación:** Módulo encargado de seleccionar la información correcta de acuerdo a la entrada recibida mediante un algoritmo de predicción.
- **Movimiento del robot:** Toma el dato enviado por el módulo de clasificación y lo compara mediante condicionales para en su efecto elegir el bloque de movimientos a ejecutar en el robot.

En el siguiente apartado se describe a detalle el proceso de construcción y funcionamiento de cada módulo que conforma el controlador de gesticulación del robot.

#### 4.4.1. Visión

El módulo de visión está enfocado en eliminar toda la información de la imagen de entrada que no es relevante para el sistema de control de gesticulación, para ello realiza varias etapas en tratamiento de imagen, estos son: adquisición continua de imagen, pre-procesamiento, reconocimiento facial y envío de rostro detectado, como se puede observar en la Figura 4.4.1.1.

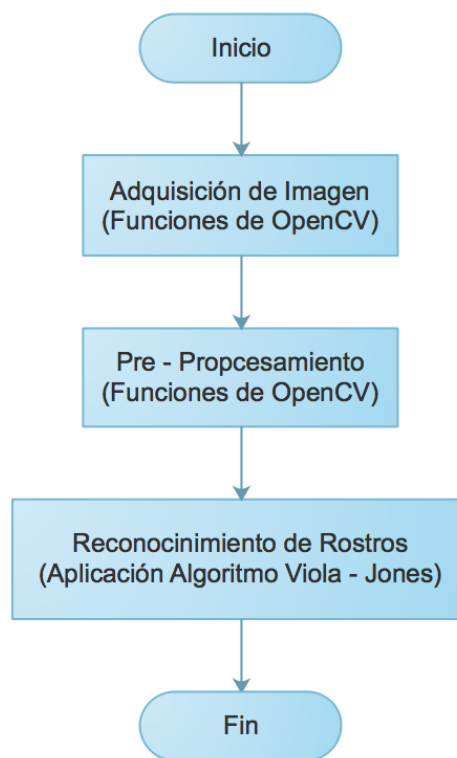


Figura. 4.4.1.1 Diagrama de flujo del módulo de Visión.

**A. Adquisición de Imagen:** Es la captura continua de imagen a través de la cámara web incorporada en el robot social, en esta subsección se utilizaron algunas funciones de OpenCV y codificadas en un Script de Python. En la Tabla 4.4.1.1 se describen las funciones empleadas

Tabla 4.4.1.1 Funciones de OpenCV - módulo de adquisición de imagen.

<b>Función</b>	<b>Descripción</b>
Cap=cv2.VideoCapture(1)	La variable “Cap” Captura video continuo de la cámara web del robot social, el número 1 es la cámara web conectada.
Cap.read()	Comprueba que la variable “Cap” lea correctamente los frames o cuadros.
cv2.imshow()	Despliega una pantalla con el frame o cuadro de imagen adquirido por Cap.
Cap.release()	La variable Cap finaliza la captura de frames.
cv2.destroyAllWindows()	Cierra la ventana desplegada por la variable Cap.

**B. Pre-procesamiento:** La cámara web captura imágenes con una orientación de 180° (cámara girada por el diseño del robot), y la resolución de entrada es de 720 pixeles, la etapa de pre-procesamiento se encarga de girar la imagen la orientación de entrada, reducir o re-escalar los pixeles a procesar y convertir la imagen en tonos de grises para enviarlo a la siguiente etapa. Para ello utiliza varias funciones de OpenCV escritas en Python, las cuales se observan en la Tabla 4.4.1.2.

Tabla 4.4.1.2 Funciones del módulo de pre-procesamiento.

<b>Función</b>	<b>Descripción</b>
cv2.flip(imagen, -1)	Función que gira la imagen -180 (-1)
cv2.resize(imagen (w, h), interpolation = cv2.INTER_LINEAR)	Reduce el tamaño de la imagen de 720x640 a 480x320 pixeles, la interpolación es bilineal
cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)	Se convierte la imagen de color a una imagen en tono de grises.

**C. Reconocimiento de rostros:** El módulo de Clasificación requiere como entrada la imagen del rostro de la persona que interactúa, por lo que la demás información presente en la imagen debe ser desechada, para ello entra en funcionamiento la etapa de detección o reconocimiento de rostros, el cual utiliza algoritmo de Viola-Jones, uno de los primeros métodos para detectar objetos en tiempo real en una imagen.

El algoritmo de Viola-Jones es usado ampliamente en la detección de caras en imágenes y video, ya que se basa en la comparación entre las intensidades luminosas de regiones rectangulares de las imágenes denominadas características de tipo Haar (Haar-like features), las cuales son calculadas aplicando una imagen integral. La Figura 4.4.1.2 se puede observar la estructura del algoritmo de Viola-Jones [40].

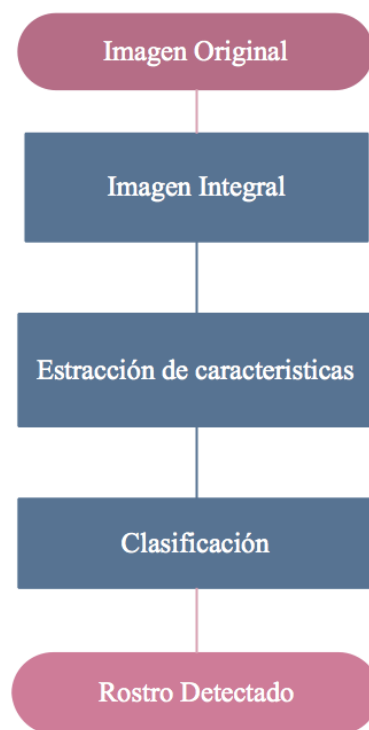


Figura. 4.4.1.2 Estructura del algoritmo de Viola-Jones [41].

- **Las características de tipo Haar:** son rasgos o caracteres muy simples que se buscan en las imágenes y que consisten en la diferencia de intensidades luminosas entre regiones rectangulares

adyacentes, por tanto, quedan definidas por unos rectángulos y su posición relativa a la ventana de búsqueda, adquiriendo un valor numérico resultado de la comparación que evalúan. Existen 3 tipos de característica (features) mostrados en la Figura 4.4.1.3 que presenta Viola-Jones:

- Características de dos rectángulos cuyo valor es la diferencia entre las sumas de los píxeles contenidos en ambos rectángulos. Las regiones tienen la misma área y forma y son adyacentes.
- Características de tres rectángulos que calculan la diferencia entre los rectángulos exteriores y el interior multiplicado por un peso para compensar la diferencia de áreas.
- Características de cuatro rectángulos que computan la diferencia entre pares diagonales de rectángulos.

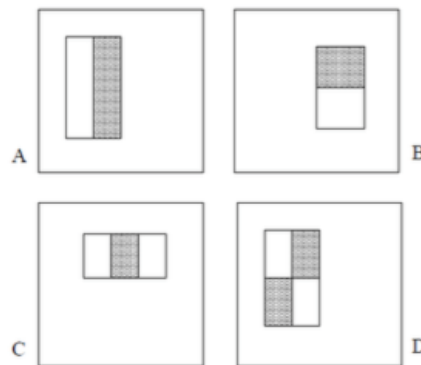


Figura. 4.4.1.3 Ejemplos de características de dos, tres y cuatro rectángulos [40].

- **Imagen Integral:** Se emplea como una herramienta de computo rápido para cada uno de los rectángulos. La integral de una imagen respecto un punto  $x$ ,  $y$  radica en la sumatoria de los píxeles por arriba y a la izquierda de dichos puntos, incluyendo  $x$ ,  $y$  [42]. La función matemática se describe a continuación:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (6)$$

Este método permite el uso de programación dinámica para la adquisición de valores dentro de la imagen a través de otros valores calculados previamente.

- **Proceso de aprendizaje:** Su función principal es obtener la cascada de clasificadores mediante un entrenamiento supervisado, esto se realiza mediante el uso de un pseudocódigo basado en AdaBoost, el cual es un meta-algoritmo adaptativo de machine learning [40].

En Viola-Jones el algoritmo AdaBoost elige entre un gran conjunto de filtros, las características de Haar, con la finalidad de seleccionar en cada instante cuál de ellos se ajusta mejor al elemento a clasificar.

- **Cascada de decisión:** Viola-Jones maneja un árbol binario de clasificadores para decidir si una región se trata de una cara o no. Esta cascada está formada por nodos en los que se analizan las diferentes características de las diferentes regiones de la imagen. En la Figura 4.4.1.4 se observa la estructura de una cascada de decisión.

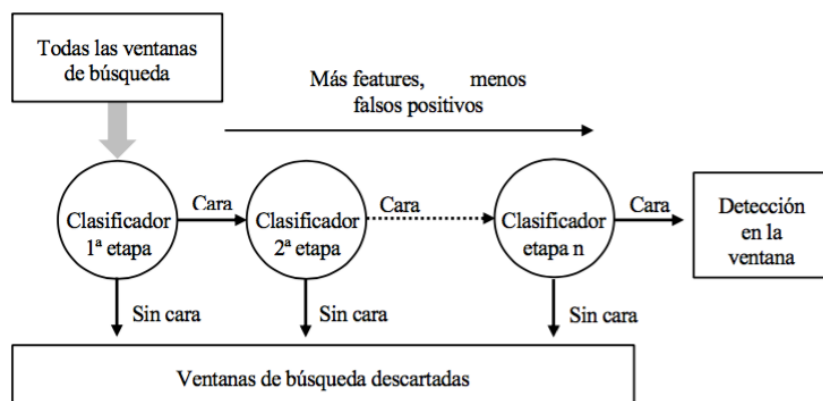


Figura. 4.4.1.4 Estructura de un clasificador tipo cascada [40].

En la etapa de detección de rostros se aplica el algoritmo de Viola-Jones descrito anteriormente, por su característica principal que es el bajo coste computacional, este algoritmo está disponible en la librería OpenCV lo que

facilita su incorporación en los Scripts de Python. La Tabla 4.4.1.3 presenta las funciones empleadas en la detección de rostros.

Tabla 4.4.1.3 Funciones utilizadas en la etapa de reconocimiento de rostros.

Función	Descripción
F_Cas = cv2.CascadeClassifier()	Crea e inicia el archivo cascada de rostros (archivo XML), lo coloca en memoria para que esté disponible para su uso.
faces = F_Cas.detectMultiScale(gray, 1.3, 5)	Detecta el rostro real de la imagen, 1.3 es el factor de escala para los rostros, 5 representa en número de rostros a detectar.
cv2.rectangle(imagen, (x,y), (x+w, y+h), (255, 255, 0), 2, 1)	Función que grafica un rectángulo sobre el rostro detectado, el valor (255, 255, 0) representa el color del borde del rectángulo, mientras que los valores 2 y 1 representan el grosor de la línea y el tipo de línea respectivamente.

#### 4.4.2. Clasificación

El algoritmo encargado de la clasificación será una red neuronal convolucional, la cual presenta una ventaja frente a las redes neuronales convencionales al ser capaces de clasificar imágenes con mayor precisión. La red neuronal se enfocará en identificar las emociones del ser humano.

Los pasos realizados para la creación de la red neuronal convolucional son:

- Preparación de datos
- Creación del modelo
- Entrenamiento del modelo y evaluación del modelo

Se toma como referencia los procedimientos realizados por el investigador Swaraj Jena en su proyecto titulado “Emotion Recognition using Deep Learning”, en el cual realiza la comparación entre dos modelos de redes neuronales convolucionales, a fin de establecer el porcentaje de precisión de cada red [43].

#### **A. Preparación de datos**

Cuando se trabaja con algoritmos basados en redes neuronales, es primordial que los datos empleados sean preparados en un formato adecuado para evitar errores en la depuración del código. La base de datos Fer2013 dispuesto por Kaggle Inc, la cual consiste en una serie imágenes de rostros de 48x48 píxeles en escala de grises, y registrado automáticamente para que el rostro esté más o menos centrada [44], cumple con el condicionante de preparación de datos. La Figura 4.4.2.1 muestra los datos de Fer2013.

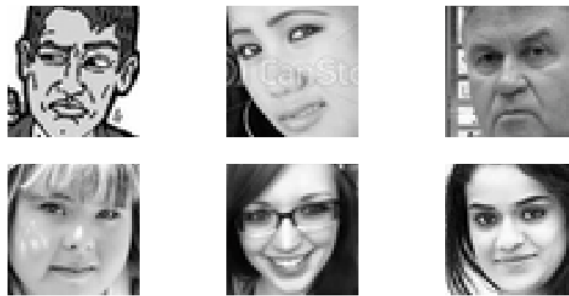


Figura. 4.4.2.1 Imágenes de la base de datos Fer2013 [44].

En la Figura 4.4.2.2 se observa como está estructurado la base de datos Fer2013, consta de 3 columnas con los argumentos emoción, píxeles y el uso, así como la cantidad de elementos dispuestos para el entrenamiento en un total de 28.709 y para el test con 3.589 datos públicos y 3589 datos privados, mediante el uso de funciones del paquete Pandas para Python se fusionaron los datos públicos y privados para tener una base de datos de testeo o prueba mayor.



```

Base de datos FERC-2013

Datos de la Columna ['emotion' 'pixels' 'Usage']

Valores de uso unico ['Training' 'PublicTest' 'PrivateTest']

Tamaño de uso:

--- Training 28709
--- PublicTest 3589
--- PrivateTest 3589

Fusión de conjuntos de datos privados y públicos

Nuevo tamaño de datos de uso:

--- Training 28709
--- Testing 7178

```

Figura. 4.4.2.2 Descripción de Fer2013.

