

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS ELECTRÓNICOS**

**INTERPRETACIÓN DE EXPRESIONES FACIALES EN ADULTOS
MAYORES UTILIZANDO LA VISIÓN ARTIFICIAL DEL ROBOT
HUMANOIDE NAO**

**AUTORES:
FRANCISCO RAFAEL CALVOPIÑA IGLESIAS
PEDRO ESTEBAN VALLADARES ROMERO**

**DIRECTOR:
VÍCTOR VINICIO TAPIA CALVOPIÑA**

Quito, febrero del 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros, Francisco Rafael Calvopiña Iglesias, con documento de identificación N°1721257879, y Pedro Esteban Valladares Romero, con documento de identificación N° 1722956644, manifiesto mi voluntad y cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: INTERPRETACIÓN DE EXPRESIONES FACIALES EN ADULTOS MAYORES UTILIZANDO LA VISIÓN ARTIFICIAL DEL ROBOT HUMANOIDE NAO, mismo que ha sido desarrollado para optar por el título de: Ingenieros Electrónicos, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en mi condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



.....
Francisco Rafael Calvopiña Iglesias
Cédula: 1721257879



.....
Pedro Esteban Valladares Romero
Cédula: 1722956644

Febrero 2017

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación INTERPRETACIÓN DE EXPRESIONES FACIALES EN ADULTOS MAYORES UTILIZANDO LA VISIÓN ARTIFICIAL DEL ROBOT HUMANOIDE NAO realizado por Francisco Rafael Calvopiña Iglesias y Pedro Esteban Valladares Romero, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, febrero 2017



Víctor Vinicio Tapia Calvopiña

CI: 1708547219

DEDICATORIA

Dedico la realización de este proyecto a mi padre, por su apoyo en todo momento de mi carrera, por haberme enseñado sobre la responsabilidad y los valores para guiar mi vida. A mi madre, por su amor y alegría con los que hace ver a los problemas pequeños. A mis hermanos, que con su ejemplo me han guiado a crear mi propio camino. A Dios, por esta familia que me apoya a alcanzar mis metas.

Francisco Rafael Calvopiña Iglesias

El presente proyecto de titulación le dedico a toda mi familia, en especial a mis padres Mariana y Jhon que fueron la motivación principal desde los inicios de mi vida académica y un recordatorio constante de que cada esfuerzo tiene su recompensa para así poder culminar mis estudios universitarios y no desistir.

Pedro Esteban Valladares Romero

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana por la formación académica y humana recibida, a cada uno de los docentes que nos brindaron sus conocimientos, paciencia y respeto. Al ingeniero Vinicio Tapia por su calidad humana y por el tiempo dedicado para el desarrollo del proyecto.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1	2
PLANTEAMIENTO DEL PROBLEMA	2
1.1 Planteamiento del Problema.....	2
1.2 Justificación.....	2
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos.....	3
1.4 Alcance.....	4
CAPÍTULO 2	5
MARCO TEÓRICO.....	5
2.1 Robótica	5
2.2 Clasificación de los Robots.....	6
2.3 Robots Humanoides	6
2.4 Clasificación de los Robots Humanoides.....	8
2.4.1 Robot semi-humanoides.....	8
2.4.2 Humanoides de tamaño pequeño	9
2.4.3 Robot Humanoides de tamaño completo	10
2.5 Robot NAO	10
2.5.1 Conectividad	13
2.5.2 Interacción.....	14
2.5.3 Sensores FSR	14
2.5.4 Unidad de Inercia	14
2.5.5 Sonares	14
2.5.6 Sensores de posición	15

2.5.7 Sensores táctiles	15
2.5.8 NAOqi Framework (O.S).....	15
2.5.9 Plataforma cruzada.....	16
2.5.10 Lenguaje cruzado	16
2.5.11 Aplicaciones distribuidas	16
2.5.12 Introspección	16
2.5.13 Broker.....	17
2.5.14 Proxy	17
2.5.15 Módulos.....	17
2.5.16 Proceso de NAOqi.....	17
2.6 Programación	18
2.6.1 Choregraphe	19
2.6.2 Python 2.7 SDK e IDLE.....	19
2.6.3 NAOqi Python API	20
2.6.4 wxPython.....	21
2.7 Cinemática del Robot NAO	21
2.7.1 Calculo de los grados de libertad con criterio de Kutzbach-Grübler	22
2.7.2 Cinemática Directa del robot por Denavit-Haartenberg NAO.....	24
2.8 Reconocimiento de Expresiones Faciales	26
2.9 Necesidades del Adulto Mayor	27
2.10 Estado del Arte.....	29
CAPÍTULO 3	31
DESARROLLO E IMPLEMENTACIÓN.....	31
3.1 Módulos de NAOqi.....	31
3.1.1 Síntesis de Voz.....	31
3.1.2 Correr un Behavior.....	33
3.1.3 Creación de Movimientos	35

3.1.4 Reconocimiento de Expresión Facial.....	37
3.1.5 Guardar y Reconocer Rostros en el Robot.....	38
3.1.6 Reconocimiento de Voz.....	40
3.1.7 Sensores de Tacto.....	42
3.1.8 Rastreo de Rostros.....	43
3.1.9 Obtener Fecha.....	44
3.1.10 Enviar e-mail.....	45
3.1.11 Guardar Datos.....	46
3.1.12 Leds.....	47
3.1.13 Reconocimiento de Objetos.....	47
3.1.14 Obtener la Imagen de la Cámara.....	48
3.2 Planteamiento de la Aplicación.....	50
3.3 Programación de la Aplicación.....	50
3.3.1 Presentación Inicial.....	51
3.3.2 Iniciación de la Rutina.....	51
3.3.3 Introducción Saludo.....	51
3.3.4 Clasificación de las Actividades según las Expresiones Faciales.....	51
3.3.5 Actividad Chistes.....	52
3.3.6 Actividad Frases.....	52
3.3.7 Actividad Reproducción de Música.....	53
3.3.8 Actividad Historias.....	53
3.3.9 Actividad Bailes.....	53
3.3.10 Actividad Reconocimiento de Figuras.....	53
3.3.11 Actividad Ejercicios.....	54
3.3.12 Notificaciones y Guardar Datos.....	54
CAPÍTULO 4.....	56
ANÁLISIS DE RESULTADOS.....	56

4.1 Pruebas y análisis en el reconocimiento de expresiones faciales.....	56
4.2 Pruebas y análisis del reconocimiento de voz.....	57
4.3 Prueba de la aplicación.....	59
CONCLUSIONES	61
RECOMENDACIONES	63
LISTA DE REFERENCIAS	64
ANEXOS.....	1

ÍNDICE DE FIGURAS

Figura 2.1. Primer robot controlado por un microprocesador.....	5
Figura 2.2. Robot ASIMO de honda	7
Figura 2.3. ENON. Fuente: (thefutureofthings, 2016)	8
Figura 2.4. Rollin.	8
Figura 2.5. Pepper	8
Figura 2.6. ASIMO	9
Figura 2.7. SDR.....	9
Figura 2.8. Hoap.....	9
Figura 2.9. Romeo.....	10
Figura 2.10. Topsy	10
Figura 2.11. Versión del robot NAO T2, T14 y H25.	11
Figura 2.12. Dimensiones del robot humanoide NAO V5 H25.	13
Figura 2.13. Puerto Ethernet y USB, detrás de la cabeza del robot NAO.	13
Figura 2.14. Proceso de ejecución de NAOqi.	18
Figura 2.15. Ambiente de programación multiplataforma del robot NAO.	19
Figura 2.16. Scripts de Python, en Choregraphe e IDLE,	20
Figura 2.17. Ambiente grafico del programa wxFormBuilder.	21
Figura 2.18. Junturas del robot NAO	22
Figura 2.19. Estudio de los grados de libertad	23
Figura 2.20. Pierna izquierda cálculos de cinemática.	24
Figura 2.21. Respuesta inicial de los niños con autismo en una terapia con el robot NAO.	29
Figura 2.22. Robot NAO en el tratamiento de niños con diabetes.....	30
Figura 2.23. Robot Nao juego de enseñanza de geografía	30
Figura 3.1. Configuración para introducir texto a un bloque “Say Text”.....	32
Figura 3.2. Programación necesaria para conectarse al módulo de síntesis de voz. ..	33
Figura 3.3. Configuración para funcionamiento del bloque “Run Behavior”.....	33
Figura 3.4. Conexión con el módulo, consulta y correr un behavior.	34
Figura 3.5. Programación de movimientos en una línea de tiempo de Choregraphe. 36	
Figura 3.6. Manejo de movimientos con bezier.	37
Figura 3.7. Función obtener la expresión facial.....	38
Figura 3.8. Comparación del número mayor entre cinco números.	38

Figura 3.9. Esta función contiene toda la información del rostro de la persona.	39
Figura 3.10. Información esencial de los rostros.	39
Figura 3.11. Este comando suscribe a la interrupción.	41
Figura 3.12. Clasifica las acciones según la palabra reconocida.	42
Figura 3.13. Suscribirse al sensor MiddleTactilTouched y función asociada.....	43
Figura 3.14. Inicio y conexión con el modulo.	44
Figura 3.15. Programación en comandos en Python para obtener la fecha.	45
Figura 3.16. Programación para enviar un mail por SMTP.	46
Figura 3.17. Guardar un diccionario y una lista.....	46
Figura 3.18. Extraer el diccionario y la lista.	47
Figura 3.19. Programación del módulo ALLeds.....	47
Figura 3.20. Bloque de reconocimiento de imágenes, objetos y lugares.	48
Figura 3.21. Configuraciones iniciales y conexión al módulo.....	49
Figura 3.22. Función para obtener una imagen.....	50
Figura 3.23. Imágenes correctas para el reconocimiento.	54
Figura 4.1. Prueba de distancias óptimas para comandos de voz.	58
Figura 4.2. Prueba de comandos de voz en el piso y sobre una tarima.....	58
Figura 4.3. Prueba de guardar rostros en interfaz WxPython.	59
Figura 4.4. Envío de un correo electrónico con los datos de la aplicación.	60

ÍNDICE DE TABLAS

Tabla 2.1. Clasificación general de los robots	6
Tabla 2.2. Ejemplos de robots semi- humanoides.....	8
Tabla 2.3. Ejemplos de robots humanoides de tamaño pequeño	9
Tabla 2.4. Ejemplos de robots humanoides de tamaño completo	10
Tabla 2.5. Especificaciones Técnicas del Robot NAO V5	12
Tabla 2.6. Sensores del robot NAO	15
Tabla 2.7. Lenguajes de programación	18
Tabla 2.8. Parámetros de Denavit-Haartenberg	25
Tabla 3.1. Comandos de entonación	32
Tabla 3.2. Métodos del módulo behaviors	34
Tabla 3.3. Métodos del módulo detección de rostros.....	40
Tabla 3.4. Métodos del módulo reconocimiento de voz	41
Tabla 3.5. Nombres de los sensores	42
Tabla 3.6. Métodos del módulo rastreo de rostros	44
Tabla 3.7. Comandos para obtener la fecha	45
Tabla 3.8. Resoluciones, colores y frames por segundo	49
Tabla 3.9. Clasificación de las actividades	52
Tabla 4.1. Análisis de las expresiones faciales	57
Tabla 4.2. Tabla de porcentaje del Error Promedio	59

RESUMEN

El proyecto trata del desarrollo de una aplicación de interacción humano robot entre un adulto mayor y el robot humanoide NAO perteneciente a los laboratorios de la Universidad Politécnica Salesiana, con la finalidad de que reconozca expresiones faciales y de acuerdo a estas programar actividades con rutinas de movimientos y diálogos básicos con el adulto mayor.

La aplicación inicia con una interfaz gráfica desarrollada en wxPython en la cual se almacena el rostro y se introduce sus datos personales, para luego continuar con el reconocimiento de expresiones faciales y la clasificación de las actividades que se desarrolló en Python.

Para el diseño de las actividades se investigó las necesidades del adulto mayor, y se analizó las terapias que se pueden realizar con el robot, como las terapias de orientación a la realidad, reminiscencia y lúdicas que se desarrollaron con librerías de visión artificial y reconocimiento de voz.

Las actividades que se desarrollaron en Python son canciones, historias, ejercicios, frases y chistes, pero también se dio uso a Choregraphe en la actividad de reconocer figuras, movimientos y bailes.

Al final de cada sesión de asistencia el robot NAO envía los datos personales, expresiones faciales y las actividades desarrolladas entre el robot y el adulto mayor vía correo electrónico a una persona encargada del adulto mayor y de forma local al tocar la cabeza del robot.

ABSTRACT

The project deals with the development of a robot human interaction application between an older adult and the humanoid robot NAO belonging to the laboratories of the Salesian Polytechnic University, in order to recognize facial expressions and according to these program activities of movement routines and basic dialogues with the adult higher.

The application starts with a graphical interface developed in wxPython where a face is saved and enter personal data, then continue with the recognition of facial expressions and the classification of activities that was developed in Python.

For the design of the activities, the needs of the elderly were investigated, and the therapies that could be performed with the robot were analyzed, such as reality-oriented, reminiscence and playful therapies that were developed with artificial vision libraries and recognition of voice.

The activities that were developed in Python are songs, stories, exercises, phrases and jokes, but Choregraphe was also used in the activity of recognizing figures, movements and dances.

At the end of each attendance session the robot NAO send personal data, facial expressions and activities developed between the robot and the older adult via e-mail to a person in charge of the older adult and locally by touching the robot's head

INTRODUCCIÓN

Actualmente el desarrollo de la robótica con respecto a los robots humanoides ha logrado alcanzar una interacción suficientemente natural entre robots y usuarios, que permite la exploración de nuevos campos de aplicación, como los fines médicos de rehabilitación, tratamientos de enfermedades y de educación, aunque resulta complicado, se han obtenido buenos resultados.

Son más de 30 años de investigación dedicados a los robots humanoides que van desde la locomoción bípeda, algoritmos inteligentes, mecanismos, visión artificial, síntesis de voz, y una cantidad de innumerables aportes que ha permitido lograr la interacción actual.

En este proyecto nos hemos enfocado en el adulto mayor, al cual se lo define como una persona adulta cuya edad supera los 65 años, los cuales por haber terminado su ciclo de vida laboral, es decir se jubilan tanto por edad, por algún padecimiento en su salud o accidentes en su mismo entorno laboral y que debido al ritmo de vida acelerado que exige la sociedad, sus familiares no pueden estar pendientes del adulto, siendo la soledad y falta de atención un problema mayúsculo en el adulto mayor, este desencadena factores psicológicos depresión, demencia, etc. y a su vez problemas físicos debido a que optan por un estado sedentario que va a terminar acelerando el proceso de envejecimiento.

Los robots humanoides como el robot NAO de Aldebaran, son una base para desarrollar aplicaciones y conociendo los problemas que aquejan al adulto mayor se pretende utilizar la robótica para la asistencia del mismo, como parte de una solución a sus problemas. El proyecto se ha basado en el desarrollo de un sistema, que interprete las expresiones faciales del adulto mayor antes y después de una actividad como juegos, canciones, bailes e interacciones de formas verbales.

CAPÍTULO 1

PLANTEAMIENTO DEL PROBLEMA

1.1 Planteamiento del Problema

En esta época se vive aceleradamente y no disponemos de tiempo para compartir con los miembros de la familia. El adulto mayor es una parte fundamental en el desarrollo de las sociedades ya que por su edad han ido recogiendo vasta experiencia siendo considerados como sabios, pero el tiempo ha ido cambiando y el problema ahora es que son considerados una carga por sus propios familiares.

Los adultos mayores son una población diversa y activa, pero que no cuentan con un rol específico en la sociedad, debido a que la mayoría los miran como personas improductivas y frágiles por su edad, teniendo como resultado que este grupo sea aislado y llevado a la inactividad, viéndose así afectada su calidad de vida y deteriorando su salud.

Siendo así que los adultos mayores y personas de avanzada edad deben permanecer activas y recreándose, manteniendo principios sanos de salud física y psicológica, tal cual en la ciudad de Quito se cuenta con el Sistema integral de atención al adulto mayor, más conocido como Sesenta y Piquito el cual se enfoca en tres ejes de acción, pero el que nos interesa es el recreativo y el de emprender, en los cuales se realiza talleres para que el adulto mayor se entretenga, interactúe y desarrolle con más personas de la misma edad.

1.2 Justificación

La necesidad de asistencia al adulto mayor es de vital importancia para mantener su salud mental y física, por lo que se estima que en el 2050 la cantidad de adultos mayores se duplique a dos mil millones en el mundo.

La asistencia al adulto mayor tiene como condicionante la capacidad económica familiar, se realizan en casa o en centros especializados denominados geriátricos, aunque es una realidad que tanto en casa como en los centros especializados no se presta una asistencia óptima a las necesidades del adulto mayor, una muestra de esto es la alta tasa de suicidios, debido a que en casa los integrantes de la familia trabajan

todo el día, por lo cual los adultos mayores pasan mucho tiempo solos, pero los costos que implica brindar una asistencia óptima personalizada al adulto mayor no es alcanzable para la mayoría de familias, debido a esto se busca nuevas alternativas como el robot humanoide NAO que está en continuo desarrollo para mejorar sus características en la interacción con humanos.

En la actualidad se busca la aplicación de tecnologías innovadoras que estén orientadas a la ayuda social, debido a esto, este trabajo permitirá conocer y utilizar robots humanoides, contribuyendo al avance para aplicaciones futuras en el campo de la asistencia al adulto mayor, aprovechando el aspecto amigable del robot humanoide NAO, el cual brinda un sentimiento de comodidad a las personas que interactúan.

El robot NAO se puede ir adaptando a los intereses y necesidades de diversas personas, mediante el uso de software de alto nivel (Choregraphe y Python), se le puede programar para ejecutar diferentes tareas, tales como reconocimiento de expresiones faciales, conversaciones cortas y puede ser utilizado para cumplir diversas órdenes entre las cuales destacan realizar tareas funcionales, informar, educar, entretener y la más importante asistir a humanos y por ende se lo puede aplicar en adultos mayores, niños y personas discapacitadas. Todo esto se puede lograr con una adecuada programación e implementando las librerías provistas por el fabricante del robot.

1.3 Objetivos

1.3.1 Objetivo General

Reconocer expresiones faciales del adulto mayor, mediante la adaptación de librerías de visión artificial, programando rutinas de movimientos y diálogos básicos en el robot NAO en función de cada expresión facial reconocida.

1.3.2 Objetivos Específicos

- Investigar información de estudios y trabajos previos, donde se indique las necesidades del adulto mayor por déficit de atención, para un posterior análisis e interpretación.

- Adaptar las librerías genéricas de reconocimiento facial del robot NAO, mediante programación en Choregraphe y lenguaje Python, para reconocer expresiones en el rostro del adulto mayor como: triste, feliz, enojado, sorprendido y neutro (tranquilo).
- Programar rutinas de movimiento y diálogos básicos en el robot NAO, mediante el uso de Choregraphe y lenguaje Python, para cambiar el estado de ánimo del adulto mayor a un estado de ánimo positivo.
- Programar un algoritmo en el robot NAO, mediante el uso Choregraphe y lenguaje Python, que compruebe e informe si no hay un cambio en el estado de ánimo en adulto mayor después de interactuar con una rutina.
- Realizar diez pruebas de campo con adultos mayores, para comprobar la programación de reconocimiento de expresiones faciales y el cambio en el estado de ánimo del adulto mayor a través de la interacción con el robot NAO.
- Realizar un análisis con los resultados obtenidos de las pruebas de campo, para determinar la eficacia en el reconocimiento de expresiones faciales.

1.4 Alcance

El objetivo de este proyecto es realizar la programación del robot humanoide NAO, con lenguaje de alto nivel utilizando lenguaje Python y la herramienta de programación gráfica Choregraphe, donde el usuario podrá guardar su rostro desde una interfaz gráfica hecha en WX que es una extensión de Python, y el robot podrá reconocer expresiones faciales en el rostro del adulto mayor haciendo uso de la visión artificial del mismo. Una vez analizada la expresión el robot NAO procederá a hacer rutinas de ejercicios, bailes, tocar canciones del agrado del adulto mayor, reconocimiento de comandos de voz, contar historias y realizar preguntas para obtener un cambio en el estado de ánimo del adulto mayor.

CAPÍTULO 2

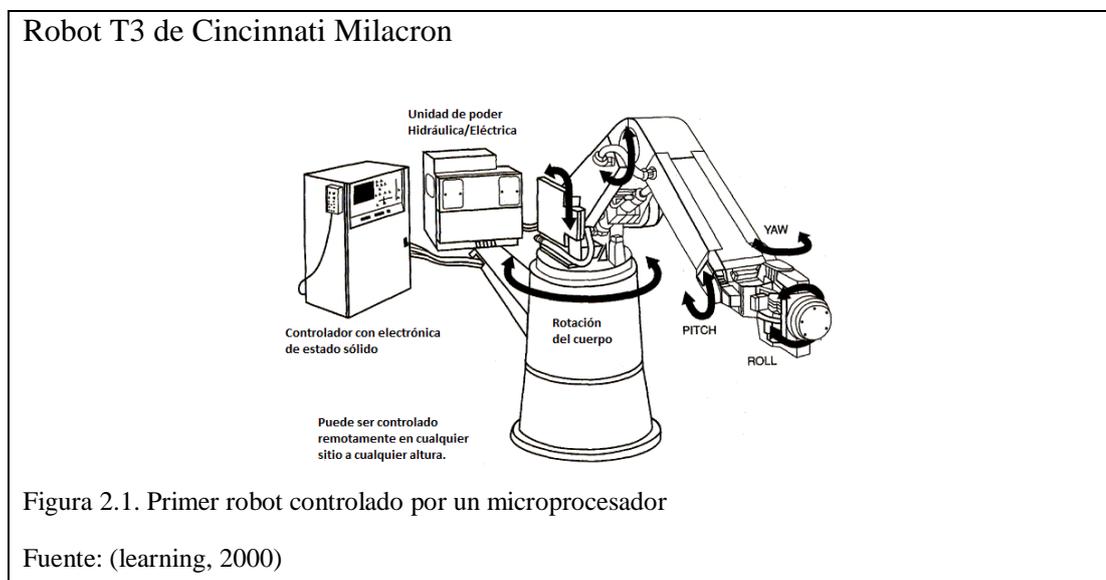
MARCO TEÓRICO

2.1 Robótica

La robótica es una ciencia de carácter interdisciplinario que abarca ciencias como la informática, la mecánica, la electrónica, ingeniería de control y otras, que se dedica al diseño y la construcción de robots.

La definición de robot difiere entre las grandes asociaciones de robótica japonesa, europea y norteamericana, debido a que se considera en Europa y Norteamérica a un robot como una máquina para realizar una tarea, pero en Japón lo toman como un compañero del humano. Por esta razón una definición que se adapta a la aplicación sería la siguiente: “Un robot se define básicamente como una máquina automática o autónoma que posee cierto grado de inteligencia, capaz de percibir su entorno y de reproducir ciertos comportamientos del ser humano”. (Riccillo, 2012)

El concepto de robot nació de la imaginación de varios literatos, que en sus obras plasmaban seres con vida artificial, aunque el término robot fue introducido en 1921 por el dramaturgo Karel Capek en su novela satírica Rossums Universal Robots, en la cual los robots reemplazan a los humanos en la ejecución de tareas sin descansar.



En especial la electrónica marco un nuevo ritmo de desarrollo en la robótica con la implementación de controladores y computadores, que llevaron a la creación del

primer robot industrial controlado con microrordenador The Tomorrow Tool o T3 por la empresa Cincinnati Milacron Figura 2.1. A partir de este momento conforme se aumentaba la capacidad de procesamiento o se reducía el tamaño de los componentes se lograban robots más sofisticados.

Los dos campos de aplicación de robots que se han desarrollaron son los industriales que son diseñados para dotar de flexibilidad y aumentar la productividad de un proceso y los robots de servicios en los que se encuentran los robots domésticos y robots asistentes.

2.2 Clasificación de los Robots

Existen diversos tipos de clasificaciones de robots según algunos criterios, sin embargo, tienen una clasificación general Tabla 2.1.

Tabla 2.1.

Clasificación general de los robots

Clasificación General de los Robots		
Móviles	Terrestres: ruedas, patas	
	Submarinos, aéreo - espaciales	
Humanoides	Diseño complejo	
Industriales	Brazos mecánicos	Robot Manipuladores

Nota: La clasificación general de los robots,

Fuente: (Cortés, 2011)

Aunque la clasificación más detallada se refiere al uso, al medio y a la inteligencia que posee un robot.

2.3 Robots Humanoides

Los robots humanoides son los que más llaman la atención en la sociedad debido a que su investigación y desarrollo parte de la idea de hacerlos amigables, simpáticos y con características humanas para aplicaciones de asistencia. Su apariencia física es similar a un humano tienen un torso, una cabeza, dos brazos, dos piernas o una base móvil,

aunque algunas formas de robots humanoides pueden modelar sólo una parte del cuerpo.

Un objetivo de los robots humanoides es mejorar la experiencia en la interacción humano robot a través del desarrollo de tecnologías como la síntesis de voz, reconocimiento facial, interpretación de expresiones faciales, reconocimiento de voz, reconocimiento de objetos, interfaces con pantallas y expresar emociones a través de colores e movimientos del robot.

El desarrollo de los robots humanoides empezó con el robot ASIMO de Honda, específicamente partió de la investigación y experimentación de la locomoción bípeda desde 1986 con el robot E0 este prototipo tenía la capacidad de andar anteponiendo una pierna después de la otra, pero entre cada paso se demoraba 5 segundos debido a que debía estabilizar el centro de gravedad para no caerse. (Honda, 2016).

Honda continuo con el estudio de la locomoción bípeda en robots hasta lograr importantes avances en la estabilización del centro de gravedad, aumentar la velocidad de desplazamiento y caminar en superficies desiguales.

Robot ASIMO 2011



Figura 2.2. Robot ASIMO de honda

Fuente: (University, 2000)

La versión de ASIMO de 2011 Figura 2.2 reparte café, entrega mensajes, empuja un carrito y puede desplazarse a 6 km/h, además posee funciones como cargar su batería de forma autónoma, coordinación de varios ASIMO y calcular la velocidad de desplazamiento de las personas alrededor.

2.4 Clasificación de los Robots Humanoides

Los robots humanoides se categorizan en tres tipos Semi-Humanoide, tamaño pequeño, tamaño grande y otras aplicaciones muy específicas.

2.4.1 Robot semi-humanoides

Surgen de la falta de estabilidad de los robots bípedos los cuales no pueden trasladarse fácilmente y rápidamente, poseen una base con ruedas en vez de piernas. Algunos de los robots más significativos son los siguientes Tabla 2.2.

Tabla 2.2.