La forma gráfica de cómo están distribuidos los datos de acuerdo al tipo de emoción (etiqueta) se puede visualizar en la Figura 4.4.2.3. La aplicación encargada de graficar los datos es la función `plot_bar_graph` desarrollada empleando la librería `Matplotlib`, la cual recibe la cantidad de datos y las etiquetas de cada emoción, la Figura 4.4.2.4 muestra el código de la función.

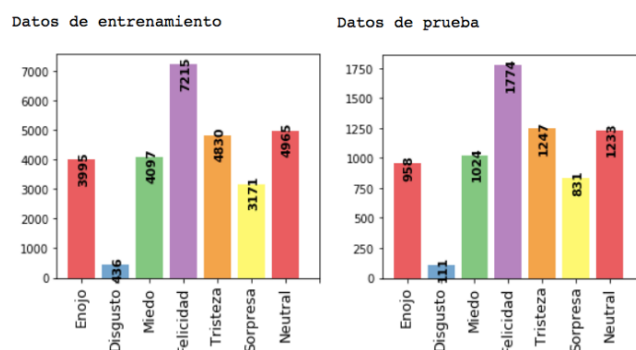


Figura. 4.4.2.3 Distribución de los datos de entrenamiento y test.

```

def plot_bar_graph( y ,xlabels):
    fig = plt.figure(figsize=(12,3))
    ax = fig.add_subplot(1,6,6)
    ax.bar(np.arange(0,len(y)), y, color=set3,alpha=0.5)
    ax.set_xticklabels(xlabels, rotation=90, fontsize=10)

    ax.set_xticks(np.arange(0.5,len(y)+0.5,1))

    for i, v in enumerate(y):
        ax.text(i+0.25, v, str(v), color='blue', fontweight='bold',rotation=90,)
    plt.tight_layout()
    plt.show()

```

Figura. 4.4.2.4 Función para la creación de gráficos de barras.

La cantidad de datos destinada a la emoción disgusto para el entrenamiento es de 436 y para el test o prueba es de 111, estos valores son relativamente bajos en comparación a los otros, por ese motivo se decidió combinar los datos de disgusto con los datos de enojo, ya que estas dos emociones tienden a ser similares. La Figura 4.4.2.5 muestra la distribución de los datos modificados.

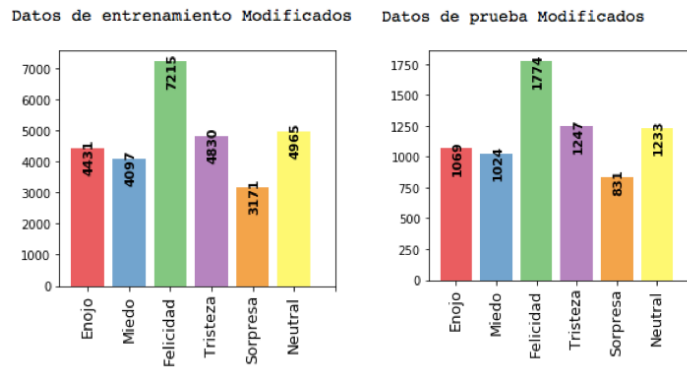


Figura. 4.4.2.5 Distribución de datos modificados.

## B. Creación del modelo de red neuronal convolucional (CCN)

Una vez organizado correctamente los datos el siguiente paso es la creación de la red neuronal. En la Figura 4.4.2.6 se puede visualizar las capas añadidas al modelo de red neuronal convolucional.

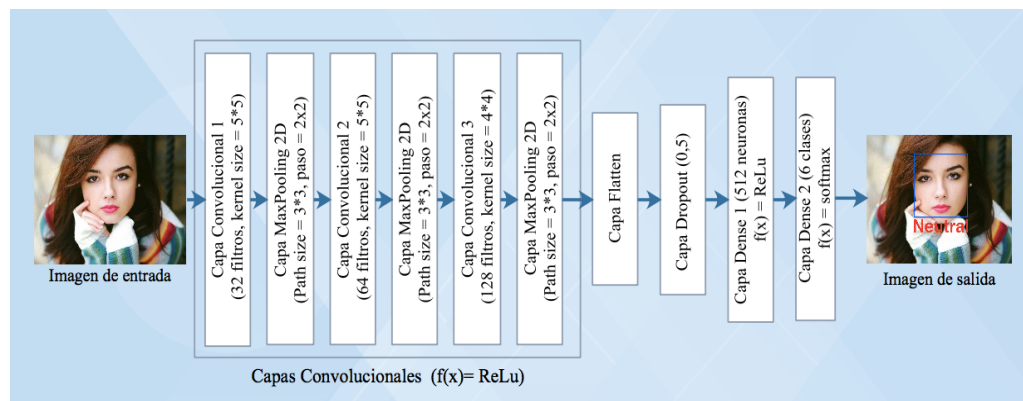


Figura. 4.4.2.6 Estructura de la red neuronal convolucional.

A través del uso de la librería Keras dedicada a la codificación de algoritmos de redes neuronales, se crea el modelo de red neuronal convoluciona planteada

en la Figura 4.4.2.6. El primer paso es la importación de las funciones de Keras para la creación de la red, estos son:

- **Modelo Secuencial (keras.models sequential):** determina el tipo de modelo de red a construir, el modelo a crea es lineal por lo que se emplea Sequential().
- **Capas (keras.layers):** importa las capas necesarias para la red neuronal, estas pueden ser: Dense, Dropout, Activation, Flatten, etc.
- **Capas convolucionales:** Importa las capas de convolución, por ejemplo: Conv2D, MaxPooling2D

Luego de importar las funciones se construye la red, esto se realiza mediante el método add() que proporciona Keras y sirve para añadir capas a la red.

- **Primera Capa:** Capa convolucional (Conv2D) con 32 filtros, el tamaño del filtro es de 5x5 pixeles, tiene un relleno con el mismo valor (0), el tamaño de la variable de entrada (input\_shape) es 48x48 pixeles y 1 por ser un dato en escala de grises, por último se asigna como función de activación el rectificador unidad de rectificación lineal (ReLU) [45]-[29], el cual solo permite el paso de los valores positivos, mientras que los valores negativos toman un valor de 0.
- **Segunda Capa:** Capa de reducción (MaxPooling2D) con una reducción de 3 (tercera parte) y una zancada de 2.
- **Tercera Capa:** Capa convolucional (Conv2D) con 64 filtros de tamaño 5x5 pixeles, relleno con el mismo número (0) y emplea como función de activación el rectificador ReLU.
- **Cuarta Capa:** Capa de reducción (MaxPooling2D) con una reducción de 3 (tercera parte) y una zancada de 2.
- **Quinta Capa:** Capa convolucional (Conv2D) con 128 filtros de tamaño 4x4 pixeles, relleno con el mismo número (0) y emplea como función de activación el rectificador ReLU.
- **Sexta Capa:** Capa de reducción (MaxPooling2D) con una reducción de 3 (tercera parte) y una zancada de 2.

- **Séptima Capa:** Capa central de aplanamiento (Flatten), la cual almacena (aplana) los datos en un arreglo unidimensional.
- **Octava Capa:** Capa dropout que desactivará de manera aleatoria neuronas durante el entrenamiento, esto permite evitar el sobreajuste de la red, el valor de activación es de 50% (0.5)
- **Novena Capa:** Capa densa u oculta (Dense) que cuenta 512 neuronas las cuales están conectadas a todas las neuronas de la capa dropout, y su función activación es ReLU.
- **Decima Capa:** Capa densa (Dense), cuenta con 6 neuronas que son las categorías de salida y la función de activación es Softmax, el cual emite un valor comprendido entre 0 y 1 [45].

Las capas de convolución tienen la finalidad de capturar la información de mayor relevancia de la imagen de entrada como los detalles del rostro, a la salida de cada capa convolucional se encuentra una capa de reducción (MaxPooling2D) que permite reducir el tamaño de los datos a procesar, optimizando significativamente el uso de los recursos computacionales. La Figura 4.4.2.7 muestra el código de las capas asignadas a la red neuronal convolucional.

```

model = Sequential()

model.add(Conv2D(32, (5,5), padding='same', input_shape=(48,48,1), activation="relu"))
model.add(MaxPooling2D((3,3), strides=(2,2)))

model.add(Conv2D(64, (5,5), padding='same', activation="relu"))
model.add(MaxPooling2D((3,3), strides=(2,2)))

model.add(Conv2D(128, (4,4), padding='same', activation="relu"))
model.add(MaxPooling2D((3,3), strides=(2,2)))

model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu'))
model.add(Dense(6, activation='softmax'))

```

Figura. 4.4.2.7 Capas asignadas a la red neuronal convolucional (Keras-Python).

Una vez definida todas las capas de la red neuronal convolucional, esta se procede a compilar, para ello se emplea la función compile () de Keras. La Tabla 4.4.2.1 muestra la descripción de la función mencionada.

Tabla 4.4.2.1. Función para compilación del modelo CNN

Función	Descripción
<code>model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])</code>	La función compile() traduce el lenguaje de alto nivel a binario, y necesita de los parámetros: optimizador, función de pérdida y la métrica del modelo.
<code>model.summary()</code>	Función que imprime en pantalla la estructura de la red compilada, Figura 4.4.2.8.

Los parámetros para compilar el modelo son los descritos a continuación:

- **Optimizador:** ADAM, método para la optimización estocástica, que calcula las tasas de aprendizaje adaptativo individuales para diferentes parámetros a partir de las estimaciones de los primeros y segundos momentos de los gradientes [46].
- **Función de Pérdida o función objetivo:** CROSSENTROPY CATEGORY, se emplea como un método probabilístico cuando la salida tiene varias clases, para este caso se tiene 6 clases.
- **Métrica o función de evaluación:** ACCURANCY, determina la precisión en el reconocimiento de expresiones del modelo de red neuronal convolucional.

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	832
max_pooling2d_1 (MaxPooling2D)	(None, 23, 23, 32)	0
conv2d_2 (Conv2D)	(None, 23, 23, 64)	51264
max_pooling2d_2 (MaxPooling2D)	(None, 11, 11, 64)	0
conv2d_3 (Conv2D)	(None, 11, 11, 128)	131200
max_pooling2d_3 (MaxPooling2D)	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dropout_1 (Dropout)	(None, 3200)	0
dense_1 (Dense)	(None, 512)	1638912
dense_2 (Dense)	(None, 6)	3078
Total params: 1,825,286		
Trainable params: 1,825,286		
Non-trainable params: 0		

Figura. 4.4.2.8 Descripción de la red neuronal convolucional (CCN) en Keras.

### C. Entrenamiento y evaluación del modelo

Creada la arquitectura del modelo de red neuronal se procede a su entrenamiento. El tipo de entrenamiento para el sistema planteado es de carácter supervisado, debido a que se conoce la respuesta de los datos de entrenamiento como los de evaluación o prueba.

En la etapa de entrenamiento se utilizaron las siguientes funciones que provee la librería Keras:

- **ImageDataGenerator():** Crea variantes de la base de datos para obtener un volumen mayor de datos para el entrenamiento.
- **fit():** Normaliza los datos variantes.
- **model.fit\_generator():** Función que entrena el modelo en el cual se especifican los parámetros de entrenamiento, como por ejemplo, datos de entrenamiento, épocas, lote, etc.
- **model.save\_weights():** Guarda los datos de la red entrenada, esto permite almacenar los pesos (W) en un archivo con extensión .h5 que luego se utiliza para validar el modelo con los datos de prueba.

- **model.evaluate():** Función que evalúa el modelo neuronal entrenado, para ello utiliza como parámetro los datos de prueba.

El entrenamiento de la red neuronal se realizó durante 100 épocas, el tamaño de lote asignado fue de 128 dando como resultado 224 valores a procesar por cada época. Al finalizar el entrenamiento se validó el comportamiento del modelo durante el entrenamiento de 9 horas y 27 minutos, como se aprecia en la Figura 4.4.2.9.



Figura. 4.4.2.9 Gráficas del comportamiento de la red neuronal.

Se puede ver que la función de pérdida del modelo tiende a disminuir en la iteración 100 mientras que la precisión tiende a incrementar. El valor de precisión obtenido en la validación del modelo es mayor al obtenido en el entrenamiento. La precisión que obtuvo el modelo para la detección de gestos faciales es del 61% como se observa en la Figura 4.4.2.10.

```
7178/7178 [=====] - 34s 5ms/step
Test loss : 1.017
Test accuracy : 0.611
Testing finished
```

Figura. 4.4.2.10 Resultados de la evaluación del modelo.

#### 4.4.3. Movimiento del robot

El módulo de clasificación envía el dato de la emoción detectada al subsistema movimiento del robot, el cual tiene por objetivo comparar el dato y asignar el movimiento de su base de datos, si el valor obtenido es verdadero entonces escribe en el puerto serie el bloque de movimientos de la emoción comparada. En la Figura 4.4.3.1 se visualizan diagrama de flujo del módulo.