Ejemplos de robots semi- humanoides

Robot	Características
<p data-bbox="416 882 632 913">ENON – Fujitsu</p>  <p data-bbox="336 1137 600 1196">Figura 2.3. ENON. Fuente: (Gentuth, 2016)</p>	<p data-bbox="735 931 1315 963">Fecha de lanzamiento: 2005 / Precio: 60 000</p> <p data-bbox="735 976 1369 1043">Autónomamente ofrece soporte a clientes en oficinas o centros comerciales.</p> <p data-bbox="735 1057 1362 1088">Especificaciones: 54 cm (ancho) y 130 cm (alto)</p> <p data-bbox="735 1102 1209 1133">Peso: 50 Kg -- 10 Grados de libertad</p>
<p data-bbox="440 1209 608 1240">Rollin Justin</p>  <p data-bbox="336 1460 643 1563">Figura 2.4. Rollin. Fuente: (German Aerospace Center, 2007)</p>	<p data-bbox="735 1258 1098 1290">Fecha de lanzamiento: 2009</p> <p data-bbox="735 1303 1257 1335">Fabricado: Centro Aeroespacial Alemán</p> <p data-bbox="735 1348 1369 1415">Puede atrapar objetos a través de cámaras y software de localización</p> <p data-bbox="735 1429 1347 1460">Especificaciones: 1,7 m (diámetro) y 2 m (alto)</p> <p data-bbox="735 1473 1209 1505">Peso: 45 Kg -- 43 Grados de libertad</p>
<p data-bbox="395 1583 651 1615">Pepper - Aldebaran</p>  <p data-bbox="336 1856 651 1960">Figura 2.5. Pepper Fuente: (Soft Bank Robotics, 2014)</p>	<p data-bbox="735 1610 1315 1641">Fecha de lanzamiento: 2015 / Precio: 20 000</p> <p data-bbox="735 1655 1369 1767">Reconocer las principales emociones humanas y adaptar su comportamiento de acuerdo al comportamiento del usuario</p> <p data-bbox="735 1780 1369 1848">Especificaciones: 121 cm (alto) x 42 cm (espesor) x 48 cm (ancho)</p> <p data-bbox="735 1861 1369 1928">Peso: 28 Kg -- 17 Articulaciones y 3 ruedas omnidireccionales</p>

2.4.2 Humanoides de tamaño pequeño

Los humanoides de tamaño pequeño tienen una locomoción bípeda y son perfectos para trabajar con niños y adulto mayores. Se los considera ideales para aplicaciones de entretenimiento y se los compara con una mascota. Algunos de los robots más importantes se encuentran en la Tabla 2.3, a esta categoría pertenece el robot NAO de Aldebarán que se le detalla más adelante.

Tabla 2.3.

Ejemplos de robots humanoides de tamaño pequeño

Robot	Características
<p style="text-align: center;">ASIMO</p>  <p>Figura 2.6. ASIMO Fuente: (Honda, 2016)</p>	<p>Asimo "Advanced Step in Innovative Mobility" Robot creado por Honda ASIMO versión 2011 es una máquina autónoma con la capacidad de tomar decisiones y hacer cambios en su comportamiento de acuerdo al entorno en el que esté. Lo que se busca es la coexistencia con el ser humano.</p>
<p style="text-align: center;">SDR 3X</p>  <p>Figura 2.7. SDR Fuente: (Robotics Today SDR 3X, 2000)</p>	<p>Desarrollado por Sony en el 2000. Tiene la capacidad de cambiar la dirección mientras camina, pararse en una pierna, patear una pelota y bailar. Está compuesto de 24 juntas, micrófonos, sensores de toque y de posición.</p>
<p style="text-align: center;">HOAP 3</p>  <p>Figura 2.8. Hoap Fuente: (Robotics Today HOAP 3, 2005)</p>	<p>Fabricado por Fujitsu en el 2005 de arquitectura abierta soporta programación open C/C++, peso de 6,8 Kg con una altura de 48 cm, puede caminar sobre terreno plano y realiza movimientos de sumo. La característica importante es que aprende ciertas acciones y las utiliza en la situación oportuna.</p>

Nota: Los robots humanoides de tamaño pequeño más importantes.

2.4.3 Robot Humanoides de tamaño completo

Estos robots tienen una estructura similar a la de un cuerpo humano, pero difieren de los de tamaño pequeño en su aplicación, ya que deben de ser capaces de desarrollar todas las actividades que un ser humano adulto realiza en la vida diaria, es decir que deben tener capacidades de comunicación, de movimientos y de inteligencia. Algunos robots que se han desarrollado se los describe en la siguiente en la Tabla 2.4.

Tabla 2.4.

Ejemplos de robots humanoides de tamaño completo

Robot	Características
<p style="text-align: center;">ROMEO</p>  <p>Figura 2.9. Romeo Fuente: (Robotics A. , 2015)</p>	<p>Fabricado por Aldebaran Robotics en Francia Es un robot asistente personal con funciones de vigilancia e interfaces hombre máquina. Es usado en la exploración de una gama de soluciones para proporcionar asistencia en el hogar. Su estructura es muy fuerte. Su altura es de 1,46 m con un peso de 36 Kg</p>
<p style="text-align: center;">TOPSY</p>  <p>Figura 2.10. Topsy Fuente: (Borel, 2010)</p>	<p>Fabricado por Topsy en Vietnam Es un robot jugador de ping pong diseñado para jugar tenis de mesa contra un humano, con una altura de 1,88 m con un peso de 120 Kg su actual versión utiliza servomotores.</p>

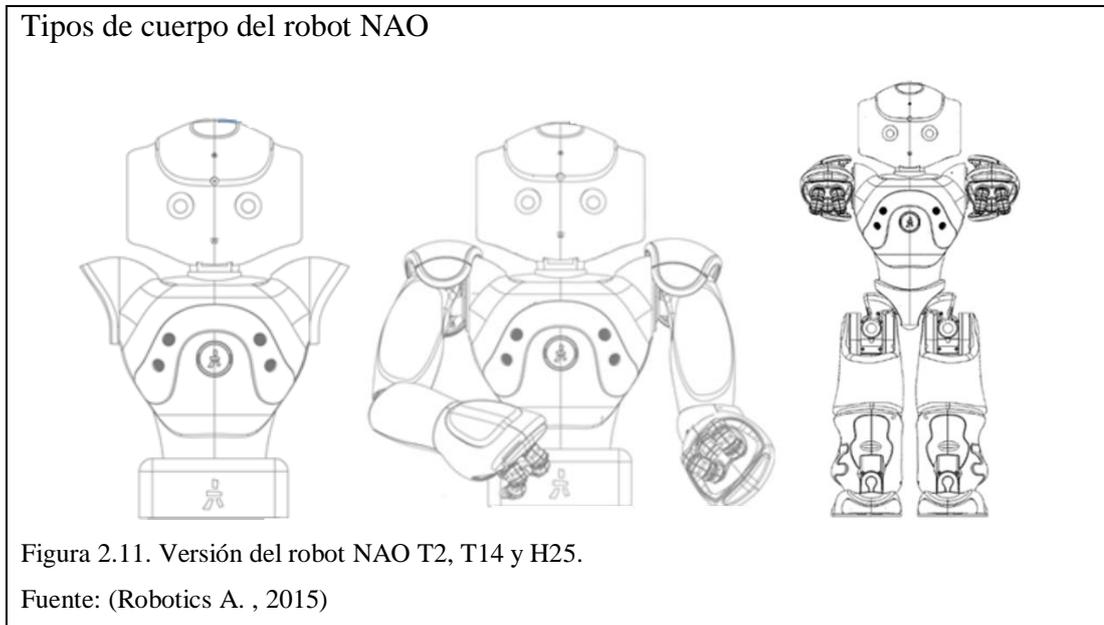
Nota: Los robots humanoides de tamaño completo más importantes construidos.

La construcción de estos robots surge de trabajos investigativos que buscan comprender la estructura y movimientos del cuerpo humano.

2.5 Robot NAO

El robot NAO es el primer robot humanoide creado por la empresa francesa Aldebaran Robotics, el cual tiene una altura de 58 cm, con más de 9000 unidades vendidas de la 5ta versión NAO Next Gen. Siendo NAO un robot interactivo con 25 grados de libertad que permiten movimientos humanoides casi naturales y caminar de forma

bípida en su versión académica completa H25, pero existen dos modelos más los cuales son el robot NAO que solo tiene cabeza y torso (versión T2), mientras que la segunda forma consta de cabeza, torso y dos brazos (versión T14) Figura 2.11.



Este robot cuenta con una etapa de control de bajo nivel, donde consta de un procesador embebido en la tarjeta ubicada en el torso del robot, el cual es capaz de leer todos los sensores dentro de un ciclo de 8 milisegundos y monitorea todos los motores para asegurarse que no haya daños por sobrecalentamiento y actualiza los nuevos valores de los actuadores de control.

El control de alto nivel es realizado por una placa de computadora embebida dentro de la cabeza del robot, la cual ejecuta el sistema operativo Linux con un procesador x86 AMD GEODE, y para la comunicación entre la parte de control de bajo nivel con la de alto nivel, cuenta con un kernel de tiempo real para el sistema operativo Linux llamado OpenEmbedded, adicionalmente tiene dos cámaras de tipo CMOS, un conector para Ethernet y comunicación Wi-Fi.

El robot NAO se puede comunicar con una Pc a través de cable Ethernet y también por medio de comunicación inalámbrica Wi- Fi. Puede interactuar con múltiples robots NAO usando sensores infrarrojos, red inalámbrica, cámaras, micrófono y sus altavoces, mientras que una forma de interactuar o comunicarse con el usuario es a través de sus entradas que pueden ser sensores de contacto, cámaras y micrófonos. Sus

salidas para comunicarse con el usuario son altavoces y leds ubicados en sus ojos, cabeza, torso y pies.

La empresa Aldebaran ofrece una forma de comunicarse mediante el SDK- Toolkit llamado NAOqi para desarrollo, la cual provee una interfaz al hardware de alto y bajo nivel usando los lenguajes de programación C/C++ y Python. Las especificaciones técnicas se detallan en la Tabla 2.5.

Tabla 2.5.

Especificaciones Técnicas del Robot NAO V5

Dimensiones					
Alto	Ancho	Profundidad	Peso	Material del armazón	Batería
574 mm	275 mm	311 mm	5.4 Kg	ABS-PC/PA-66/XCF-30	21.6v 2Ah Litio Polímero
CPU					
Procesador		Memoria RAM:	Memoria Flash	Memoria Micro SDHC	
Intel ATOM Z530 1.6 GHz		32 Bits	1 GB	2 GB	8 GB
Batería					
Voltaje operación	Voltaje máximo	Potencia	Carga	Autonomía:	
21.6 V / 2.25 Ah	25.2 V	48.6 Wh	< 3 horas	60 min modo activo	90 min modo normal

Nota: Especificaciones técnicas del robot NAO V5.

Fuente: (Robotics A. , 2015)

Las dimensiones del robot se muestran en la Figura 2.12.

Dimensiones del robot NAO

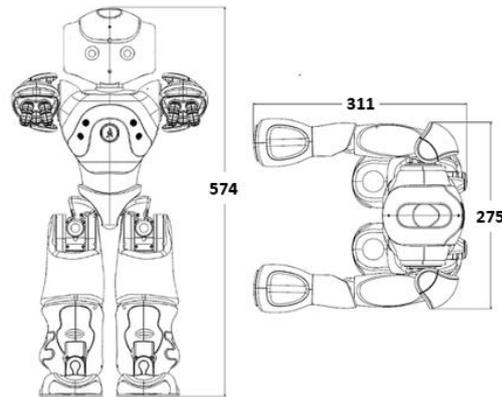


Figura 2.12. Dimensiones del robot humanoide NAO V5 H25.

Fuente: (Robotics A. , 2015)

2.5.1 Conectividad

El robot NAO, cuenta con un puerto Ethernet con conector RJ45 - 10/100/1000 base T, y comunicación inalámbrica Wi- Fi IEEE 802.11 a/b/g/n. Tiene un puerto USB en la parte trasera de su cabeza, con el cual se realizan las actualizaciones del sistema del robot, pero también permite conectar dispositivos externos como el Kinect de Xbox, el sensor 3D de Asus Xtion y dispositivos con comunicación serial que tengan el chip serial FTDI ya que NAO cuenta con NAOqi que usa Linux y tiene instalado los drivers para estos chips anteriormente mencionados. La ubicación de los puertos se encuentra en la Figura 2.13 (Robotics E. , 2015).

Conectividad del Robot NAO

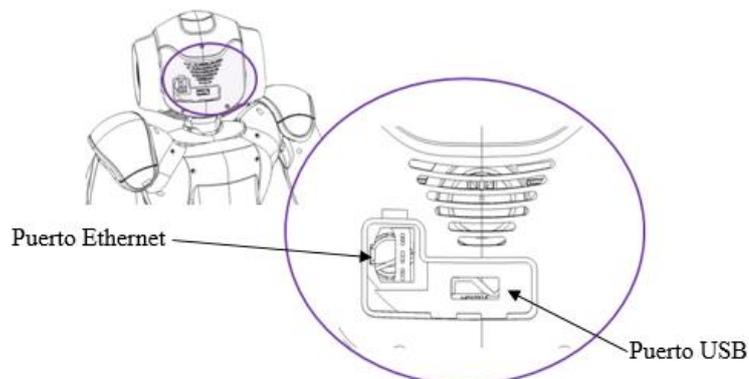


Figura 2.13. Puerto Ethernet y USB, detrás de la cabeza del robot NAO.

Fuente: (Robotics A. , 2015)

2.5.2 Interacción

Para la interacción con el entorno y personas el robot NAO, tiene 2 parlantes estéreo los cuales se encuentran ubicados en los extremos de su cabeza, 4 micrófonos para reconocimiento de voz y escuchar el entorno. Cuenta con 2 cámaras en la parte frontal de su cara, con una resolución de 1280x960 y 30fps, las cuales pueden ser usadas para identificar objetos dentro de su campo visual.

En la parte de cada uno de sus ojos cuenta con luces infrarrojas, las cuales tiene tres propósitos que son: usar al robot como control remoto, hacer que el robot reciba ordenes desde un control remoto o hacer que varios robots se comuniquen entre sí, pero no es recomendado por el fabricante.

Para hacer que el robot se comunique con otro robot usa un paquete de decodificación llamado LIRC (Linux Infrared Remote Control), el cual sirve para enviar y recibir información desde otro robot.

2.5.3 Sensores FSR

FSR, que sus siglas en ingles son Force Sensitive Resistors, los cuales se encuentran ubicados en la planta de los pies del robot teniendo 4 sensores en cada pie. Estos sensores cambian de resistencia acorde con la presión aplicada a ellos, funcionan en el rango de 0 -25 N.

2.5.4 Unidad de Inercia

La unidad de inercia de este robot consta de un giroscopio de 3 ejes cuya precisión mide hasta aproximadamente 500°/s y un acelerómetro de 3 ejes que soporta hasta 2g. La unidad de inercia del robot NAO se encuentra ubicada en el torso embebido en su tarjeta electrónica.

2.5.5 Sonares

Tiene 2 sensores ultrasónicos para medir la distancia a la que se encuentra los objetos en su entorno. Estos sensores ultrasónicos se encuentran ubicados en su torso, en el lado derecho se encuentra emisor y en el lado izquierdo un receptor.

Estos sensores funcionan a una frecuencia de 40 KHz, con una resolución de entre 1 a 4 cm dependiendo de la distancia, y pueden medir en el rango de los 20 cm hasta los 80 cm. Si el obstáculo se encuentra a menos de 20 cm los sensores no registran información del obstáculo, pero el robot detecta que hay un objeto presente, si el obstáculo se encuentra a más de 80 cm de distancia retornan un valor estimado de la distancia.

2.5.6 Sensores de posición

Usa 36 encoders magnéticos llamados MRE (Magnetic Rotary Encoders), los cuales basan su funcionamiento en efecto Hall, estos encoders tiene una precisión de 12 bits, donde 4096 pulsos por vuelta corresponden a 0.1° de precisión.

2.5.7 Sensores táctiles

El robot tiene principalmente sensores táctiles en 4 lugares de su cuerpo, en la cabeza, pecho manos y pies, los cuales son de diferentes tipos Tabla 2.6.

Tabla 2.6.

Sensores del robot NAO.

	Cabeza	Pecho	Mano	Pie
Tipo	Capacitivo	Pulsador	Capacitivo	Bumper
Cantidad	3	1	3 en cada mano	1 en cada pie

Nota: Tabla de sensores de contacto y táctiles del robot NAO.

Fuente: (Robotics A. , 2015).

2.5.8 NAOqi Framework (O.S)

NAOqi es el nombre del software principal que se ejecuta en el robot NAO para controlarlo, mientras que NAOqi Framework, es una plataforma robótica que se usa para programar los robots de Aldebaran. El cual se encarga de responder a las necesidades comunes de los robots como: paralelismo, sincronización, eventos y recursos.

El NAOqi framework, permite la comunicación homogénea entre distintos módulos. Permite programación homogénea y también compartir información de forma

homogénea. Debido a que es una plataforma cruzada y de lenguaje cruzado lo cual permite crear aplicaciones distribuidas.

2.5.9 Plataforma cruzada

Se usa este término debido a que NAOqi framework puede ser usada y desarrollar programas en Linux, Windows y MacOS.

2.5.10 Lenguaje cruzado

NAOqi framework funciona con lenguaje cruzado, ya que tiene la facilidad de poder programar los robots con lenguaje Python y C++. Donde usando Python el usuario puede ser capaz de ejecutar el código desde el computador o directamente en el robot. Mientras que, usando C++, es un lenguaje más complicado, el cual necesita ser compilado para cada sistema operativo. Por lo que, si se desea correr un programa en C++ en el robot, se necesita usar una herramienta para compilar código cruzado y de esta manera pueda ejecutar en el sistema operativo del robot NAOqi OS.

2.5.11 Aplicaciones distribuidas

Una aplicación en tiempo real soporta varios procesos y módulos que pueden ser ejecutados en varios robots a la vez. Se puede conectar una aplicación ejecutable a otro robot usando solamente la dirección IP y el puerto, y todos los métodos del API de otros ejecutables está disponible de la misma forma de los métodos. NAOqi selecciona automáticamente entre las llamadas rápidas directas (LPC) y las llamadas remotas (RPC).

2.5.12 Introspección

La introspección es el fundamento del API del robot, las capacidades, monitoreo y acciones sobre las funciones supervisadas. El robot sabe todas las funciones del API disponibles. Al asociar una función, usando tres líneas de código, automáticamente se beneficia de las siguientes características:

- Llama funciones en ambos lenguajes C++ y Python.
- Saber que funciones están siendo ejecutadas.
- Ejecutar funciones de forma local o remota.
- Llamadas en espera, paro, ejecutando funciones.

2.5.13 Broker

El broker conocido también como corredor o agente, es un objeto que proporciona:

- Servicios de directorio, permitiendo encontrar módulos y métodos.
- Acceso a la red, permitiendo a los métodos, de módulos adjuntos ser llamados desde afuera del proceso.

La mayor parte de las veces no se necesita preocuparse de los brokers debido a que estos hacen su proceso de una forma transparente.

2.5.14 Proxy

Un proxy es un objeto que se comportará como el modulo que representa. Por ejemplo, si se crea un proxy del módulo ALLeds, se obtendrá un objeto que contenga todos los métodos de ALLeds.

Para crear un proxy para un módulo, se tiene dos opciones:

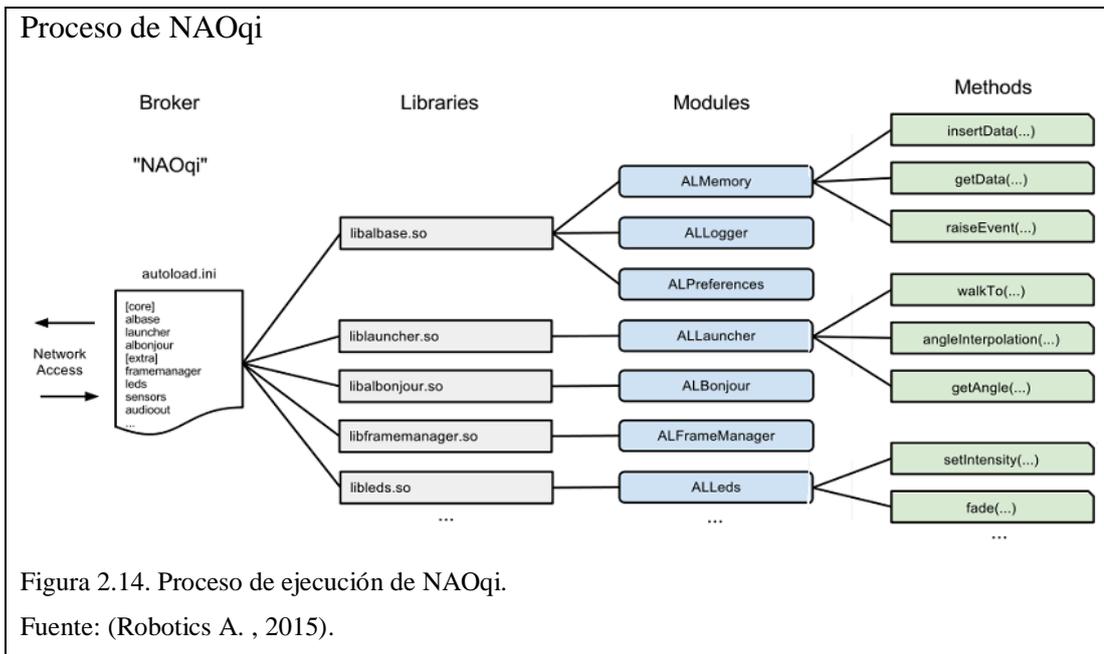
- Se usa el nombre del módulo, que en este caso será el código que se está corriendo y el modulo al cual se desea conectar deben estar en el mismo broker, esto recibe el nombre de “llamada local”.
- Usar el mismo nombre del módulo, la dirección IP y el puerto del broker. En este caso el modulo debe estar en el broker correspondiente.

2.5.15 Módulos

Cada módulo es una clase dentro de una librería. Cuando una librería es cargada del autoload.ini, esta automáticamente instancia la clase del módulo. Una vez que el constructor de una clase que deriva de ALModule, puede unir métodos. Esto advierte al broker los nombres y las firmas de los métodos para que estén disponibles para otros.

2.5.16 Proceso de NAOqi

El robot NAO trae un ejecutable de NAOqi el cual hace de broker en el robot o más conocido como broker. Cuando se inicia, carga el archivo de preferencias llamado autoload.ini, el cual define cuales librerías debe cargar. Cada librería contiene uno o más módulos que usan al broker para avisar sobre sus métodos Figura 2.14.



El broker provee los servicios de búsqueda de modo que cualquier modulo en el árbol o en la red pueda encontrar el método que se ha anunciado. De manera que cargar los módulos formando un árbol de métodos que conectan a los módulos, y los módulos conectan al broker.

2.6 Programación

NAO está basado en el software Gentoo de Linux y soporta varios lenguajes de programación como: C, C++, URBI, Python y .Net Framework y Choregraphe que es un lenguaje de programación gráfico para el robot Tabla 2.7.

Tabla 2.7.

Lenguajes de programación

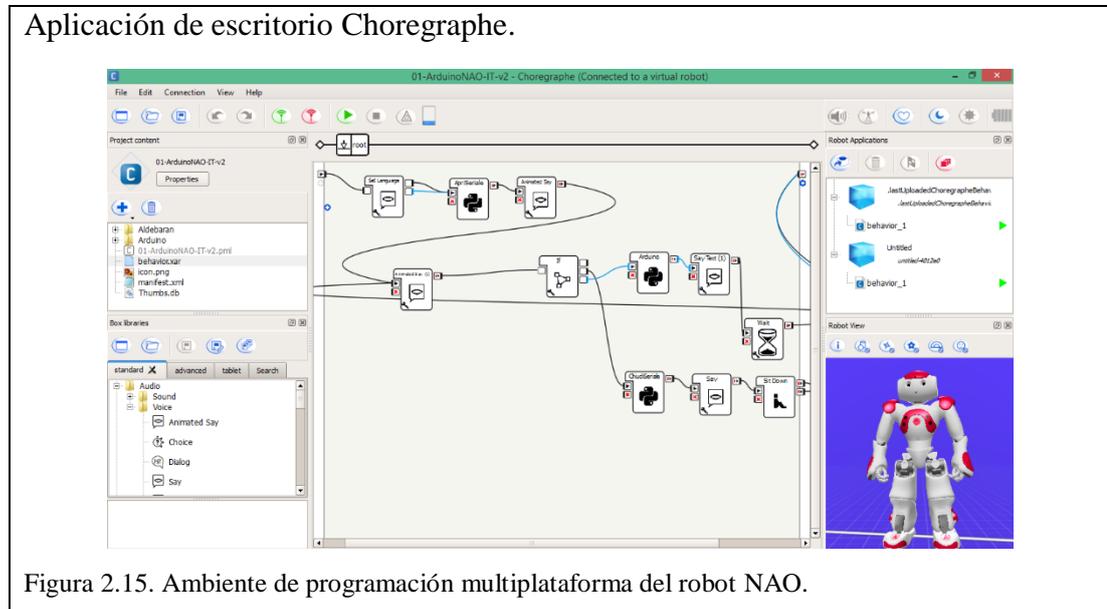
Lenguajes de Programación	Funciona en		Choregraphe	
	Computador	Robot	Hacer Apps	Editar código
Python	✓	✓	✓	✓
C++	✓	✓	✗	✗
Java	✓	✗	✗	✗
JavaScript	✓	✓	✓	✗

Nota: Tabla de lenguajes de programación que soporta el robot NAO.

Fuente: (Robotics A. , 2015)

2.6.1 Choregraphe

Choregraphe permite crear aplicaciones que contengan diálogos, servicios y behaviors complejos, así también como interacción con personas, enviar emails, sin insertar una sola línea de código, todo se lo hace en lenguaje gráfico Figura 2.15.



Choregraphe es una aplicación multi plataforma, permite al usuario:

- Crear animaciones y diálogos para el robot NAO.
- Probar en un robot simulado, o directamente en un robot real.
- Controlar y monitorear el robot.
- Mejorar los behaviors de Choregraphe con lenguaje Python.

2.6.2 Python 2.7 SDK e IDLE

El lenguaje Python es un lenguaje de programación interpretado el cual se enfoca a favorecer la legibilidad del código y su sintaxis Figura 2.16. Las ventajas de este lenguaje es que posee una licencia de código abierto y es un lenguaje de alto nivel incrustable, debido a que se puede insertar lenguaje Python dentro de un programa que este escrito en lenguaje C y C++ (Tecnologías, 2014).

Programación lenguaje Python.