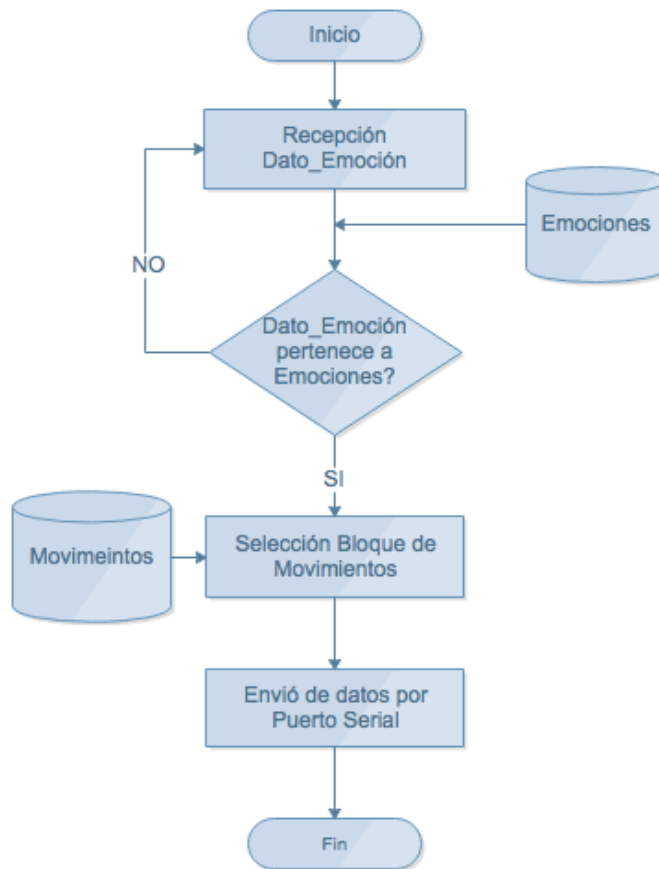


Figura. 4.4.3.1 Diagrama de flujo del módulo movimiento del robot.

Ingresa el dato de la emoción detectada por el módulo clasificador a ser comparado con una base de emociones, si este dato pertenece a la base de datos emociones se elige el bloque de movimientos correspondientes a la emoción detectada, para enviar los movimientos por el puerto serial a la placa controladora de servos para mover el robot. En la Figura 4.4.3.2 se visualiza el código empleado para leer y escribir en el puerto serial (se emplea la librería PySerial – Python).

```

import serial
import time

ser = serial.Serial('/dev/tty.usbserial', 115200, timeout = 1)

def Write_Serial_Port(SendData):
    ser.write(SendData.encode())
    time.sleep(0.01)
    return
  
```

Figura. 4.4.3.2 Código para escribir en el puerto serial.



#### 4.4.4. Pruebas y resultados

##### A. Funcionamiento de la red neuronal.

Los resultados obtenidos en la evaluación del modelo se observan en la Figura 4.4.4.1. Para obtener estos datos se realiza el proceso de leer los datos de evaluación o prueba normalizados de la base datos Fer2013, enviarlos al modelo para su predicción y mediante la implementación de una función de la librería Matplotlib – Python se puede visualizar los resultados obtenidos en diagramas de barras.

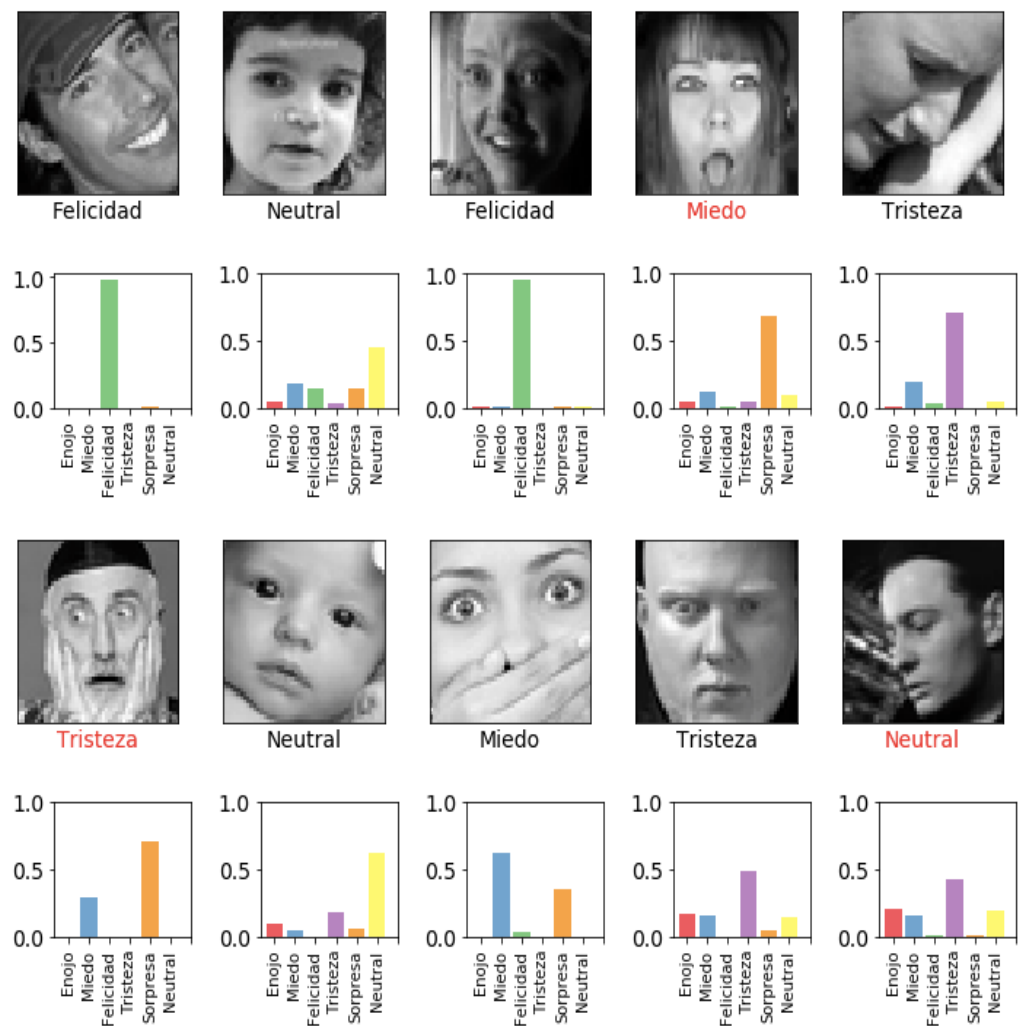


Figura. 4.4.4.1 Resultados obtenidos de la predicción del modelo.

Como se puede apreciar en la figura anterior se ve la clasificación obtenida por el modelo, existen imágenes con etiquetas de color rojo que muestran la emoción errónea que ha establecido el modelo.

Existe el 39% donde el modelo de red neuronal tendiera a predecir los datos de forma incorrecta. Para observar de forma más gráfica el comportamiento de la red se utiliza las llamadas matrices de confusión, las cuales nos permiten visualizar el comportamiento de los modelos bajo un aprendizaje supervisado [47].

La matriz de confusión se grafica mediante el uso la función `confusion_matrix()` que proporciona la librería `Sklearn.metrics` para el desarrollo de aplicaciones de aprendizaje profundo en Python, la cual necesita como parámetros, las etiquetas de las emociones, los datos pronosticados y el formato de color de la matriz, la Figura 4.4.4.2 muestra el pseudocódigo empleado.

```
from sklearn.metrics import confusion_matrix
labels = ['Enojo', 'Miedo', 'Felicidad', 'Tristeza', 'Sorpresa', 'Neutral']

def graficar_matriz_confusion(y_true, y_pred, cmap=plt.cm.Blues):

    cm = confusion_matrix(y_true, y_pred)
    i=0
    fig = plt.figure(figsize=(6,6))
    matplotlib.rcParams.update({'font.size': 16})
    ax = fig.add_subplot(111)
    matrix = ax.imshow(cm, interpolation='nearest', cmap=cmap)
    fig.colorbar(matrix)
    for i in range(0,6):
        for j in range(0,6):
            ax.text(j,i,cm[i,j],va='center', ha='center')

    ticks = np.arange(len(labels))
    ax.set_xticks(ticks)
    ax.set_xticklabels(labels, rotation=45)
    ax.set_yticks(ticks)
    ax.set_yticklabels(labels)
    plt.tight_layout()
    plt.ylabel('Etiquetas Asignadas')
    plt.xlabel('Etiquetas Pronósticadas')
```

Figura. 4.4.4.2 Seudocódigo de matriz de confusión

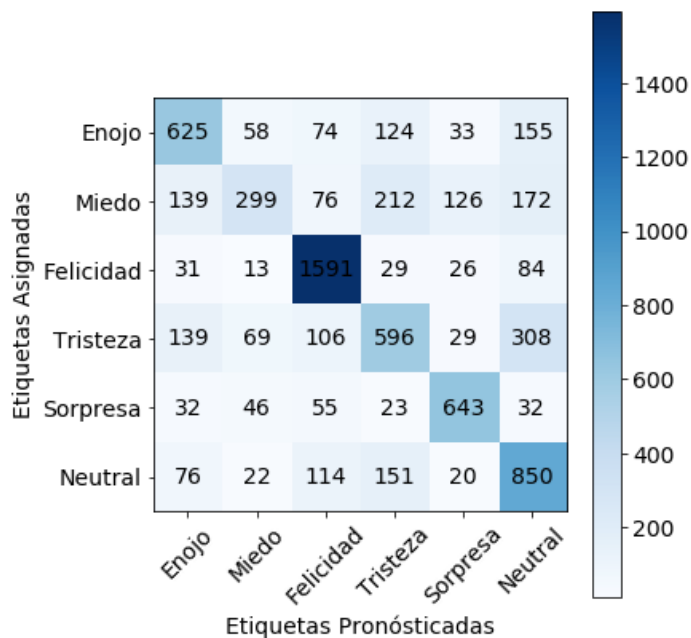


Figura. 4.4.4.3 Matriz de Confusión del modelo de red neuronal.

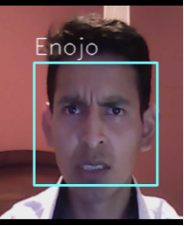



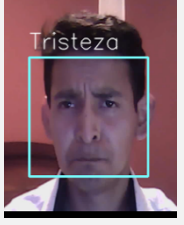

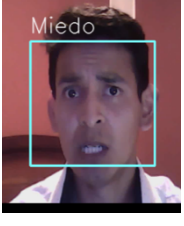

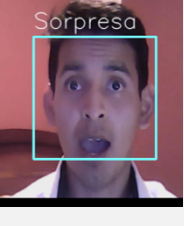



En la Figura 4.4.4.3 se observa el número de datos pronosticados por la red neuronal convolucional para cada etiqueta asignada, dando como resultados: para la etiqueta enojo 625 datos correctos, miedo con 299, felicidad con 1591, tristeza con 596, sorpresa con 643 y para la etiqueta neutral 850 datos correctos. Existen imágenes donde la variación de un gesto se puede considerar como un gesto de tristeza o miedo entonces el comportamiento de la red tiende a confundirla y pronostica una etiqueta incorrecta, esto se puede ver en la fila número 2 de la matriz de confusión donde muestra los datos acertados para la etiqueta miedo no divergen mucho de la etiqueta tristeza con un valor de 212 aciertos.

El modelo de red neuronal propuesto para el control de gesticulación del robot social interactivo con aspecto humano cumple la condición de detectar el gesto generado por la persona, el cual tienen un valor de 61% en su predicción.

### B. Funcionamiento del controlador de gesticulación

Los resultados obtenidos de la implementación del controlador para una imagen de entrada en tiempo real se visualiza en la Tabla 4.4.4.1.

Tabla 4.4.4.1. Resultados del controlador de gesticulación del robot social.

Gestos	Gesto reconocido (CNN)	Gesto del robot
Enojo		
Felicidad		
Tristeza		
Miedo		
Sorpresa		
Neutral		

Para cada rostro detectado por el reconocimiento facial y mediante el algoritmo de clasificación se va asignando la etiqueta de acuerdo al tipo de gesto comprendido entre: enojo, felicidad, tristeza, miedo, sorpresa y neutral.

Una vez asignado el gesto se envía el bloque de movimientos al controlador. Los bloques de movimientos generados se visualizan en la Figura 4.4.4.4. La configuración de posición para cada uno de los servomotores del robot social, que generan los diferentes gestos faciales se encuentra en el Anexo B.

```
Gesto detectado: Enojo
Bloque de Movimientos
#16 P1500 #17 P1500 #20 P1200 #21 P1500 #22 P1800 #23 P1500 #24 P2000 #0 P1400 #1 P1500 #2 P1540 #3 P1500 #4 P1800 #5 P1100 T200
Gesto detectado: Miedo
Bloque de Movimientos
#16 P1600 #17 P1300 #20 P1300 #21 P1700 #22 P1700 #23 P1200 #24 P2000 #0 P1400 #1 P1500 #2 P1490 #3 P1540 #4 P1450 #5 P1500 T200
Gesto detectado: Neutral
Bloque de Movimientos
#16 P1700 #17 P1200 #20 P1300 #21 P1400 #22 P1700 #23 P1600 #24 P2000 #0 P1400 #1 P1500 #2 P1600 #3 P1450 #4 P1600 #5 P1300 T200
Gesto detectado: Sorpresa
Bloque de Movimientos
#16 P1600 #17 P1300 #20 P1200 #21 P1500 #22 P1800 #23 P1500 #24 P1500 #0 P1400 #1 P1500 #2 P1450 #3 P1570 #4 P1350 #5 P1550 T200
```

Figura. 4.4.4.4. Datos del controlador de la gesticulación del robot.