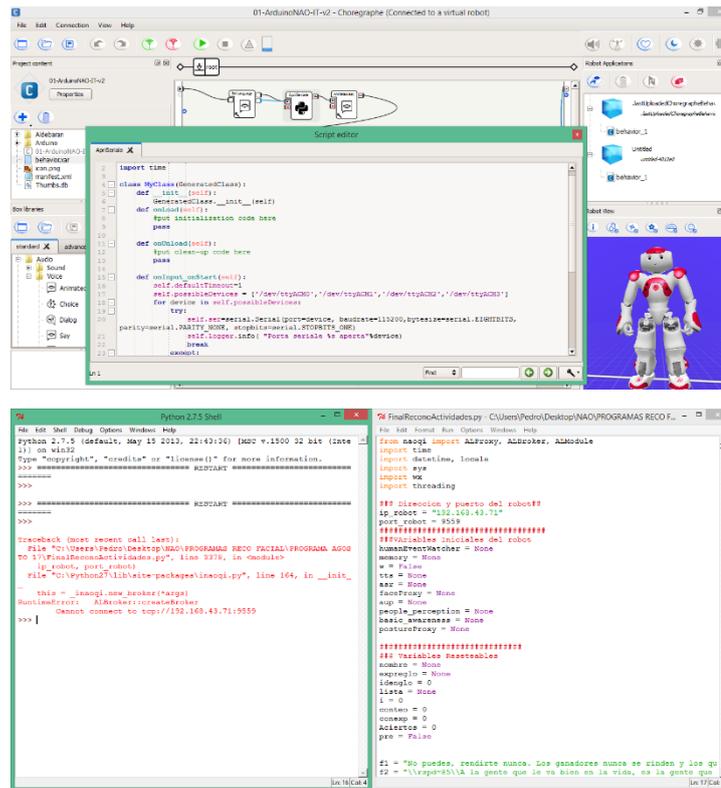


Figura 2.16. Scripts de Python, en Choregraphe e IDLE,

Se puede usar lenguaje Python de dos formas para programar el robot NAO. La primera forma es utilizando el script de Python que se encuentra en los bloques dentro del entorno gráfico de Choregraphe. Mientras que la segunda forma es abriendo un script del IDLE propio de Python, con el cual se puede acceder de manera remota al robot y también crear módulos y hacerlos correr de forma remota.

2.6.3 NAOqi Python API

Una parte del SDK de Python es usada internamente por Choregraphe. NAOqi es el módulo principal que usa Python para programar al robot NAO. Ahora visualizaremos tres partes fundamentales para la programación en Python, los cuales son ALProxy, ALBroker y ALModule.

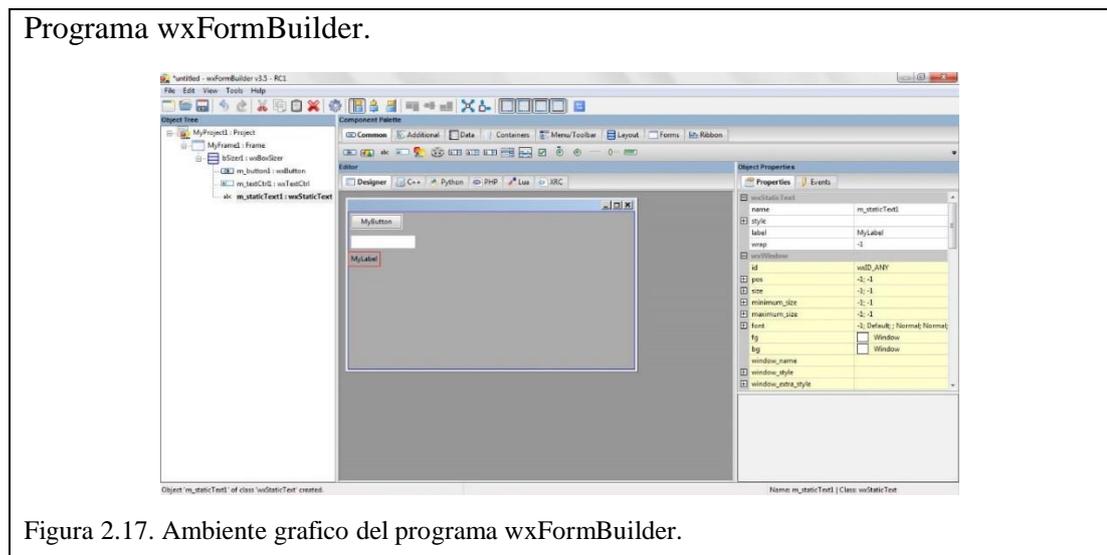
- ALProxy(name, ip, port): es el objeto el cual permite crear un proxy al módulo donde: *name*: es el nombre del módulo, *ip*: es la dirección IP del broker en la cual el módulo está corriendo y *port*: es el número de puerto del broker.
- ALBroker(name, ip, port): se usa cuando se necesita escribir módulos de NAOqi en Python, para cuando se necesite suscribirse a un evento con una

llamada. En *name*: nombre del módulo, *ip*: dirección ip del broker, *port*: el puerto del broker, si se lo pone en 0, buscara un puerto libre y lo usara.

- `ALModule(name)`: esta clase es la que permite escribir módulos de NAOqi en Python.

2.6.4 wxPython

wxPython es una herramienta de interfaz gráfica de usuario (GUI toolkit), para el lenguaje de programación Python, el cual permite crear programas robustos con una interfaz gráfica muy funcional de una forma fácil y sencilla Figura 2.17.

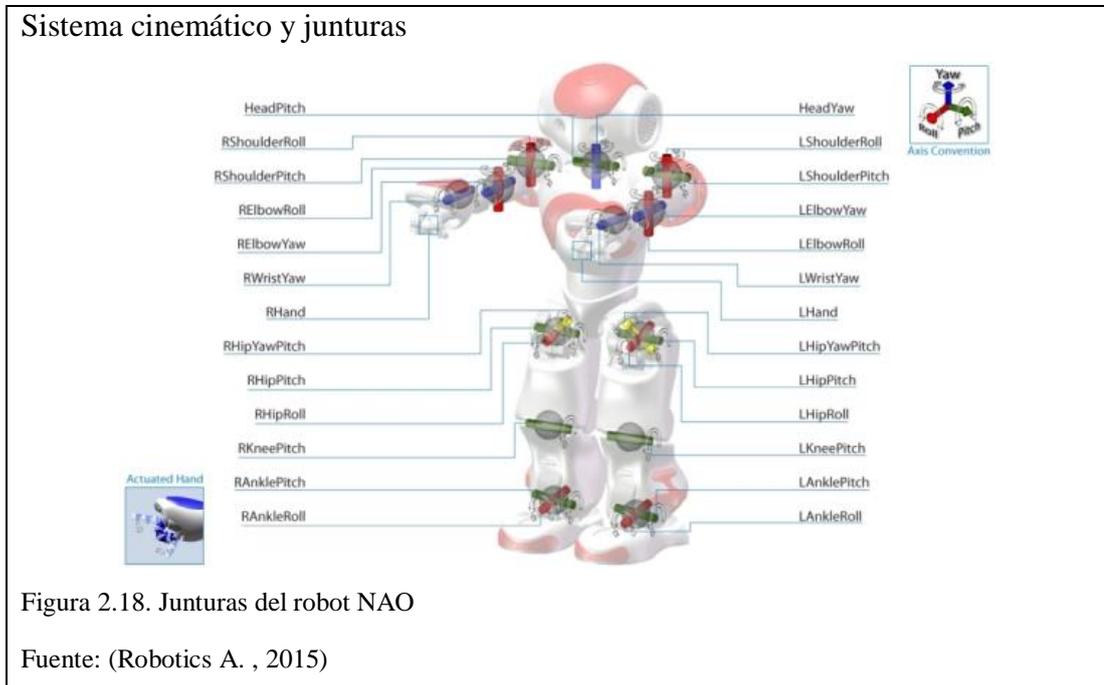


Para crear ventanas desde un entorno gráfico se puede utilizar el programa wxFormBuilder, el cual es un constructor que transforma la interfaz wxWidget GUI a formato Python, en el cual se editan botones, ventanas, frames de una forma gráfica y se controla desde el IDLE de Python.

2.7 Cinemática del Robot NAO

La cinemática es una rama de la mecánica clásica la cual describe el movimiento de los cuerpos y sistemas de los cuerpos sin tomar en cuenta la masa de dichos objetos ni las fuerzas que causan dichos movimientos. Por lo que se encarga del estudio de la geometría del movimiento, donde comienza con una descripción de la geometría del sistema y las condiciones iniciales y los valores conocidos de la posición, velocidad o aceleración de varios puntos que son parte del sistema, después con argumentos

geométricos se puede determinar la posición, velocidad y la aceleración de cada parte del sistema. El robot NAO V5 H25, es la versión completa del robot humanoide de Aldebaran el cual tiene 25 grados de libertad. Cuenta con enlaces, juntas, actuadores y uniones Figura 2.18.



2.7.1 Calculo de los grados de libertad con criterio de Kutzbach-Grübler

Para los cálculos de grado de libertad se utilizó la ecuación para sistemas tridimensionales.

$$GDL = 6(n - 1) - 5j_1 - 4j_2 - 3j_3 - 2j_4 - j_5 \quad (2.1)$$

Donde:

GLD = Grados de libertad de todo el mecanismo

n = Número de eslabones, incluyendo el eslabón fijo o de referencia

j_i = Número de pares o uniones de grados de libertad de las juntas.

Se estudió los grados de libertad con el brazo derecho del robot Figura 2.19, tomando en cuenta el tronco como parte fija por lo tanto hay 5 eslabones con 2 movimientos de 1 grado de libertad y 2 movimientos de 2 grados de libertad, tomando en cuenta el tronco como parte fija.

Brazo derecho del robot NAO

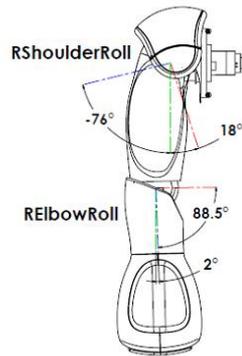


Figura 2.19. Estudio de los grados de libertad

Fuente: (Robotics A. , 2015)

Calculo de grados de libertad en el brazo derecho con la Ecuación 2.1

$$GDL = 6(n - 1) - 5j_1 - 4j_2 - 3j_3 - 2j_4 - j_5$$

$$n = 5, j_1 = 2, j_2 = 2$$

$$GDL = 6(5 - 1) - 5(2) - 4(2)$$

$$GDL = 6$$

Debido a que el robot Nao es un robot simétrico el cálculo del brazo izquierdo es igual que el lado derecho teniendo así 6 GDL, con un total de 12 GDL.

El estudio de los grados de libertad de la parte inferior del robot se lo realizó con la pierna derecha Figura 2.20. Tiene 5 eslabones con 2 movimientos de 1 grado de libertad y 2 movimientos de 2 grados de libertad, tomando en cuenta el tronco como parte fija.

Pierna izquierda del robot NAO

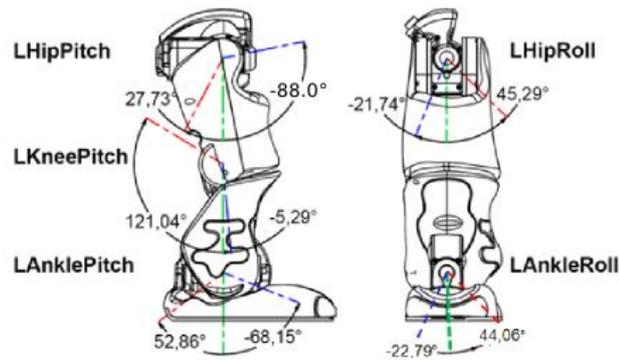


Figura 2.20. Pierna izquierda cálculos de cinemática.

Fuente: (Robotics A. , 2015)

Cálculos de grados de libertad pierna derecha con la Ecuación 2.1

$$GDL = 6(n - 1) - 5j_1 - 4j_2 - 3j_3 - 2j_4 - j_5$$

$$n = 4, j_1 = 1, j_2 = 2$$

$$GDL = 6(4 - 1) - 5(1) - 4(2)$$

$$GDL = 5$$

El cálculo de la pierna izquierda es igual que la del lado derecho teniendo así 5 GDL, con un total de 10 GDL en la zona inferior.

Para la cabeza del robot Nao hay 2 GDL, mientras que en la pelvis hay dos juntas las que están acopladas en un servo y no se pueden mover independientemente teniendo así 1 GDL, dando un total de:

$$GDL \text{ del robot Nao} = 25$$

2.7.2 Cinemática Directa del robot por Denavit-Haartenberg NAO

Para el cálculo de la cinemática directa e inversa del robot es necesario calcular independientemente cada brazo, pierna y cabeza tomando como referencia el torso.

Cálculo de cinemática directa de la pierna izquierda

En los cálculos anteriores se determinó que las piernas tienen seis juntas referirse Figura 2.20. Los parámetros se muestran en la Tabla 2.8

Tabla 2.8.

Parámetros de Denavit-Haartenberg

Juntura	a	α	d	θ
Torso	A (0, Cadera Y, - Cadera Z)			
LHipYawPitch	0	$-\frac{3\pi}{4}$	0	$\theta_{1-\frac{\pi}{2}}$
LHipRoll	0	$-\frac{\pi}{2}$	0	$\theta_{2+\frac{\pi}{4}}$
LHipPitch	0	$\frac{\pi}{2}$	0	θ_3
LkneePitch	- thighLength	0	0	θ_4
LAnklePitch	- TibiaLength	0	0	θ_5
LAnkleRoll	0	$-\frac{\pi}{2}$	0	θ_6
Rotación	$R_z(\pi)R_y\left(-\frac{\pi}{2}\right)$			
Final	A(0,0,-FoodHeight)			

La matriz de transformación queda de la siguiente manera.

$$T_{Inicio}^{Final} = A_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y\left(-\frac{\pi}{2}\right) A_6^{Fin} \quad (2.2)$$

Cinemática directa de piernas, brazos y cabeza

Los cálculos de cada articulación a través de Denavit-Haartenberg son los siguientes:

$$T_{Base}^{Cabeza} = A_{Base}^0 T_0^1 T_1^2 R_x\left(\frac{\pi}{2}\right) R_y\left(\frac{\pi}{2}\right) A_2^{Cabeza} \quad (2.3)$$

$$T_{Base}^{ManoIzq} = A_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 R_x\left(\frac{\pi}{2}\right) R_z\left(\frac{\pi}{2}\right) A_5^{ManoIzq} \quad (2.4)$$

$$T_{Base}^{ManoDer} = A_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 R_x\left(\frac{\pi}{2}\right) R_z\left(\frac{\pi}{2}\right) A_5^{ManoDer} \quad (2.5)$$

$$T_{Base}^{PieIzq} = A_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y\left(-\frac{\pi}{2}\right) A_6^{PieIzq} \quad (2.6)$$

$$T_{Base}^{PieDer} = A_{Base}^0 T_0^1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6 R_z(\pi) R_y\left(-\frac{\pi}{2}\right) A_6^{PieDer} \quad (2.7)$$

2.7.3 Cinemática inversa de la pierna izquierda del robot Nao

La cinemática de cada pierna del robot tiene 6 juntas, las cuales son:

θ_1 : Yaw/ Pitch de la cadera, θ_2 : Roll de la cadera, θ_3 : Pitch de la cadera, θ_4 : Pitch de la rodilla, θ_5 : Pitch del tobillo y θ_6 : Roll del tobillo.

Construyendo la ecuación para la transformación numérica y simbólica de matrices para formar un sistema no lineal:

$$\hat{T} = (A_{Base}^0)^{-1} T (A_6^{Final})^{-1}$$

Operando y teniendo como resultado las ecuaciones de traslación:

$$T'_{(1,4)} = l_2 \sin \theta_5 - l_1 \sin(\theta_4 + \theta_5), \quad T'_{(2,4)} = (l_2 \cos \theta_5 + l_1 \cos(\theta_4 + \theta_5)) \sin \theta_6,$$

$$T'_{(3,4)} = (l_2 \cos \theta_5 + l_1 \cos(\theta_4 + \theta_5)) \cos \theta_6$$

Donde las longitudes: l_1 es el largo del muslo y l_2 es el largo de la tibia.

Teniendo en cuenta que, para los cálculos de las piernas, la simetría de las mismas emerge a valores opuestos tanto para la cadera como para los tobillos y las demás uniones son idénticas mostradas en la ecuación.

$$T_{Base}^{PiernaIzq}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = M T_{Base}^{PiernaDer}(\theta_1, -\theta_2, \theta_3, \theta_4, \theta_5, -\theta_6) M$$

Para finalizar tenemos las ecuaciones de los ángulos $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$: y θ_6 .

$$\theta_1 = \cos^{-1} \left(\frac{T''_{(1,3)}}{\sin(\theta_2 + \frac{\pi}{4})} \right) + \frac{\pi}{2} \quad (2.8)$$

$$\theta_2 = \pm \cos^{-1} (T''_{(2,3)}) - \frac{\pi}{4} \quad (2.9)$$

$$\theta_3 = \sin^{-1} \left(\frac{T''_{(2,2)}}{\sin(\theta_2 + \frac{\pi}{4})} \right) \quad (2.10)$$

$$\theta_4 = \pm \left(\pi - \cos^{-1} \left(\frac{l_1^2 + l_2^2 - \|\bar{0} - \bar{p}\|_2}{2l_1 l_2} \right) \right) \quad (2.11)$$

$$\theta_5 = \sin^{-1} \left(\frac{T''_{(2,4)}(l_2 + l_1 \cos \theta_4 + l_1 T''_{(1,4)} \sin \theta_4)}{l_1^2 \sin^2 \theta_4 + (l_2 + l_1 \cos \theta_4)^2} \right) \quad (2.12)$$

$$\theta_6 = \begin{cases} \tan^{-1} \left(\frac{T'_{(2,4)}}{T'_{(3,4)}} \right) & \text{si } (l_2 \cos \theta_5 + l_1 \cos(\theta_4 + \theta_5)) \neq 0 \\ \text{indefinido} & \text{si } (l_2 \cos \theta_5 + l_1 \cos(\theta_4 + \theta_5)) = 0 \end{cases} \quad (2.13)$$

2.8 Reconocimiento de Expresiones Faciales

El estudio de las expresiones faciales empezó con la investigación de como expresan emociones el ser humano y los animales. (Darwin, 1872).

Se determinó 6 expresiones faciales fundamentales que son universales sin importar la cultura. La expresión facial de felicidad, tristeza, sorpresa, miedo, enojo y disgusto es la misma en europeos, africanos, americanos y asiáticos. (Ekamn & Friesen, 1971)

En la teoría de la comunicación de emociones se determinó que el estado de ánimo está compuesto de un 7 % lo que dice, 38 % la entonación con que lo dice y en un 55% la expresión facial. (Mehrabian, 1972).

Los primeros trabajos de reconocimiento de expresión facial que marcaron dos líneas de desarrollo son los de método base y el método basado en las características.

En esencial el método base utiliza un set de datos de imágenes de personas previamente procesadas de las cuales se toma de referencia y se compara arrojando el mejor resultado cuando se presenta una imagen de una persona. (Cootes, Edwards, & Taylor, 1998).

El método basado en las características es un método que analiza el movimiento y deformación de la boca cejas y ojos. Y según condiciones define la expresión que está expresando. (Black & Yacoob, 1997).

Se han desarrollado varios trabajos de investigación para aumentar el grado de éxito en el reconocimiento de expresiones faciales y se le ha encontrado potencial en la detección de mentiras, vigilancia y robots humanoides. El reconocimiento se implementó en los robots humanoides para mejorar la interacción humano robot, basándose en que la comunicación entre humanos está influenciada por las emociones.

2.9 Necesidades del Adulto Mayor

El proceso de envejecimiento cuenta con una serie de situaciones como es la disminución de las facultades físicas, padecimiento de enfermedades crónicas, disminución de recursos económicos, aislamiento, pérdida de seres queridos, etc. los que pueden desencadenar aparición de procesos psicopatológicos. (Chicaiza, 2011)

Uno de los mayores males que afectan al adulto mayor producido por la recesión de sus funciones fisiológicas, el deterioro de su capacidad articular y muscular es el

sedentarismo el cual muestra la imagen de una persona inútil, enferma e inactiva. Para la lucha con el sedentarismo no hay actividades específicas. Cualquier manifestación corporal, ya sea a través de juegos libres que respeten las capacidades del adulto mayor combaten el sedentarismo.

Ayudar al adulto mayor no solo se refiere alcanzar mejoras en un sentido médico o rehabilitador de discapacidades funcionales, sino de alcanzar un sentimiento de bienestar y autosuficiencia, un alivio de la tensión psíquica que genera la vejez. (Ruiz Munera, 2005)

Otro factor que se ve afectado en el adulto mayor es su salud mental ya que surgen trastornos emocionales como padecimientos de ansiedad, depresión y baja autoestima, teniendo pensamientos de que son una carga para la familia e inútiles. En esencia para la parte cognitiva del adulto mayor se han desarrollado varias terapias como la de reminiscencia, orientación a la realidad y ludoterapia que son muy usadas en centros geriátricos.

La terapia de reminiscencia es un método agradable, bien aceptado y de fácil aplicación ya que, a partir de viejas fotografías, libros, revistas, vestimentas, artículos domésticos, etc. se rememoran acontecimientos históricos y vitales del paciente. Se prioriza las vivencias que suscita al paciente frente a la precisión del recuerdo. Su objetivo es estimular la identidad y autoestima del paciente. Por ser un método generalmente aplicado de forma grupal favorece las relaciones sociales y comunicativas (Baynes, Saxby, & Ehlert, 1987)

La orientación a la realidad es un método que se basa en la orientación básica como la información referente a la fecha y al lugar, datos personales y circunstanciales relevantes para el paciente, que ayudan a los pacientes confusos y es necesaria para el desarrollo de otras funciones cognitivas. Este método se lo realiza de forma individual o grupal, puede mejorar significativamente el grado de “desorientación” y comunicación. (Baynes, Saxby, & Ehlert, 1987).

La ludoterapia busca que el adulto mayor sea incentivado a realizar actividades lúdicas como: terapias de juegos, música, bailes, arte y conjuntamente con la integración del cuerpo con la mente.

Teniendo como base que la recreación es una necesidad vital del adulto mayor, que le brindan un bienestar tanto físico como mental y terapéutico. Teniendo como beneficio liberar al adulto mayor del sedentarismo y la inactividad permitiendo así que pueda interactuar con su entorno. (Pérez, 2010)

2.10 Estado del Arte para aplicaciones con el robot NAO

Se han realizado aplicaciones de interacción humano robot con robots humanoides en el campo de la medicina y educación, aunque complejas en su desarrollo han presentado buenos resultados. Como la rehabilitación de problemas neuronales, la parálisis cerebral, esclerosis múltiple y autismo.

Uno de los proyectos de interacción humano robot desarrollados en el campo de la medicina consiste en analizar la respuesta de los niños con autismo al interactuar con el robot NAO para encontrar nuevos procedimientos en el tratamiento del autismo Figura 2.21. (Shamsuddin, y otros, 2012).

Terapia con el robot NAO



Figura 2.21. Respuesta inicial de los niños con autismo en una terapia con el robot NAO.

Fuente: (Shamsuddin, y otros, 2012)

Para fines educativos han sido utilizados exitosamente en la transferencia de conocimientos de la diabetes a niños, como soporte para su vida diaria con la

enfermedad, en este proyecto se utilizó el robot NAO Figura 2.22 (Henkemans, BP, & Bosch, 2013).



En escuelas para la enseñanza de geografía con la creación de un juego que pide a los estudiantes señalar en un mapa los lugares. El robot NAO juega como participante o como instructor Figura 2.23. (Ribeiro, Pereira, Deshmukh, Aylett, & Paiva, 2011) .



Unos de los más interesantes que se están desarrollando con el robot NAO, trata de utilizar al robot como una interfaz para la inteligencia cognitiva de IBM Watson.

CAPÍTULO 3

DESARROLLO E IMPLEMENTACIÓN

En este capítulo se detalla los módulos NAOqi utilizados y la programación en Choregraphe y Python.

3.1 Módulos de NAOqi

Los módulos del robot que se utilizaron para desarrollar la aplicación se los programó en Choregraphe y Python. Cada módulo contiene una lista de métodos y eventos.

3.1.1 Síntesis de Voz

Para la síntesis de voz en el robot NAO se usa el módulo ALTextToSpeech el cual permite al robot hablar. Este envía comandos al motor de texto-a-hablar (text-to-speech engine) y también permite la personalización de la voz como: entonación, acentos, velocidad, distorsión aguda y grave. El resultado de la síntesis de voz se puede escuchar a través de los parlantes ubicados en la cabeza del robot.

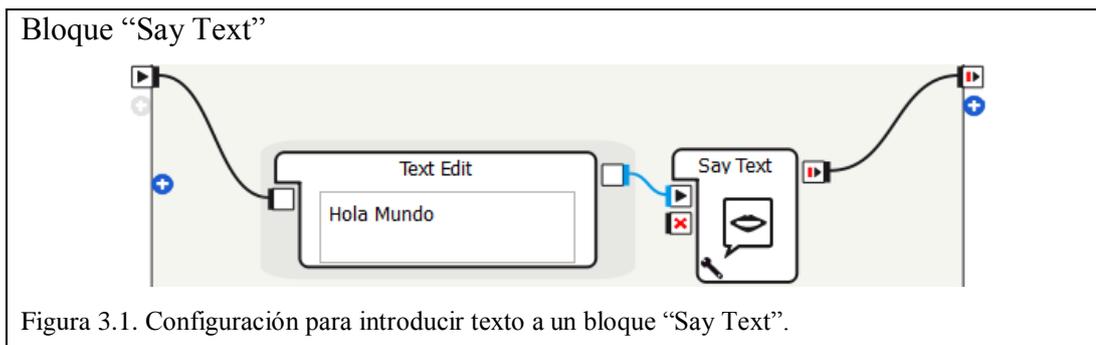
Programación en Choregraphe

En Choregraphe existen dos bloques los cuales nos permiten hacer que el robot hable, el primero es el bloque “Say text” y el segundo es el bloque “Say”, los cuales contienen dos entradas y una salida de tipo bang. Al ser entradas y salidas (I/O) de tipo bang significa que representan eventos simples, es decir, que no llevan consigo ningún tipo de información, solo son estimuladas y la información o dato se pierde.

Bloque “Say Text”

Este bloque Figura 3.1 se encarga de decir el texto que recibe por su entrada, puede ser desde otro bloque que retorne texto o un simple bloque editor de texto (Text Edit), que envía texto por su salida cuando su entrada es estimulada.

Básicamente es un bloque con código en Python, el mismo que usa el módulo de síntesis de voz ALTextToSpeech, tomando las propiedades guardadas por defecto o previamente manipuladas en el robot como: volumen, velocidad de voz y entonación.



Bloque "Say"

Este bloque permite ingresar texto, hay que abrirlo haciendo doble clic para entrar a su configuración, donde permitirá realizar cambios de parámetros de voz.

Programación en Python

En Python no se admite tildes, ni la letra ñ para esto se debe cambiar las palabras y utilizar comandos para crear una entonación. Los comandos van entre la frase que va a hablar el robot Tabla 3.1.

Tabla 3.1.