Cada vez que una imagen es detectada en por la cámara esta pasa al clasificador que la procesa dando como resultado la etiqueta o el valor al que pertenece y los movimientos que deben realizar los servomotores para tratar de imitar el gesto etiquetado.

Como ejemplo práctico se analiza el bloque de movimientos para el gesto enojo (Tabla 4.4.4.2):

- ✓ Los servomotores a moverse a la posición marcado por el gesto detectado son un total de 12
- ✓ La codificación de la cadena de caracteres se envía en una sola cadena con todas las posiciones dadas y con un tiempo de ejecución del movimiento de 200 milisegundos (T200).
- ✓ Algunos de los motores no presentan cambios en su posición inicial debido al patrón de repetición.

Tabla 4.4.4.2. Movimientos para el gesto enojo.

Movimientos	Numero (#n)	Posición (P)
Movimiento de labio Inferior	16	1500
Movimiento de labio Superior	17	1500
Movimiento de labio derecho superior	20	1200
Movimiento de labio derecho inferior	21	1500
Movimiento de labio izquierdo superior	22	1800
Movimiento de labio izquierdo inferior	23	1500
Movimiento de boca	24	2000
Movimientos de ojos izquierda-derecha	0	1400
Movimientos de ojos arriba-abajo	1	1500
Movimiento de parpado derecho	2	1540
Movimiento de parpado izquierdo	3	1500
Movimiento de ceja derecha	4	1800
Movimiento de ceja izquierda	5	1100

El controlador depende de todos los módulos que lo forman, dado que, si el procesamiento de imagen es pobre y la información que ingresa al algoritmo clasificador presenta mucho ruido, el mismo no será capaz de etiquetarlo correctamente generando un error en la gesticulación del robot. Del mismo modo si el bloque de movimientos debe ser codificado de manera correcta para que la tarjeta controladora pueda ejecutar la orden de mover los servomotores a la posición asignada. Como último punto se puede afirmar que el controlador cumple con la condición de generar el movimiento de acuerdo al gesto reconocido.

#### 4.5. Sistema de sincronización labial

La sincronización labial del robot interactivo social representa una característica verosímil a la forma en que los seres humanos nos comunicamos mediante el habla. En la Figura 4.5.1 se puede apreciar las funciones a desarrollar para el control sincronizado labial de robot social.

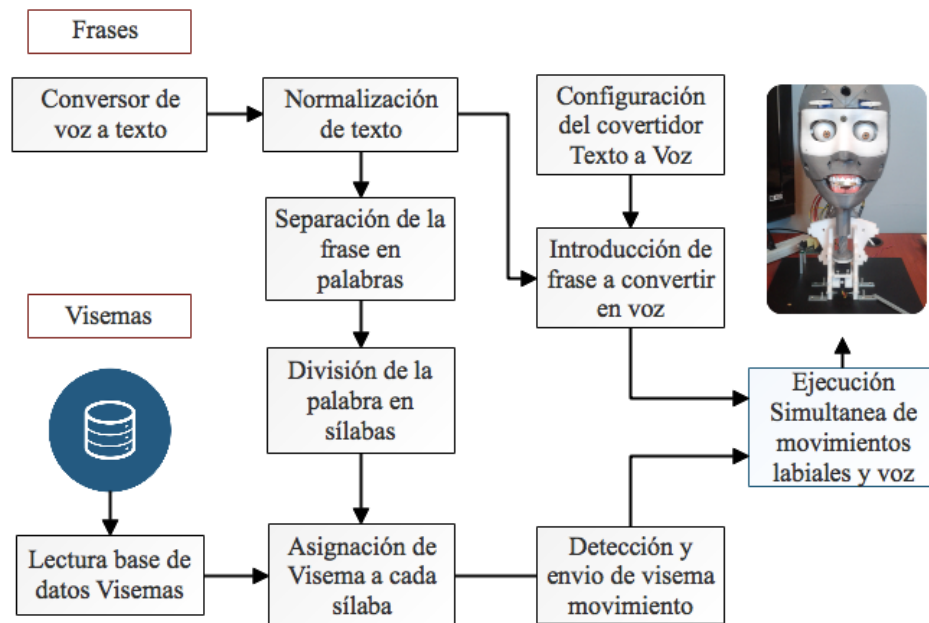


Figura. 4.5.1 Esquema secuencial de sincronización labial del robot social.

Cada ítem mostrado en el esquema funcional de sincronización labial fue programado en el lenguaje Python a través del uso de funciones creadas para el procesamiento de texto, adicionalmente se emplearon las librerías: SpeechRecognition y Lxml para convertir archivos de audio a texto y manipulación de base de datos con extensión XML.

##### 4.5.1. Convertidor de texto a voz

Hoy en día existen conversores de voz tanto de software libre o como aplicaciones de pago, los cuales presentan características distintas en el resultado de sintetizar la voz, en la siguiente tabla se realiza una comparativa entre los diferentes conversores de texto a voz disponibles en el mercado.

Tabla 4.5.1.1 Característica de los conversores de texto a voz disponible.

<b>Conversor</b>	<b>Voz</b>	<b>Amplitud de voz</b>	<b>Lenguaje</b>	<b>Velocidad</b>	<b>Comentarios</b>
Loquendo	Hombre Mujer Niños	Configurable	Varios	Configurable	Dispone de librerías para Android y Windows
Google TTS	Hombre Mujer Niños	Configurable	Varios	Configurable	Trabaja bajo servidor en línea
DECTalk	Hombre Mujer Niños	Configurable	Ingles, Español, Español Latino	Configurable	Trabaja en línea y también en placas embebidas
EPSON	Hombre Mujer Niños	Configurable	Ingles, Español, Español Latino	Configurable	Trabajo bajo servidor
IBM Watson	Hombre Mujer	Configurable	Varios	Configurable	Trabaja bajo servidor online
Pyttax	Hombre Mujer	Configurable	Varios	Configurable	Dispone de librerías para Python
Mac OS TTS	Hombre Mujer	Automático	Varios	Configurable	Dispone de librerías para Python

Los conversores de texto mostrados en la Tabla 4.5.1.1 son los más sobresalientes y de mayor uso por la comunidad informática. La mejor opción que se apega a las necesidades del proyecto es Mac OS TTS, definida como una librería propia del sistema operativo de Macintosh y corre bajo el lenguaje de programación Python. En la Figura 4.5.1.1 se puede observar el código ejemplo del uso de la librería Mac OS TTS.



```
from os import system
system("say Soy un robot interactivo social")
```

Figura. 4.5.1.1 Código del conversor de texto a voz Mac OS TTS.

#### 4.5.2. Base de datos visemas.

Está enfocada en visemas las cuales son representaciones visuales de los fonemas. Se utilizó la base de datos Control\_Visemas.xml desarrollada en el proyecto investigativo del robot social DORIS [21]. En la Figura 4.5.2.1 se puede observar cómo está configurado el código de visema-sílaba. La estructuración final de la base de datos de visemas se puede observar en el Anexo C.

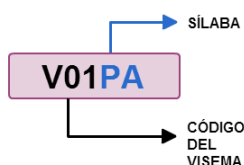


Figura. 4.5.2.1 Estructura del visema [21].

#### 4.5.3. Funciones del algoritmo de sincronización labial

Todos los algoritmos de sincronización labial están desarrollados en scripts del lenguaje de programación de Python, donde se han definido varias funciones específicas en diversos scripts para cumplir con el objetivo. Los mismos que conjuntamente logran la sincronización labial del robot. En la Figura 4.5.3.1 se muestra el diagrama de flujo de sincronización.

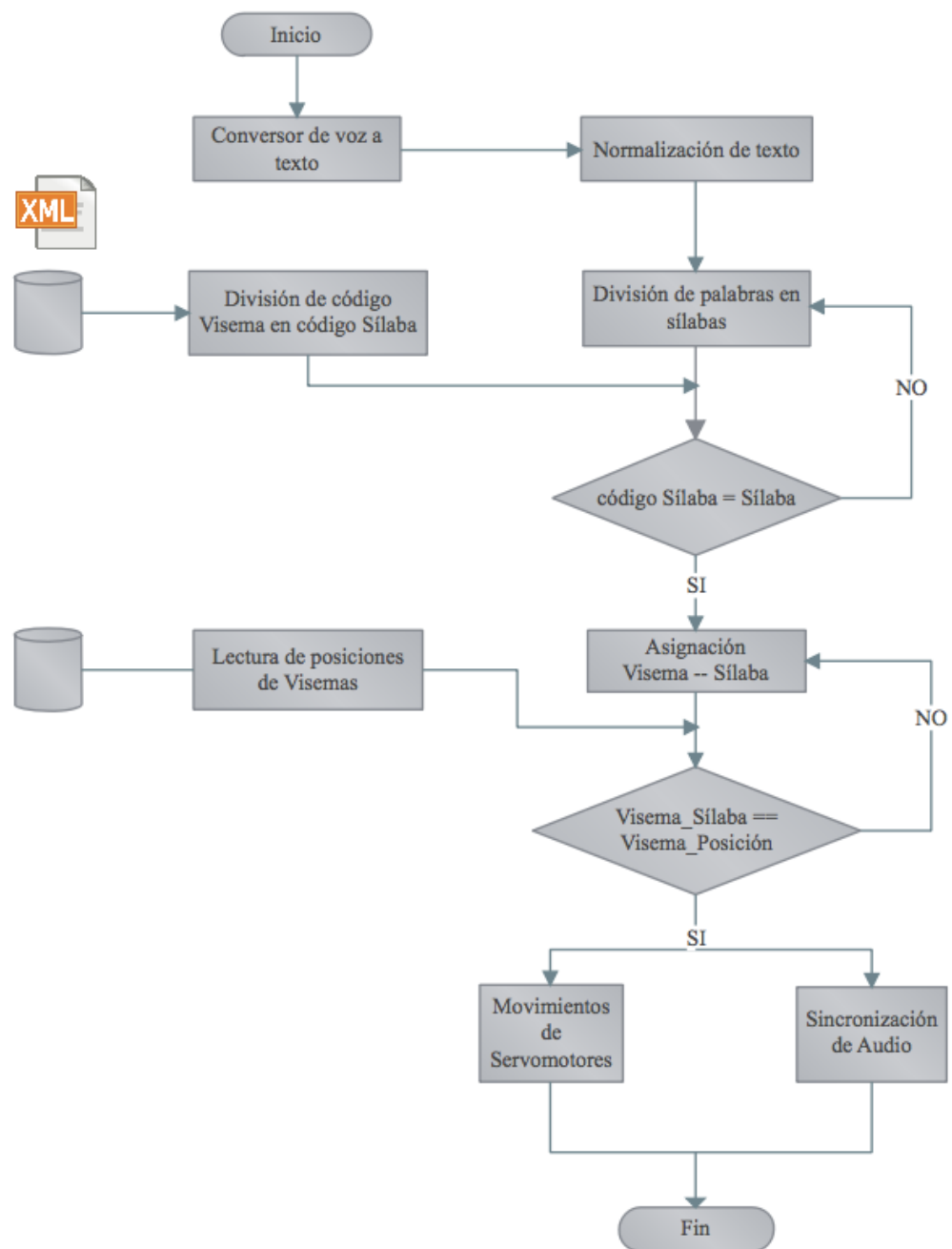


Figura. 4.5.3.1 Diagrama de flujo del algoritmo de sincronización labial.

**A. Conversor de voz a texto:** Es la función encargada de obtener el texto de un archivo de audio, para ello se emplea la API de Google para Python, que mediante una conexión a internet se puede obtener textos a partir de archivos de audio. En la Figura 4.5.3.2 se visualiza el diagrama de flujo del conversor de voz a texto.

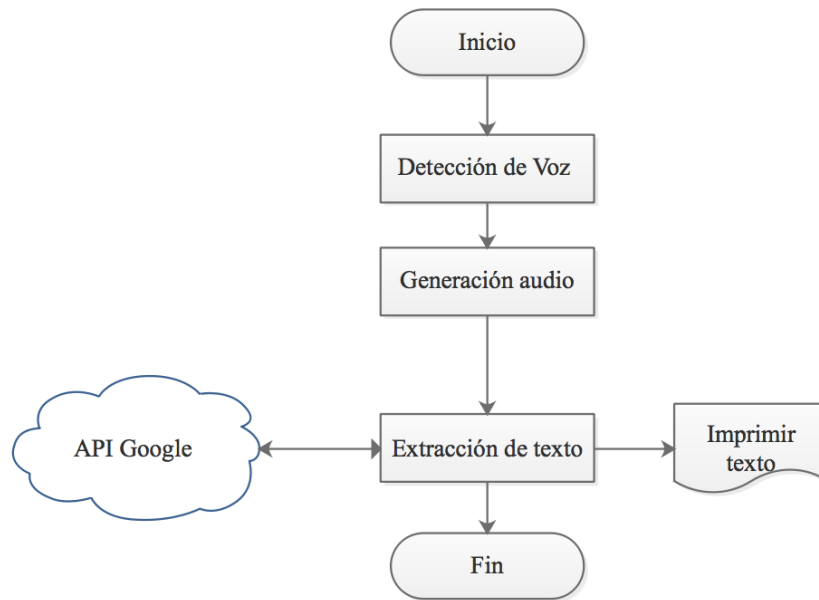


Figura. 4.5.3.2 Diagrama de flujo del conversor de voz a texto.

**B. Normalización del texto:** Proceso en el cuál el texto ingresado pasa a través de un filtro que tiene por objetivo eliminar símbolos, espacios, y otros caracteres diferentes de las 27 letras del alfabeto español, en la Figura 4.5.3.3 se puede observar el diagrama de flujo de la función normalización de texto.

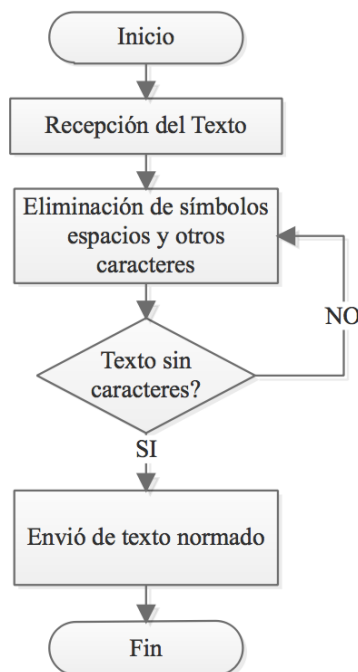


Figura. 4.5.3.3 Diagrama de flujo del subproceso Normalización del texto.

**C. Número de sílabas y división silábica:** La división silábica fue realizada con la consideración de las reglas de división del lenguaje español [19], por lo tanto, cada cadena de caracteres resultante del proceso de normalización ingresa al proceso de división silábica, donde cada palabra se descompone en sus respectivas sílabas y lo almacena en un arreglo para posteriormente ser asignado el movimiento de servomotores correspondiente a cada sílaba. La Figura 4.5.3.4 muestra el diagrama de flujo de la mención mencionada.

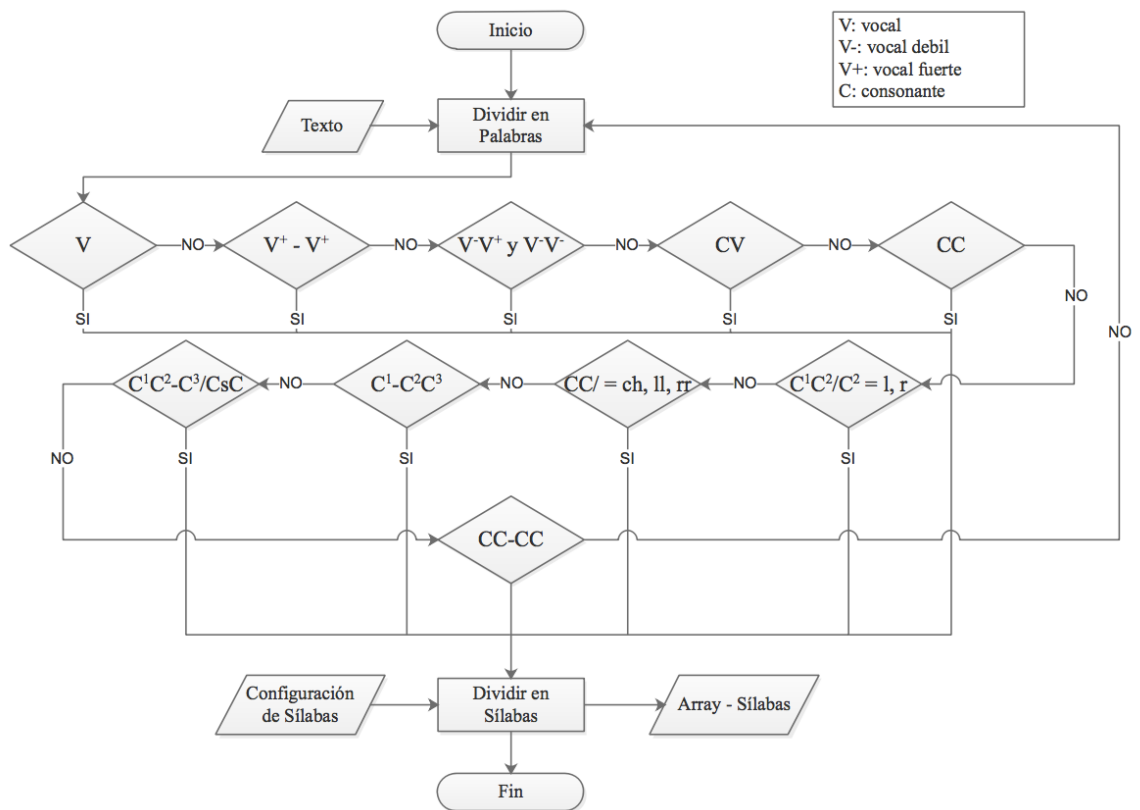


Figura. 4.5.3.4 Diagrama de flujo de la función división silábica.

**D. Conversor de texto a voz:** El conversor de texto utilizado es el Mac OS TTS propio del ordenador y el cual se puede usar en un pseudocódigo en Python; recibe el texto procesado por la parte de reconocimiento de voz y la convierte en una señal audible. En la Figura 4.5.3.5 se muestra el diagrama de flujo del conversor de texto a voz.

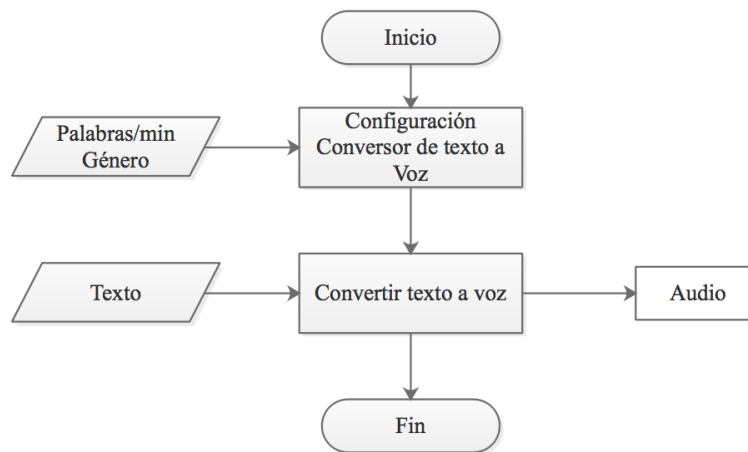


Figura. 4.5.3.5 Diagrama de flujo del convertor de texto a voz.

**E. Asignación de visemas:** Una vez realizada la división silábica, se forma un arreglo con todas las sílabas detectadas en el texto de entrada. Cada uno de los datos del arreglo es comparado con la base de datos `Codigo_Visemas.xml`, la cual se lee mediante las funciones que ofrece la librería `Lxml – Python`, posteriormente si la comparación es verdadera se van almacenado los datos de `Codigo_Visemas.xml` en un nuevo arreglo. En la Figura 4.5.3.6 se observa el diagrama de flujo del subproceso.

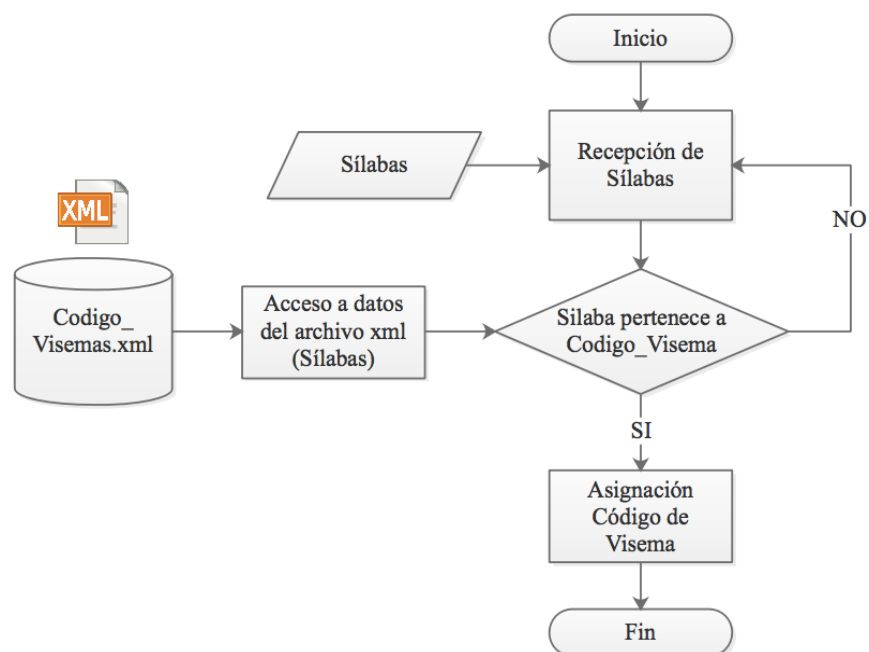


Figura. 4.5.3.6 Diagrama de Flujo de la función Asignación de Visemas

**F. Movimiento de Servomotores:** La entrada a este bloque es el arreglo Visemas el cual mediante una iteración en el programa se va comparando cada visema(i) con la posición de los 6 motores que corresponde a cada visema. La posición determinada es enviada mediante comunicación serial a la tarjeta controladora para que ejecute el movimiento. En la Figura 4.5.3.7 se muestra los diferentes procesos que conforman este bloque. La configuración de la posición para cada uno de los servomotores presentes en la boca del robot, los cuales generan el visema respectivo a la cadena de texto procesada se observan en el Anexo D.

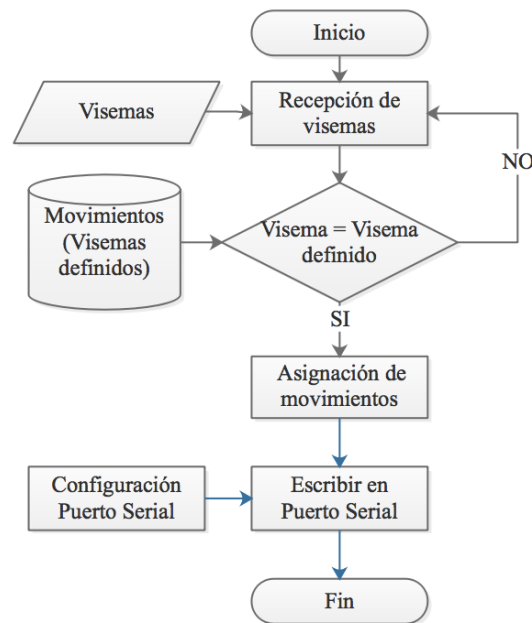


Figura. 4.5.3.7 Diagrama de flujo de la función movimiento de servomotores.















#### 4.5.4. Pruebas y resultados

##### A. Funcionamiento controlador del habla del robot interactivo social

Mediante la integración de un conversor de voz a texto se imprime el texto reconocido, el cual pasa por los bloques de normalización de texto, división y obtención del número de sílabas, después se asigna el código correspondiente de acuerdo al visema que corresponde, para finalmente enviar el bloque de movimientos de la cadena de texto identificada.

Luego de algunas pruebas con un total de 2.479 datos los visemas por sílaba definidos presentan similitudes con la mayoría de visemas-sílaba, por esta razón solo se obtuvo siete movimientos que representan a los visemas principales, estos siete movimientos se pueden apreciar en la Tabla 4.5.4.1.

Tabla 4.5.4.1. Movimientos de visemas obtenidos.

Nombre	Visema definido	Visema generado (robot)
Visema A (A)		
Visema E (E)		
Visema I (I)		
Visema O (O)		
Visema U (U)		
Visema BMPV (BM)		
Visema silencio (N)		

## B. Prueba con texto “Soy un robot social”

Los resultados obtenidos para la cadena de texto “Soy un robot social” se muestran a continuación:

```
###Texto: Soy un robot social
###Sílabas separadas: ['SOY', 'UN', 'RO', 'BOT', 'SO', 'CIAL']
###Numero de silabas: 6
###Lectura y asignación de código visemas:
[V28SOY', 'V38UN', 'V31RO', 'V10BOT', 'V31SO', 'V21CIAL']
###Visema formado:
['O', 'I', 'N', 'U', 'N', 'O', 'N', 'BM', 'O', 'N', 'O', 'N', 'I', 'A', 'N']
```

El bloque de movimientos para cada servomotor se visualiza en la siguiente figura:

O:	#16	P1700	#17	P1300	#20	P1500	#21	P1200	#22	P1400	#23	P1700	#24	P1600	T200
I:	#16	P1600	#17	P1400	#20	P1200	#21	P1400	#22	P1800	#23	P1600	#24	P1900	T200
N:	#16	P1700	#17	P1200	#20	P1300	#21	P1400	#22	P1700	#23	P1600	#24	P2000	T200
U:	#16	P1800	#17	P1200	#20	P1500	#21	P1200	#22	P1400	#23	P1800	#24	P1800	T200
N:	#16	P1700	#17	P1200	#20	P1300	#21	P1400	#22	P1700	#23	P1600	#24	P2000	T200
O:	#16	P1700	#17	P1300	#20	P1500	#21	P1200	#22	P1400	#23	P1700	#24	P1600	T200
N:	#16	P1700	#17	P1200	#20	P1300	#21	P1400	#22	P1700	#23	P1600	#24	P2000	T200
BM:	#16	P1700	#17	P1200	#20	P1200	#21	P1700	#22	P1900	#23	P1300	#24	P2000	T200
O:	#16	P1700	#17	P1300	#20	P1500	#21	P1200	#22	P1400	#23	P1700	#24	P1600	T200
N:	#16	P1700	#17	P1200	#20	P1300	#21	P1400	#22	P1700	#23	P1600	#24	P2000	T200
O:	#16	P1700	#17	P1300	#20	P1500	#21	P1200	#22	P1400	#23	P1700	#24	P1600	T200
N:	#16	P1700	#17	P1200	#20	P1300	#21	P1400	#22	P1700	#23	P1600	#24	P2000	T200
I:	#16	P1600	#17	P1400	#20	P1200	#21	P1400	#22	P1800	#23	P1600	#24	P1900	T200
A:	#16	P1300	#17	P1700	#20	P1100	#21	P1500	#22	P1900	#23	P1500	#24	P1500	T200
N:	#16	P1700	#17	P1200	#20	P1300	#21	P1400	#22	P1700	#23	P1600	#24	P2000	T200

Figura. 4.5.4.1 Movimientos de servomotores.