Comandos de entonación

Comando	Descripción
<code>\\rspd=85\\</code>	La velocidad a la que se lee el texto va desde 50 – 400.
<code>\\emph=0\\</code>	Reducir el énfasis en una palabra o una sílaba.
<code>\\emph=1\\</code>	Recaltar una palabra o una silaba.
<code>\\emph=2\\</code>	Acentuar como una tilde en una palabra o una silaba.
<code>"pitchShift", 1</code>	Se refiera al tono de la voz va desde 1.0 - 4. Recomendable de 1.0 para una voz masculina y 1.3 para una voz femenina.
<code>\\bound=W\\</code>	Entonación débil a una palabra.
<code>\\bound=S\\</code>	Entonación fuerte a una palabra.
<code>\\bound=N\\</code>	Sin entonación a una palabra
<code>\\pau=100\\</code>	Se logra una pausa en milisegundos, se utiliza para dar mayor importancia a una palabra.
<code>\\vct=90\\</code>	Cambia el tono va desde 50 a 200.

Los comandos se usan de acuerdo a la palabra puede ser uno o varios, pero no solo los comandos pueden hacer un acento, si se escribe una doble silaba sonara como un

acento. Se debe conectar al módulo de síntesis de voz para empezar a utilizar los métodos, para esto es necesario la siguiente programación Figura 3.2.

Programación de la síntesis de voz

```
from naoqi import ALProxy, ALBroker, ALModule

ip_robot = "192.168.43.71"
port_robot = 9559

tts = ALProxy("ALTextToSpeech", ip_robot, port_robot)
tts.setLanguage('Spanish')
tts.say('Hola UPS')
```

Figura 3.2. Programación necesaria para conectarse al módulo de síntesis de voz.

Se fija el idioma a través del método setLanguaje y para que hable el método say.

3.1.2 Correr un Behavior

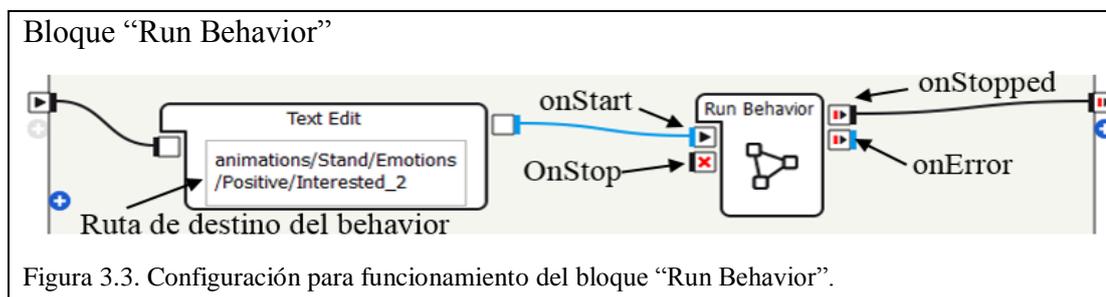
Este módulo se llama ALBehaviorManager, pertenece al núcleo de NAOqi su función es la gestionar los behaviors con métodos que permiten cargar, detener, iniciar, predeterminar y obtener información de los behaviors.

Programación en Choregraphe

Para correr un behavior desde Choregraphe se usa el bloque “Run Behavior”, es un bloque escrito en Python, contiene dos entradas y dos salidas.

Modo de funcionamiento del bloque “Run Behavior”

La primera entrada onStart Figura 3.3 es conectada a un cuadro de texto el cual debe contener la dirección del behavior a correr, nótese que previamente cada behavior previamente debe estar guardado dentro del robot. Con la segunda entrada onStop su conexión es opcional debido a que esta sirve para detener el funcionamiento del bloque.



Con respecto a sus salidas en la primera salida onStopped envía una señal al momento en que termina de correr el behavior. En la segunda salida onError se activará si un error ha ocurrido, transmitiendo un string con el mensaje de error.

Programación en Python

En Python se debe conectar al módulo de gestión de behavior para empezar a utilizar los métodos, para esto es necesario la siguiente programación Figura 3.4.

```

Programación de gestión de behaviors

from naoqi import ALProxy

### Conexión con el modulo de Gestión de Behaviors
managerProxy = ALProxy("ALBehaviorManager", "192.168.10.115", 9559)
### Asigna a una variable los Behaviors instalados en el robot con el
### Método getInstalledBehaviors()
Instalados = managerProxy.getInstalledBehaviors()
## Imprime los Behaviors
print "Behaviors en el robot"
print Instalados
### Consultar un Behavior si esta instalado
behaviorName = 'juego-6428be/behavior_1'
if (managerProxy.isBehaviorInstalled(behaviorName)):
    print "si"
### Para Correr un behavior el robot debe estar en posición de pie
### Conexión con el modulo de movimiento
motion = ALProxy("ALMotion", "192.168.10.115", 9559)
### Posición de pie
motion.wakeUp()
### Iniciar el behavior
managerProxy.runBehavior(behaviorName)

```

Figura 3.4. Conexión con el módulo, consulta y correr un behavior.

Los métodos indispensables para el gestionar los behaviors se explican en la Tabla 3.2 sin embargo existen más métodos que se encuentran en la documentación del módulo.

Tabla 3.2.

Métodos del módulo behaviors

Métodos	Descripción
getInstalledBehaviors	Proporciona los behaviors instalados en el robot
isBehaviorInstalled	Devuelve un verdadero o falso sobre un behavior en particular.
runBehavior	Inicia un behavior
stopBehavior	Detener un módulo en específico.
stopAllBehaviors	Detener todos los Behavior este método resulta práctico.

Para correr un behavior es necesario colocar al robot en posición de pie.

3.1.3 Creación de Movimientos

El modulo que se encarga de la locomoción del robot NAO se llama ALMotion, el cual contiene métodos para el control y reflejos del robot. Este módulo da información del hardware del robot como, por ejemplo: el número de articulaciones, los nombres de dichas articulaciones y los límites de los movimientos.

ALMotion funciona a una frecuencia de 50Hz, cada vez que se llama al API para hacer una solicitud de movimiento se crea una tarea de movimiento (motion task), para manejar dicha solicitud.

Debido a que este es un módulo de núcleo, maneja ciertos reflejos y actualizaciones de las posiciones del robot cada ciclo, lo que hace que el consumo de CPU sea constante. El robot NAO utiliza las unidades del sistema internacional (SI).

Para el control tiene cuatro grupos de métodos:

- Rigidez de las articulaciones: controla la rigidez de las juntas, es decir, encender y apagar motores.
- Posición de las articulaciones: interpolación y reactivación del control.
- Caminar: se encarga del control de la distancia y velocidad del robot además de la ubicación del robot.
- Desplazamiento del robot en el plano, cinemática inversa y las restricciones de todo el cuerpo.

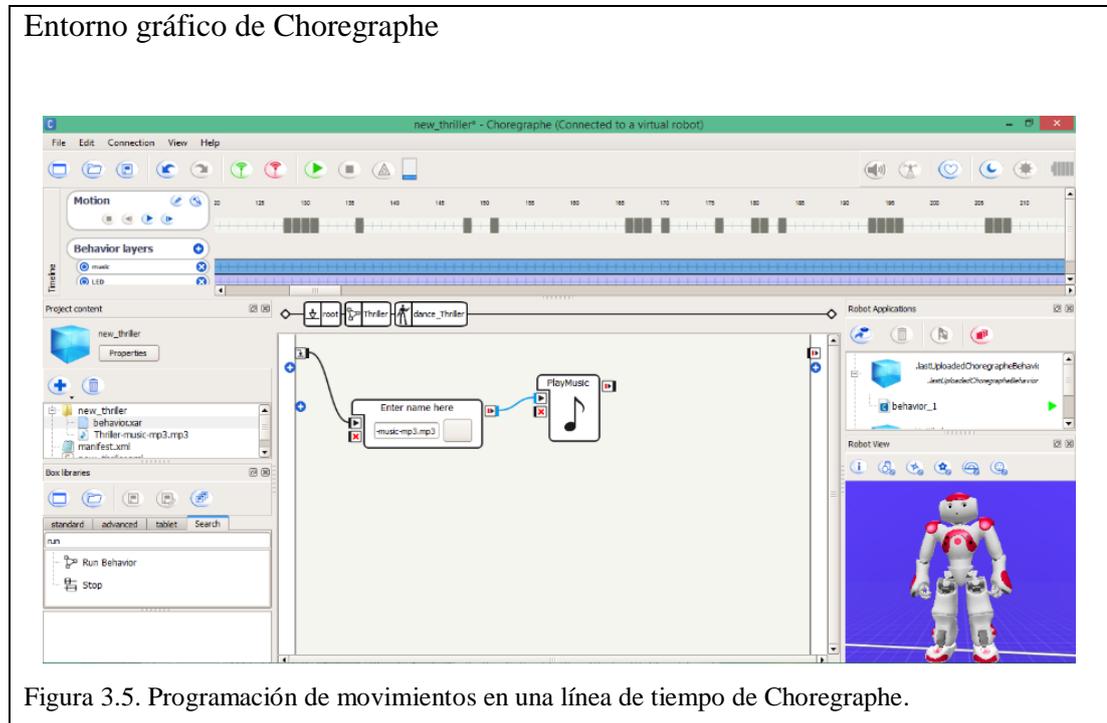
El módulo ALMotion, implementa algunos “reflejos” como:

- Evadir auto colisiones.
- Evadir colisiones externas.
- Asistente de caídas.
- Rigidez inteligente.
- Efecto de diagnóstico

Programación en Choregraphe

Choregraphe Figura 3.5 es una herramienta de programación que permite programar movimientos en una línea de tiempo, quitando la rigidez de las articulaciones y

moviéndolas a conveniencia. Lo que se puede destacar de este método de programación de forma gráfica es que los movimientos grabados en una línea de tiempo se pueden exportar a un script de Python, con absolutamente toda la cinemática y reproducirla fuera de Choregraphe, directamente desde el Idle.



Exportación de movimientos desde Choregraphe a Python

Se pueden exportar movimientos en dos lenguajes en C++ y en Python. Y también exportar movimientos en dos tipos de la forma: simplificada (simplified) y en bezier.

Programación en Python

Los movimientos se realizaron por los métodos de interpolación de ángulos simplificada e interpolación de Bezier. Para mover una o dos articulaciones se usa la interpolación de ángulos simplificado, pero para movimientos de todo el cuerpo es necesaria la interpolación de Bezier la programación se muestra en Figura 3.6.

Programación de movimientos

```
from naoqi import ALProxy
### Variables
names = list()
times = list()
keys = list()
### Datos de las variables obtenidos de Choregraphe
## Nombre de la Articulacion
names.append("LAnklePitch")
###Tiempo en el cual realiza el movimiento
times.append([0.933333, 3])
### Bezier
keys.append([[0.121359, [3, -0.311111, 0], [3, 0.688889, 0]], [-0.409751,

names.append("LAnkleRoll")
times.append([0.933333, 3])
keys.append([[0.0153604, [3, -0.311111, 0], [3, 0.688889, 0]], [-0.116564,
### Conexión con el módulo de movimiento
motion = ALProxy("ALMotion", "192.168.10.103", 9559)
### conexión con el módulo de posturas
postureProxy = ALProxy("ALRobotPosture", "192.168.10.103", 9559)

motion.wakeUp()
motion.angleInterpolationBezier(names, times, keys)
## postura de pie
postureProxy.goToPosture("Stand", 0.5)
```

Figura 3.6. Manejo de movimientos con bezier.

Los métodos para Bézier es `angleInterpolationBezier` y para interpolación por ángulo es `angleInterpolation`.

3.1.4 Reconocimiento de Expresión Facial

Este módulo se llama `ALFaceCharacteristics`, pertenece a la percepción de personas de NAOqi su función es la de analizar el rostro de las personas para detectar características como el género, edad, grado de sonrisa y la estimación de la expresión facial. Para funcionar se necesita utilizar el módulo `ALBasicAwareness` para rastrear y asignar una identificación a las personas detectadas.

Programación en Python

El módulo `analyzeFaceCharacteristics` depende de la identificación que se asigna el rastreo de rostros. Una vez que se rastrea un rostro a este se le asigna una identificación numérica a la cual se le aplica el análisis de las características, y se procede a extraer los datos de la memoria sobre el análisis de la expresión facial, estos datos se procesan con un algoritmo que determina el dato con mayor valor, que corresponde a la estimación de la expresión. La programación se muestra en la Figura 3.7.

Obtener la expresión

```
def get_people_perception_data(self, id_person_tracked):
    print "obtener expresion"
    global basic_awareness
    memory = ALProxy("ALMemory", ip_robot, port_robot)
    people_perception = ALProxy("ALFaceCharacteristics", ip_robot, port_robot)
    people_perception.analyzeFaceCharacteristics(id_person_tracked)
    memory_key = "PeoplePerception/Person/" + str(id_person_tracked) + "/ExpressionProperties"
    try:
        ex = memory.getData(memory_key)
    except Exception, e:

        print "Error obtener la expresion"
        print str(e)
        memory.subscribeToEvent("ALBasicAwareness/HumanTracked",
                                "humanEventWatcher",
                                "onHumanTracked")

        basic_awareness.stopAwareness()
        basic_awareness.startAwareness()
    print ex
    return ex
```

Figura 3.7. Función obtener la expresión facial.

El movimiento del robot en la aplicación dificulta realizar el análisis de las expresiones por esta razón utilizamos un try y en caso de un error se reinicia el rastreo de rostros. Los datos del análisis de las expresiones tienen un formato [Tranquilo, Feliz, Sorprendido, Enojado, Triste] cada expresión tiene un valor de 0 a 1. Para determinar el mayor desarrollamos la siguiente función Figura 3.8.