Los resultados de la sincronización de audio y movimiento de los servomotores para el texto ingresado se aprecian en la Figura 4.5.4.2 que muestra el conjunto de visemas que forma un trozo de la señal de audio, señal que cuenta con 6 segmentos.

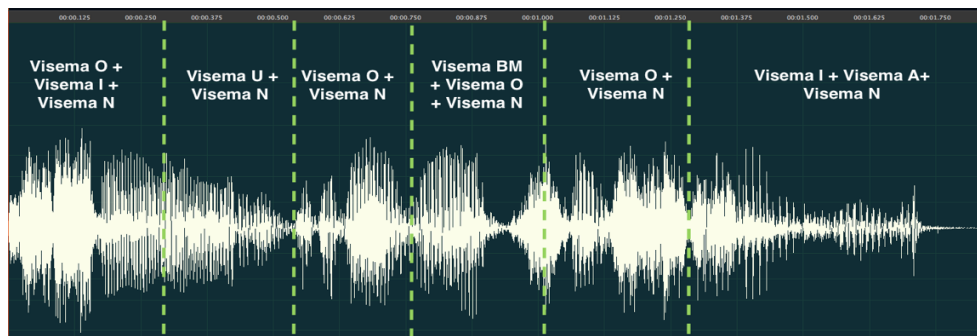


Figura. 4.5.4.2 Señal de salida de audio (voz robot)



En la Figura 4.5.4.3 se puede observar el comportamiento de los servomotores para cada conjunto de visemas enviado durante la sincronización. Hay un total de 7 servomotores que mediante la variación de la señal PWM forman el visema definido en cada bloque dando como resultado la formación de 15 visemas. Mediante el uso de Excel se graficó los datos de los servomotores obteniendo así la gráfica lineal del comportamiento cuando el robot habla.

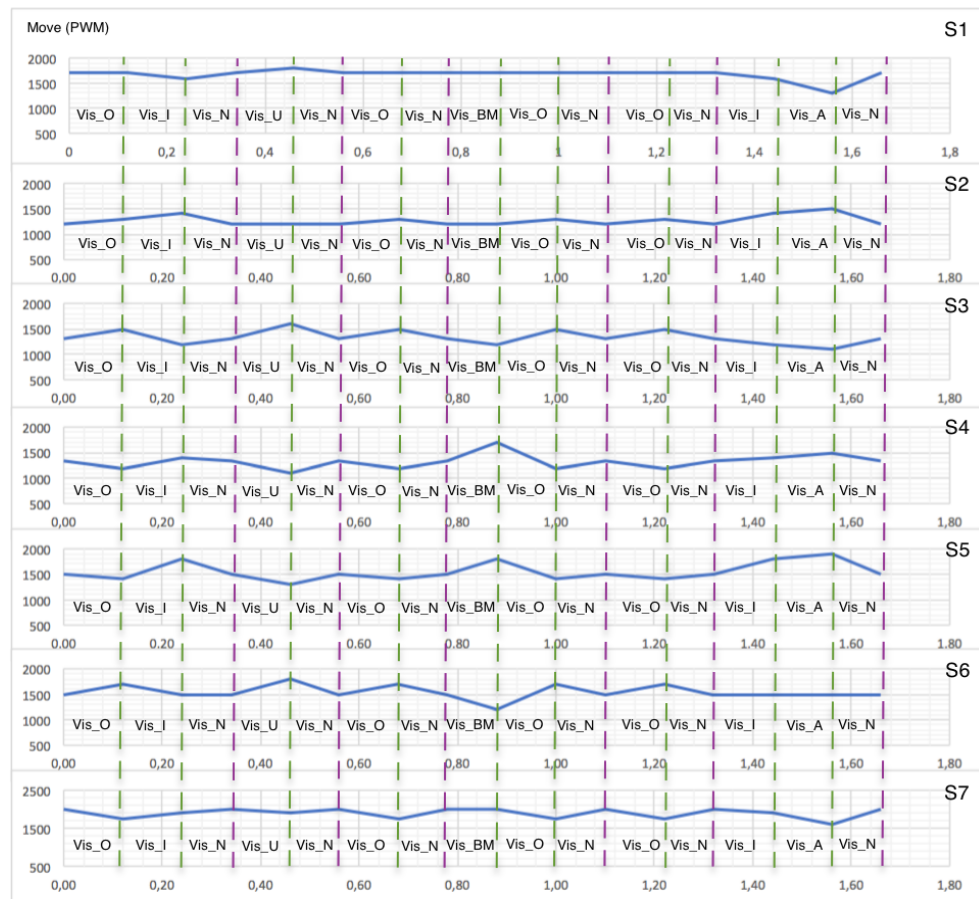


Figura. 4.5.4.3 Comportamiento de los servomotores durante la sincronización.

La coordinación entre la señal de audio y el movimiento de cada servomotor, inmerso en la postura visible del visema, presentan un tiempo: para el audio de 1,70 segundos y para el posicionamiento de servomotores de 1,69 segundos. La variación entre estos es mínima por lo que no influye en el resultado. En el Anexo A se puede apreciar el diagrama de funcionamiento del robot interactivo social, mientras que en los Anexos E, F, G se visualiza el código fuente que controla al robot .

## CAPITULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1. Conclusiones

A continuación, se presentan las conclusiones que se obtuvieron en el control de la gesticulación del robot social interactivo:

- El diseño del robot social cumple con la característica de ser similar al aspecto humano debido a que su estructura externa está basada en el robot humanoide InMoov, por otra parte, mediante el uso del software de diseño y modelado 3D (SolidWorks) se diseñaron los mecanismos que simulan el movimiento de ojos, parpados, cejas y labios del robot. Todos estos elementos fueron fabricados con la tecnología de impresión 3D, logrando así obtener como resultado la estructura de un robot social interactivo con aspecto humanoide.
- El robot social interactivo está provisto con 15 grados de libertad enlazados cada uno de estos a un actuador de tipo servomotor que actúan bajo la señal de modulación por ancho de pulso (PWM) provista por la placa controladora SSC-32, cuyo rango de actuación es de 500 mseg para 0 vcc y 2500 mseg para 5 vcc.
- A través del uso de la librería de Keras para el lenguaje de programación Python se incorporó la red neuronal convolucional al robot social interactivo diseñado, para que, al recibir una imagen de los gestos faciales de una persona, sea capaz de imitar dicho gesto, contribuyendo así al campo de la imitación de la expresión facial en tiempo real de un ser humano. El

sistema puede reconocer e imitar los gestos faciales humanos de manera correcta y precisa con una tasa de precisión del 61,1%. El tiempo en que el robot responde hacia el estímulo de entrada a través del gesto es relativamente rápido debido a su valor de 0,1seg.

- El sistema de sincronización desarrollado permite al robot simular los movimientos de la boca humana. La variación del tiempo de sincronización entre la señal de audio generada por la conversión de una cadena de texto en audio, y el posicionamiento secuencial de los servomotores respetando el código de visema detectado, es mínima por lo tanto se considera una sincronización eficiente.

## **5.2. Recomendaciones**

- En el diseño estructural del robot social interactivo se recomienda contar con instrumentos de precisión para dimensionar correctamente los componentes y piezas del mismo, debido a que los errores de diseño pueden ocasionar que los actuadores se atasquen evitando que el robot funcione correctamente.
- El sensor de video (cámara) empleado para la adquisición continua de imagen debe tener buenas especificaciones, tales como: alta resolución, un procesamiento igual o mayor a 30 fotogramas por segundo, conectividad, entre otros, debido a que se trabaja con un algoritmo que detecta patrones, y adicionalmente se debe ubicar en un lugar que permita captar de mejor manera la imagen, por otra parte, se puede optar por colocar la cámara fija a la base.
- Para lograr elevar la tasa de predicción de la red neuronal convolucional, se deberá trabajar con una base de datos de mayor tamaño, de la misma forma se debería remplazar el entrenamiento mediante CPU por la GPU que facilitará el proceso computacional y el modelo se podrá entrenar por un número mayor de épocas.

- En la sincronización labial dar mayor robustez al seudocódigo encargado de separar las palabras en sílabas y de asignar el código de visema, además aumentar la base de datos de las sílabas para aportar con cadenas de texto de mayor tamaño.
- Se deberá generar una función que permita calcular el tiempo para de sincronización entre el sonido de la palabra emitida y el movimiento de los motores para mejorar el tiempo sincronizado, el cual tiene un desfase de 0,01 seg.
- En la facultad de ingeniería en Sistemas, Electrónica e industrial se debería fortalecer los conocimientos en programación de algoritmos de control, enfatizando las herramientas actuales como el modelado de redes neuronales artificiales y sobre todo el análisis de datos.
- Para los trabajos e investigaciones a desarrollarse en el futuro sobre la temática de robótica social, se puede considerar el brindarle un sistema que estimule la generación o simulación de emociones y sentimientos al robot interactivo social.

## BIBLIOGRAFÍA

- [1] M. Jang, J. Kim, and B.-K. Ahn, “A software framework design for social human-robot interaction,” *2015 12th Int. Conf. Ubiquitous Robot. Ambient Intell.*, no. Urai, pp. 411–412, 2015.
- [2] P. Baxter and J. G. Trafton, “Cognitive architectures for human-robot interaction,” *Proc. 2014 ACM/IEEE Int. Conf. Human-robot Interact. - HRI '14*, pp. 504–505, 2014.
- [3] D. Hanson, A. Olney, I. a Pereira, and M. Zielke, “Upending the Uncanny Valley,” *Proc. Natl. Conf. Artif. Intell.*, vol. 20, no. 4, pp. 24–31, 2005.
- [4] Xinhua/Meng Dingbo, “Robot ‘Jiajia,’” *New interactive “robot goddess” unveiled in east China*, 2016. .
- [5] R. Carlos, “Diseño y construcción de un rostro robótico de apariencia humana, con capacidad de ejecutar gestos y emociones,” Universidad de las Fuerzas Armadas ESPE, 2015.
- [6] C. Astudillo and H. López, “Ensamblaje y control de un rostro robótico que simule los movimientos del rostro humano,” Escuela Politécnica Nacional, 2012.
- [7] V. Silva, F. Soares, and J. S. Esteves, “Mirroring and recognizing emotions through facial expressions for a RoboKind platform,” *Bioeng. (ENBENG), 2017 IEEE 5th Port. Meet.*, pp. 3–6, 2017.
- [8] C. Ruales, O. Arteaga, and D. Rivas, “Diseño y construcción de un rostro robótico de apariencia humana, con capacidad de ejecutar gestos y emociones,” pp. 1–9, 2015.
- [9] E. A. N. S. Edirisinghe, R. A. P. Lakmal, T. G. G. Lakshitha, S. W. D. T. S. Sirimanna, and A. G. B. P. Jayasekara, “Design of a human-like robot head with emotions,” *2nd Int. Moratuwa Eng. Res. Conf. MERCon 2016*, pp. 421–426, 2016.
- [10] C. Lapusan, C. R. Rad, S. Besoiu, and A. Plesa, “Design of a humanoid robot head for studying human-robot interaction,” *Proc. 2015 7th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2015*, p. WR15-WR18, 2015.
- [11] M. Mejía and J. Núñez, “Diseño y Construcción de un robot social para interacción hombre-máquina,” Universidad de las Fuerzas Armadas ESPE,

2016.

- [12] C. L. Breazeal, *Designing Sociable Robots*. London: Library of Congress Cataloging-in-Publication Data, 2002.
- [13] C. Bartneck and J. Forlizzi, “A design-centered framework for social human-robot interaction,” in *International Workshop on Robot and Human Interactive Communication*, 2004, pp. 591–594.
- [14] F. Hegel, C. Muhl, B. Wrede, M. Hielscher-Fastabend, and G. Sagerer, “Understanding Social Robots,” in *2009 Second International Conferences on Advances in Computer-Human Interactions*, 2009, pp. 169–174.
- [15] H. Yan, M. H. Ang, and A. N. Poo, “A Survey on Perception Methods for Human-Robot Interaction in Social Robots,” *International Journal of Social Robotics*, vol. 6, no. 1. pp. 85–119, 2014.
- [16] MIT Media Lab, “Nexi and iCub.” [Online]. Available: <http://robotic.media.mit.edu/>.
- [17] A. Zarakı *et al.*, “Design and Evaluation of a Unique Social Perception System for Human-Robot Interaction,” *IEEE Trans. Cogn. Dev. Syst.*, vol. 8920, no. JULY, pp. 1–1, 2016.
- [18] R. Ballestriero, “Desde la contorsión de la realidad a lo siniestro: el incómodo hiperrealismo de maniqués, muñecas, efigies y figuras de cera,” *Brum. Res. J. Fantast.*, vol. IV, pp. 93–115, 2016.
- [19] D. A. Bautista Loza and S. Martini Popoli, “¿ Qué hay más allá del Valle Inquietante ?,” pp. 1–7, 2014.
- [20] ATR, “Telenoid.” [Online]. Available: <http://www.geminoid.jp/projects/kibans/Telenoid-overview.html>.
- [21] P. Encalada, “Control de Voz y Expresiones Gestuales en un Robot Interactivo,” Universidad Politécnica de Madrid, 2016.
- [22] Primer Fotón, “Pareidolia.” [Online]. Available: <https://primerfoton.wordpress.com/2015/08/11/pareidolia-viendo-caras-donde-no-las-hay/>.
- [23] Roemi E. Fernández Saavedra Héctor Montes Franceschi, Ed., *RoboCity16 Open Conference on Future Trends in Robotics*, no. May. 2016.
- [24] David César Loza Matovelle, “DISEÑO Y CONSTRUCCIÓN DE UNA CABEZA MECATRÓNICA DE ASPECTO REALISTA,” Universidad de

Valladolid, 2013.

- [25] K. Warwick, *Artificial intelligence the basics*. UK: University of Reading: Routledge, 2012.
- [26] P. Ponce, *Inteligencia artificial con aplicaciones a la ingeniería*, Primera Ed. México: Alfaomega Grupo Editor, 2010.
- [27] S. Sumathi, L. Ashok Kumar, and P. Surekha, *Computational Intelligence Paradigms for Optimization Problems Using MATLAB/SIMULINK*, 1st ed. CRC Press Taylor & Francis Group, 2016.
- [28] R. Benítez, G. Escudero, and S. Kannan, *Inteligencia Artificial Avanzada*, 1st ed. Barcelona: Editorial Universidad Oberta de Catalunya (UOC), 2014.
- [29] S. Khan, H. Rahmani, S. Afaq, A. Shah, and M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*, Lecture #1. Morgan & Claypool Publishers, 2018.
- [30] M. Lopez and W. Yu Liu, “Identificación de sistemas no lineales mediante redes neuronales convolucionales,” Centro de investigación y de estudios avanzados del Instituto Politécnico Nacional, 2017.
- [31] UFLDL, “UFLDL tutorial,” 2017. [Online]. Available: <http://cs231n.stanford.edu/>.
- [32] Dassault Systèmes SolidWorks, “SolidWorks,” 2017. .
- [33] Lynxmotion, “SSC-32 Servo Controller.” [Online]. Available: <http://www.lynxmotion.com/p-395-ssc-32-servo-contr>.
- [34] Spyder, “Spyder IDE.” [Online]. Available: <https://spyder-ide.github.io/>.
- [35] Project\_Jupyter, “Jupyter notebook.” [Online]. Available: <http://jupyter.org/>.
- [36] K. Choi, G. Fazekas, and M. Sandler, “Automatic tagging using deep convolutional neural networks,” *arXiv*, 2016.
- [37] W. McKinney, *Python for Data Analysis [datasci][pandas][python]*, 2nd Editio. O’Relly Media, 2012.
- [38] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2012.
- [39] G. Langevin, “InMoov Open Source 3D printed life-size robot.,” 2012. .
- [40] E. Barrero Parra, “Aceleración del algoritmo de Viola-Jones mediante rejillas de procesamiento masivamente paralelo en el plano focal,” p. 68, 2015.
- [41] OpenCV, “Face recognition.” [Online]. Available:

[https://docs.opencv.org/2.4.4-beta/doc/tutorials/introduction/desktop\\_java/java\\_dev\\_intro.html](https://docs.opencv.org/2.4.4-beta/doc/tutorials/introduction/desktop_java/java_dev_intro.html).

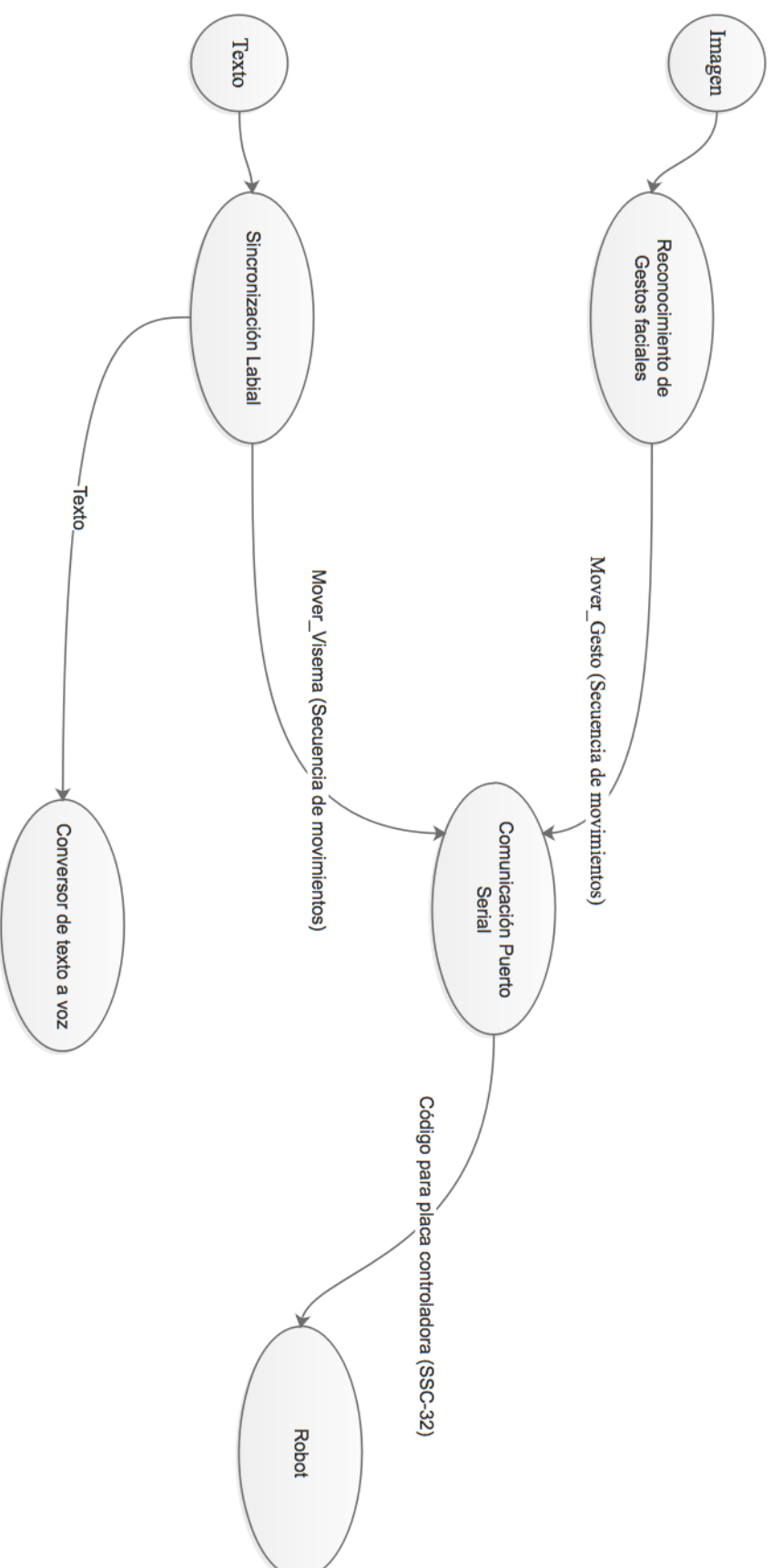
- [42] A. Miguel and D. Rodríguez, “Extracción automática de caras en imágenes captadas con móviles Android,” 2012.
- [43] S. Jena, “Machine Learning Engineer Nanodegree Capstone Project Emotion Recognition using Deep Learning,” no. May, 2017.
- [44] Kaggle, “Challenges in Representation Learning: Facial Expression Recognition Challenge.” [Online]. Available: <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>.
- [45] T. H. Vu, L. Nguyen, T. Guo, and V. Monga, “Deep Network for Simultaneous Decomposition and Classification in UWB-SAR Imagery.”
- [46] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2014.
- [47] S. Mohseni, N. Zarei, and S. Ramazani, “Facial expression recognition using anatomy based facial graph,” *2014 IEEE Int. Conf. Syst. Man, Cybern.*, pp. 3715–3719, 2014.



## **ANEXOS**

## ANEXO A

Diagrama nodal de funcionamiento del robot social interactivo.



## ANEXO B

### Posición de servomotores – gestos faciales

	Servo 0	Servo 1	Servo 2	Servo 3	Servo 4	Servo 5	Servo 6	Servo 7	Servo 8	Servo 9	Servo 10	Servo 11	Servo 12
<b>Nombre</b>	Lower lip	Upper lip	Right upper lip	Right lower lip	Left upper lip	Left lower lip	Mouth	R-L eyes	U-D eyes	Right eyelip	Left eyelip	Rigth eyebrow	Left eyebrow
<b>Gesto “Neutral”</b>													
<b>Posición</b>	1700	1200	1300	1400	1700	1600	2000	1400	1500	1600	1450	1600	1300
<b>Gesto “Sorpresa”</b>													
<b>Posición</b>	1600	1300	1200	1500	1800	1500	1500	1400	1500	1450	1570	1350	1550
<b>Gesto “Felicidad”</b>													
<b>Posición</b>	1400	1600	1000	1400	2000	1500	2000	1400	1500	1550	1490	1490	1410
<b>Gesto “Enojo”</b>													
<b>Posición</b>	1500	1500	1200	1500	1800	1500	2000	1400	1500	1540	1500	1800	1100
<b>Gesto “Miedo”</b>													
<b>Posición</b>	1600	1300	1300	1700	1700	1200	2000	1400	1500	1490	1540	1450	1500
<b>Gesto “Tristeza”</b>													
<b>Posición</b>	1500	1000	1500	1700	1500	1200	2000	1400	1750	1750	1340	1350	1600

## ANEXO C

### Codificación de Visemas y sílabas

Numero	Código de Visema	Visema
1	V01	AA
2	V02	AB
3	V03	BA
4	V04	BE
5	V05	BEA
6	V06	BI
7	V07	BIA
8	V08	BIE
9	V09	BEM
10	V10	BO
11	V11	BOA
12	V12	BOE
13	V13	BOI
14	V14	BU
15	V15	BUE
16	V16	BUI
17	V17	BUIA
18	V18	EA
19	V19	EH
20	V20	EM
21	V21	IA
22	V22	IE
22	V22	IM
24	V24	IU
25	V25	IY
26	V26	OA

Numero	Código de Visema	Visema
27	V27	OE
28	V28	OI
29	V29	OIA
30	V30	OB
31	V31	OH
32	V32	UA
33	V33	UE
34	V34	UI
35	V35	UIA
36	V36	UM
37	V37	BUO
38	V38	UH
39	V39	BIO
40	V40	BUI
41	V41	IO
42	V42	AE
43	V43	AI
44	V44	AO
45	V45	AU
46	V46	EI
47	V47	EO
48	V48	EU
49	V49	OU
50	V50	BEI
51	V51	BEO
52	V52	BEU

## ANEXO D

### Posición de los servomotores – visemas

	Servo 0	Servo 1	Servo 2	Servo 3	Servo 4	Servo 5
Nombre	Lower lip	Upper lip	Right upper lip	Right lower lip	Left upper lip	Left lower lip
Visema AA						
Posición	1300	1700	1100	1500	1900	1500
Visema EE						
Posición	1600	1400	1100	1500	1900	1500
Visema II						
Posición	1600	1400	1200	1400	1800	1600
Visema OO						
Posición	1700	1300	1500	1200	1400	1700
Visema UU						
Posición	1800	1200	1500	1200	1400	1800
Visema BFMPV						
Posición	1700	1200	1200	1700	1900	1300
Visema Neutro						
Posición	1700	1200	1300	1400	1700	1600

## ANEXO E

### Script de envío de datos a la tarjeta controladora.

```
import socket
import sys
import traceback
from threading import Thread
import serial

ser = serial.Serial('/dev/tty.usbserial', 115200, timeout = 1)

def Main():
    start_server()

def start_server():
    host = "127.0.0.1"
    port = 8888 # puerto arbitrario no privilegiado
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)

    print("Socket creado")

    try:
        soc.bind((host, port))
    except:
        print("La unión falló. Error: " + str(sys.exc_info()))
        sys.exit()

    soc.listen(5) # hacer cola hasta 5 solicitudes
    print("Socket ahora escuchando")

    # ciclo infinito: no restablecer para cada solicitud
    while True:
        connection, address = soc.accept()
        ip, port = str(address[0]), str(address[1])

        print("Conectado a " + ip + ":" + port)

        try:
            Thread(target=client_thread, args=(connection, ip, port)).start()
        except:
            print("El hilo no comenzó.")

            traceback.print_exc()

    soc.close()

def client_thread(connection, ip, port, max_buffer_size = 5120):
    is_active = True

    while is_active:
        client_input = receive_input(connection, max_buffer_size)

        if "--QUIT--" in client_input:
```

```

        print("El cliente está solicitando salir")
        connection.close()

        print("Conexión " + ip + ":" + port + " cerrado")
        is_active = False

    else:
        print("Resultado procesado: {}".format(client_input))
        connection.sendall("-".encode("utf8"))

def receive_input(connection, max_buffer_size):

    client_input = connection.recv(max_buffer_size)
    client_input_size = sys.getsizeof(client_input)

    if client_input_size > max_buffer_size:

        print("El tamaño de entrada es mayor de lo esperado
        {}".format(client_input_size))

        decoded_input = client_input.decode("utf8").rstrip()

        result = process_input(decoded_input)

        return result

def process_input(input_str):

    print("Procesando la entrada recibida del cliente")
    input_ = input_str + "\r"
    ser.write(input_.encode())

    return "HOLA " + str(input_str).upper()

if __name__ == "__main__":
    Main()

```

## ANEXO F

### Script de reconocimiento de gestos faciales.

```
import socket
import sys
import cv2
import time
import numpy as np
from keras.models import load_model

global emocion_servo, dato_enviar

def movimientos(result):

    neutral_gest = '#16 P1700 #17 P1200 #20 P1300 #21 P1400 #22 P1700 #23
P1600 #24 P2000 #0 P1400 #1 P1500 #2 P1600 #3 P1450 #4 P1600 #5 P1300 T200\r'
    surprised_gest= '#16 P1600 #17 P1300 #20 P1200 #21 P1500 #22 P1800 #23
P1500 #24 P1500 #0 P1400 #1 P1500 #2 P1450 #3 P1570 #4 P1350 #5 P1550 T200\r'
    happy_gest = '#16 P1400 #17 P1600 #20 P1000 #21 P1400 #22 P2000 #23
P1500 #24 P2000 #0 P1400 #1 P1500 #2 P1550 #3 P1490 #4 P1490 #5 P1410 T200\r'
    angry_gest = '#16 P1500 #17 P1500 #20 P1200 #21 P1500 #22 P1800 #23
P1500 #24 P2000 #0 P1400 #1 P1500 #2 P1540 #3 P1500 #4 P1800 #5 P1100 T200\r'
    fear_gest = '#16 P1600 #17 P1300 #20 P1300 #21 P1700 #22 P1700 #23
P1200 #24 P2000 #0 P1400 #1 P1500 #2 P1490 #3 P1540 #4 P1450 #5 P1500 T200\r'
    sad_gest = '#16 P1500 #17 P1000 #20 P1500 #21 P1700 #22 P1500 #23
P1200 #24 P2000 #0 P1400 #1 P1750 #2 P1750 #3 P1340 #4 P1350 #5 P1600 T200\r'

    if result == 'Neutral': emocion_servo = neutral_gest

    elif result == 'Sorpresa':emocion_servo = surprised_gest
    elif result == 'Felicidad':emocion_servo = happy_gest
    elif result == 'Enojo':emocion_servo = angry_gest
    elif result == 'Miedo':emocion_servo = fear_gest
    elif result == 'Tristeza':emocion_servo = sad_gest
    elif result == 'surprise':emocion_servo = surprised_gest

    return emocion_servo

def Main():

    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = "127.0.0.1"
    port = 8888

    try:
        soc.connect((host, port))
    except:
        print("Connection error")
        sys.exit()

    W_img = 480
    H_img = 320

    # cargamos la plantilla e inicializamos la webcam:
    face_cascade =
cv2.CascadeClassifier("/Users/edissonarias/Desktop/haarcascade_frontalface_defau
lt.xml")
    cap = cv2.VideoCapture(1)

    dato = 1500
    dato_1 = 1200

    # lectura del modelo entrenado de la red neuronal
```



```

model = load_model('/Users/edissonarias/Desktop/model_5-49-0.62.hdf5')
etiqueta =
['Enojo', 'Disgusto', 'Miedo', 'Felicidad', 'Tristeza', 'Sorpresa', 'Neutral']

# funcion que permite dibujar las etiquetas.
font = cv2.FONT_HERSHEY_SIMPLEX

while True:
    # leemos un frame y lo guardamos
    ret, img = cap.read()

    # reducimos el tamaño de la imagen
    resize = cv2.resize(img, (W_img, H_img), interpolation =
cv2.INTER_LINEAR)

    # numero de las columnas y filas de la imagen
    num_rows, num_cols = resize.shape[:2]

    # matrix de rotación a 180°
rotation_matrix = cv2.getRotationMatrix2D((num_cols/2, num_rows/2), 180,
0.8)

    # acople de imagen girada
img_rotation = cv2.warpAffine(resize, rotation_matrix, (num_cols,
num_rows))

    # tratamiento imagen escala de grises
gray = cv2.cvtColor(img_rotation, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(gray, 1.3, 5)

    # Dibujamos un rectangulo en las coordenadas de cada rostro
    for (x,y,w,h) in faces:
        cv2.rectangle(img_rotation, (x,y), (x+w,y+h), (255, 255, 0), 2, 1)
        face_crop = img_rotation[y:y+h,x:x+w]
        face_crop = cv2.resize(face_crop, (48,48))
        face_crop = cv2.cvtColor(face_crop, cv2.COLOR_BGR2GRAY)
        face_crop = face_crop.astype('float32')/255
        face_crop = np.asarray(face_crop)
        face_crop = face_crop.reshape(1,
1, face_crop.shape[0], face_crop.shape[1])
        result = etiqueta[np.argmax(model.predict(face_crop))]
        cv2.putText(img_rotation, result, (x,y-10), font, 0.9, (255,255,255),
1, cv2.LINE_AA)

        movimiento = movimientos(result)
        dato_enviar = movimiento
        #print(dato_enviar)
        message = dato_enviar
        soc.send(message.encode())
        time.sleep(0.05)
        print("Gesto detectado: ", result,)
        print("Bloque de Movimientos", movimiento)

    # Display the resulting frame
    cv2.imshow('Video', img_rotation)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        soc.send(b'--quit--')
        cap.release()
        cv2.destroyAllWindows()
        break

if __name__ == "__main__":
    Main()

```

## ANEXO G

### Script de sincronización labial.

```
import socket
import sys
from lxml import etree
from Utils import D_Silabas
from Utils import Utils
import time
import os
import Texto_a_Voz
import speech_recognition as sr

##### Reconocimiento de voz #####
def reconocimiento():
    global text_rec
    # Record Audio
    r = sr.Recognizer()
    with sr.Microphone() as source:
        print("Say something!")
        audio = r.listen(source)
    try:
        texto_rec = r.recognize_google(audio, language = "es-ES")
    except sr.UnknownValueError:
        texto_rec = "."
    except sr.RequestError as e:
        texto_rec = "Prueba diciendo otras cosas"

    return texto_rec

global Movimiento_Salida,cadena

def movimiento_servomotores(Visema_leido):
    # visemas      Lowerlip  Upperlip  r_lid-up  r_lid_down
    l_lid_up  l_lid_down mouth
    Visema_Neutro= '#16 P1700 #17 P1200 #20 P1300 #21 P1400 #22 P1700 #23
P1600 #24 P2000 T200\r'
    Visema_A_A = '#16 P1300 #17 P1700 #20 P1100 #21 P1500 #22 P1900 #23
P1500 #24 P1500 T200\r'
    Visema_E_E = '#16 P1600 #17 P1400 #20 P1100 #21 P1500 #22 P1900 #23
P1500 #24 P1700 T200\r'
    Visema_I_I = '#16 P1600 #17 P1400 #20 P1200 #21 P1400 #22 P1800 #23
P1600 #24 P1900 T200\r'
    Visema_O_O = '#16 P1700 #17 P1300 #20 P1500 #21 P1200 #22 P1400 #23
P1700 #24 P1600 T200\r'
    Visema_U_U = '#16 P1800 #17 P1200 #20 P1500 #21 P1200 #22 P1400 #23
P1800 #24 P1800 T200\r'
    Visema_BMPV= '#16 P1700 #17 P1200 #20 P1200 #21 P1700 #22 P1900 #23
P1300 #24 P2000 T200\r'

    if Visema_leido == 'N':
        Movimiento_Salida = Visema_Neutro

    elif Visema_leido == 'A':
        Movimiento_Salida = Visema_A_A

    elif Visema_leido == 'E':
        Movimiento_Salida = Visema_E_E

    elif Visema_leido == 'I':
        Movimiento_Salida = Visema_I_I

    elif Visema_leido == 'O':
        Movimiento_Salida = Visema_O_O
```

```

elif Visema_leido == 'U':
    Movimiento_Salida = Visema_U_U

elif Visema_leido == 'BM':
    Movimiento_Salida = Visema_BMPV

return Movimiento_Salida

def Main():
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    host = "127.0.0.1"
    port = 8888

    try:
        soc.connect((host, port))
    except:
        print("Connection error")
        sys.exit()

    while True:

        # Definición de vectores
        vector_visemas = []
        vector=[]
        vector_cadena=[]
        palabras = ""

        # Lectura de base de datos Visemas
        doc = etree.parse('Codigo de visemas.xml')
        raiz=doc.getroot()
        # Ingreso de datos
        cadena = reconocimiento()
        #Texto_a_Voz.Speak(cadena)

        # Eliminar signos de puntuación
        texto = D_Silabas.remove_punctuation(cadena)
        print(texto)
        palabras += texto

        # Asignar silabas mediante , al arreglo vector
        vector = D_Silabas.silabas(palabras.replace(' ', ''))
        print(vector)

        # Crear vector con silabas en mayúsculas para comparación
        for x in range(len(vector)):
            newdata = vector[x].strip()
            vector_cadena.append(newdata.upper())
        print(vector_cadena)

        vector_de_visemas = Utils.Get_Visemes_vector(vector_cadena)
        print(vector_de_visemas)
        valor_mover = Utils.Movimientos(vector_de_visemas)
        print(valor_mover)

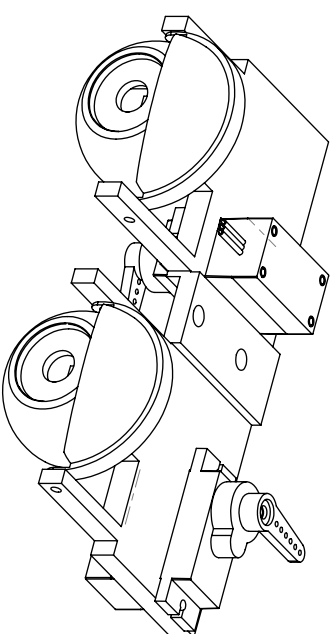
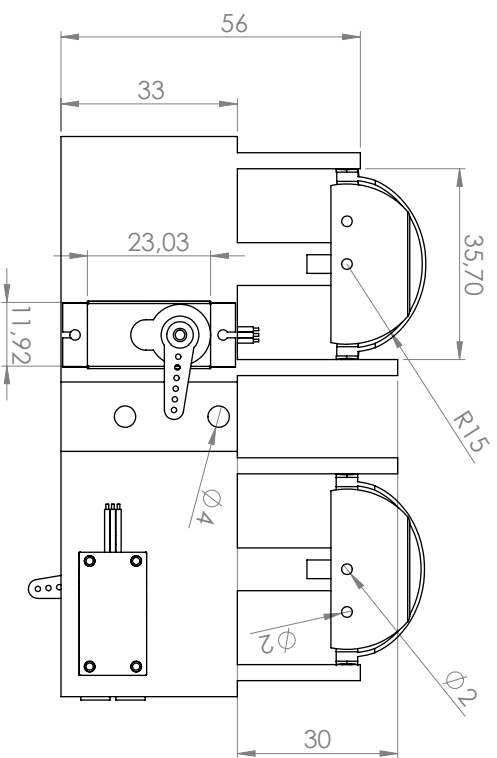
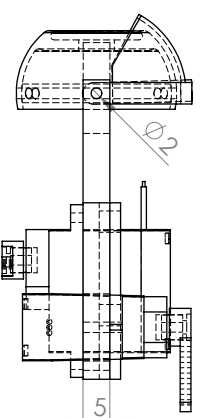
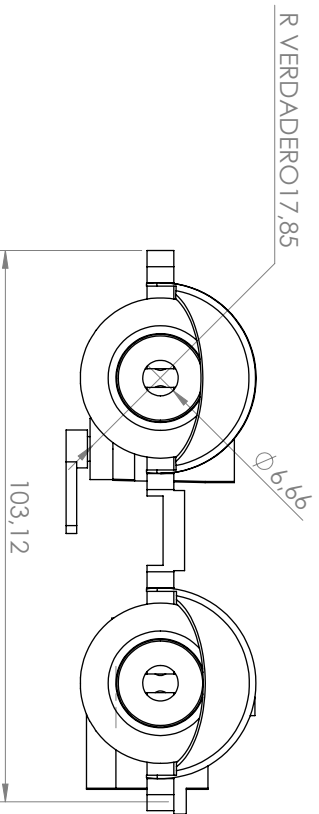
        for a in range(len(valor_mover)):
            Visema_leido = valor_mover[a]
            mover_servomotores = movimiento_servomotores(Visema_leido)
            print(mover_servomotores)
            message = mover_servomotores
            soc.send(message.encode())
            time.sleep(0.1)

if __name__ == "__main__":
    Main()

```

## **ANEXO H**

### **Planos**



TOLERANCIAS: 0.01 mm		ACABADO:		NO CAMBIAR LA ESCALA		REVISION	01
LINEAL:		Impresión 3D - 0.2 mm					
ANGULAR:							

DIBUJ	NOBRE	FIRMA	FECHA	TITULO:	MECANISMO PARA MOVIMIENTO DE OJOS - ROBOT SOCIAL
VERIF	Aitor Edisson		22-10-2017		
APROB	Ing. Encarnada P.		22-10-2017		
FABR					
CALID					

MATERIAL:		Nº DE DIBUJO	ESCALA: 1:1	HOJA 1 DE 1
PLA (ácido Poli-láctico)		01 - 2018		
PESO:				