Función de comparación

```
def comparacion(self, num1, num2, num3, num4, num5):
    if (num1 >= num2 and num1 >= num3 and num1 >= num4 and num1 >= num5 ):
        return "tranquilo"
    if (num2 >= num1 and num2 >= num3 and num2 >= num4 and num2 >= num5 ):
        return "feliiz"
    if (num3 >= num1 and num3 >= num2 and num3 >= num4 and num3 >= num5 ):
        return "sorprendido"
    if (num4 >= num1 and num4 >= num3 and num4 >= num2 and num4 >= num5 ):
        return "enojado"
    if (num5 >= num1 and num5 >= num3 and num5 >= num4 and num5 >= num2 ):
        return "triste"
```

Figura 3.8. Comparación del número mayor entre cinco números.

Esta función retorna el número mayor que equivale a la estimación de la expresión que se está mostrando.

3.1.5 Guardar y Reconocer Rostros en el Robot

ALFaceDetection es el módulo que se encarga de hacer que el robot trate de detectar y reconozca los rostros que se encuentren delante de él, todo dependiendo de los métodos usados en la programación. La detección y reconocimiento se basa en una solución ofrecida por la empresa OMRON.

Para detectar rostros (Face detection), provee la información de la posición de la cara detectada, así también una lista de las coordenadas angulares con las características importantes del rostro detectado como son: ojos, nariz y boca.

Programación en Python de guardar rostros

Este módulo funciona independientemente del rastreo de rostros y contiene un evento llamado FaceDetected que tiene dos formas de llamarlo la primera es creando un evento interrupción que se está asociado a una función como en la Figura 3.9. El otro método es consultar a la memoria del evento un número determinado de veces con un tiempo de intervalo el cual posee un mejor desempeño.

Reconocer rostros

```
def Face(self):
    print "Face"
    global memory, nombre, lista, w, faceProxy, conteo, expreglo, conexp,
    memValue = "FaceDetected"
    for i in range(0, 20):
        time.sleep(0.5)
        val = memory.getData(memValue)
        print ""
        print "*****"
        print ""
        # Chequea si se encontro un rostro
        if(val and isinstance(val, list) and len(val) >= 2):
            timeStamp = val[0]
            faceInfoArray = val[1]
            try:
                # Browse the faceInfoArray to get info on each detected face
                for j in range( len(faceInfoArray)-1 ):
                    faceInfo = faceInfoArray[j]
                    # First Field = Shape info.
                    faceShapeInfo = faceInfo[0]
                    # Second Field = Extra info (empty for now).
                    faceExtraInfo = faceInfo[1]
                    print " faceID %.3f " % (faceExtraInfo[0])
                    print " ScoreRecord %.3f " % (faceExtraInfo[1])
                    print "pp"
                    print "kk", faceExtraInfo[2], "kk"
                    print "pp"
```

Figura 3.9. Esta función contiene toda la información del rostro de la persona.

FaceDetected retorna los siguientes datos Figura 3.10

Datos retornados de FaceDetected

```
FaceDetected = [ TimeStamp, [ FaceInfo [N], Time_Filtered_Reco_Info], CameraPose_InTorsoFrame, CameraPose_InRobotFrame, Camera_Id]

FaceInfo = [ ShapeInfo, ExtraInfo [N] ]

ExtraInfo[N] = [ faceID, scoreReco, faceLabel, leftEyePoints, rightEyePoints, unused, unused, nosePoints, mouthPoints ]
```

Figura 3.10. Información esencial de los rostros.

Una vez puesto a funcionar el evento de detección de rostros FaceDetected(), retorna vectores con información como:

- **TimeStamp:** es la marca de tiempo que se usa para hacer la detección de rostros.
- **FaceInfo:** es el campo con la información del rostro detectado, para cada rostro es distinto.
- **ShapeInfo:** indica la información del rostro, como ubicación y el tamaño.
- **ExtraInfo:** entrega la información de la forma del rostro y contiene algo esencial que es el número de identificación asignado a dicho rostro (faceID).

Se extraen los campos scoreReco y faceLabel que indican el grado de certeza y el nombre a quien pertenece el rostro. Los métodos indispensables para la detección de rostros se explican en la Tabla 3.3 sin embargo existen más métodos que se encuentran en la documentación del módulo.

Tabla 3.3.

Métodos del módulo detección de rostros

Método	Descripción
getLearnedFacesList	Proporciona una lista de los nombres de rostros que están guardados.
learnFace(Nombre)	Guarda el rostro que detecta asociándolo con un nombre
forgetPerson(Nombre)	Elimina los registros de una persona específica.
clearDatabase	Elimina todos los registros de todas las personas

3.1.6 Reconocimiento de Voz

El módulo ALSpeechRecognition le da al robot la habilidad de reconocer palabras predefinidas o frases en varios lenguajes, se basa en tecnologías proporcionadas por el grupo ACAPELA para NAO V3 y NUANCE para NAO V4.

Antes de ser usado se debe crear una lista de frases o palabras para que el robot las reconozca. Se inicia ALSpeechRecognition, el cual pone una bandera booleana en SpeechDetected para saber si la frase es escuchada.

Si la frase es escuchada se compara y si coincide se activa la bandera WordRecognized. Y la palabra o frase pasa nuevamente a ser comparada en WordRecognizedAndGrammar para saber si coincide con la que escucho el robot. Hay que tener en cuenta las frases más usadas y ponerlas al inicio de la lista.

Programación en Python

El reconocimiento de voz en Python necesariamente necesita un evento de interrupción asociado a una función. Los métodos indispensables para el reconocimiento de voz se explican en la Tabla 3.4, sin embargo, existen más métodos que se encuentran en la documentación del módulo.

Tabla 3.4.

Métodos del módulo reconocimiento de voz

Método	Descripción
getAvailableLanguages	Se obtiene los lenguajes disponibles.
setLanguage("Spanish")	Define el idioma a reconocer
setVocabulary	Ingresa las palabras a reconocer en forma de arreglo
WordRecognized	Evento de interrupción al escuchar una palabra del vocabulario introducido
pause(True/False)	Detiene o Inicia el reconocimiento.

Nota: Métodos principales para el reconocimiento de voz.

Para suscribirse o de suscribirse al evento de interrupción WordRecognized se necesita la siguiente programación Figura 3.11.

Comando para suscribirse o de suscribirse

```
memory.unsubscribeToEvent("WordRecognized", "humanEventWatcher")
memory.subscribeToEvent("WordRecognized",
                        "humanEventWatcher",
                        "RecActFeliz")
```

Figura 3.11. Este comando suscribe a la interrupción.

Una vez suscrito esperara a escuchar la palabra reconocida Figura 3.12.

Función asociada al reconocimiento de voz

```
def RecoVoz(self, key, value, msg):
    """
    Rutina por Interrupcion para detectar el reconocimiento de voz
    """
    global tts, postureProxy, conteo, motion, asr, aup, memory, basic_awareness

    if(len(value) > 1 and value[1] >= 0.55):
        print " Palabra Reconocida", str(value[1])
        print " Palabra Reconocida", str(value[0])
        if(value[0] == "<...> si <...>"):
            print "si"
            memory.unsubscribeToEvent("WordRecognized", "humanEventWatcher")

        if(value[0] == "<...> no <...>"):
            asr.pause(True)
            print "no"
            postureProxy.goToPosture("Stand", 0.5)
            memory.unsubscribeToEvent("WordRecognized", "humanEventWatcher")
```

Figura 3.12. Clasifica las acciones según la palabra reconocida.

Para detener el reconocimiento usualmente se puede utilizar un sensor de tacto o vía interfaz gráfica.

3.1.7 Sensores de Tacto

Este módulo se llama ALTouch, pertenece a los sensores de NAOqi su función es la crear eventos de interrupción para cada sensor de tacto, los cuales están ubicados en las manos, piernas, pecho y cabeza del robot Tabla 3.5. En la cabeza están ubicados tres sensores, en cada mano posee tres sensores ubicados alrededor de la mano y en los pies dos sensores en la parte frontal.

Tabla 3.5.

Nombres de los sensores

FrontTactilTouched	MiddleTactilTouched	RearTactilTouched
HandRightBackTouched	HandRightLeftTouched	HandRightRightTouched
HandLeftBackTouched	HandLeftLeftTouched	HandLeftRightTouched
RightBumperPressed		LeftBumperPressed

Nota: Nombres de los eventos para las interrupciones.

Programación en Python

La programación del módulo se la realiza por evento de interrupción, un mismo evento puede contener diferentes funciones asociadas con diferente programación, pero la llamada a la misma clase Figura 3.13.

```
Suscribirse a un evento interrupción de sensor de tacto

memory.subscribeToEvent("MiddleTactilTouched",
                        "humanEventWatcher",
                        "SeguirInicio")

def SeguirInicio(self, val, pedro):
    """
    Inicio cunado se encienda y le presentamos al adulto mayor
    """
    global memory, motion, managerProxy, postureProxy, basic_awareness
    ### Desuscribirse
    memory.unsubscribeToEvent("MiddleTactilTouched", "humanEventWatcher")
    print "Inicio midle tactil"
    print "Interrupcion "
```

Figura 3.13. Suscribirse al sensor MiddleTactilTouched y función asociada.

Se necesita que al inicio de la función asociada de-suscribirse del evento para que corra una solo vez.

3.1.8 Rastreo de Rostros

Este módulo se llama ALBasicAwareness, pertenece a la percepción de personas de NAOqi, su función es la de establecer y mantener contacto visual con las personas. El módulo de rastreos de rostros funciona a base de estímulos que puede ser un sonido, detección de movimientos, aparece un rostro en la cámara y cuando se toca un sensor de tacto. Existen tres tipos de modos de enlace con una persona.

- El modo sin enlace permite enlazarse con otra persona por cualquier estímulo.
- Parcialmente enlazado permite observar los estímulos y regresar a la persona que primero detecto.
- Totalmente enlazado no presta atención a ningún estímulo solo se centra en la primera persona que detecto.

Los métodos indispensables para rastrear rostros se explican en la Tabla 3.6, sin embargo, existen más métodos que se encuentran en la documentación del módulo.

Tabla 3.6.

Métodos del módulo rastreo de rostros

Método	Descripción
setEngagementMode	Se configura el modo de enlace No enlazado / Parcialmente enlazado / Totalmente enlazado
setTrackingMode	El modo que el robot seguirá a esa persona.
startAwareness	Inicio del método de rastreo de rostros
stopAwareness	Detener el método de rastreo de rostros

Nota: Métodos principales para el rastreo de rostros.

Programación en Python

En Python se debe conectar con el módulo ALBasicAwareness y utilizar los métodos de configuración del tipo de enlace y el modo de rastreo Figura 3.14.

```
Programación para el módulo rastrear rostros

from naoqi import ALProxy

ip_robot = "192.168.1.10"
port_robot = 9559
basic_awareness = ALProxy("ALBasicAwareness", ip_robot, port_robot)
basic_awareness.setEngagementMode("FullyEngaged")
basic_awareness.setTrackingMode("Head")

basic_awareness.stopAwareness()
basic_awareness.startAwareness()
```

Figura 3.14. Inicio y conexión con el modulo.

Es necesario reiniciar el rastreo de rostros, cuando se cambia de actividad debido a que se detiene automáticamente. Para reiniciar se debe detener y correr nuevamente el modulo.

3.1.9 Obtener Fecha

Son dos librerías las necesarias para obtener la fecha, se llaman: datetime y locale, pertenecen a Python, su función es la de obtener el tiempo y establecer el formato basándose en la hora configurada del computador.

Programación en Python

Se necesita importar ambas librerías para extraer los datos de la fecha. Los comandos para obtener la fecha en nuestra aplicación se muestran en Tabla 3.7, sin embargo, existen más comandos que se encuentran en la documentación de la librería.

Tabla 3.7.

Comandos para obtener la fecha

Comando	Descripción
<code>datetime.date.today().strftime("%A")</code>	Día del semana en palabra
<code>datetime.date.today().strftime("%d")</code>	Número del día
<code>datetime.date.today().strftime("%B")</code>	Mes del año en palabra
<code>datetime.date.today().strftime("%Y")</code>	Año en numero

La programación se realiza de la siguiente manera Figura 3.15.

```
Programación para obtener la fecha

import datetime, locale
###Configuración del formato en español
locale.setlocale(locale.LC_ALL, 'Spanish_Spain.1252')

diasemana = datetime.date.today().strftime("%A")
dianumero = datetime.date.today().strftime("%d")
mes = datetime.date.today().strftime("%B")
anio = datetime.date.today().strftime("%Y")
```

Figura 3.15. Programación en comandos en Python para obtener la fecha.

Esta es una librería de Python por lo tanto no es un módulo de NAOqi.

3.1.10 Enviar e-mail

Para enviar e-mails desde Python, usamos la librería `smtplib` y la librería `email`, con la configuración del servicio POP/IMAP habilitados en la cuenta de correo electrónico. La librería `email` de Python permite leer y enviar e-mails sencillos, y de una forma más compleja se utiliza mensajes MIME, con los cuales se puede enviar archivos adjuntos como imágenes o documentos. Por otro lado, la librería `smtplib`, se usa para enviar e-mails en formato HTML por medio del protocolo para transferencia simple de correo.

Programación en Python

Se necesita importar ambas librerías para enviar un mail desde Python Figura 3.16. La información adicional para enviar un mail desde Python es el puerto 587 y dirección (SMTP) de Hotmail smtp-mail.outlook.com.

Programación en Python

```
mailServer.ehlo()
mailServer.starttls()
mailServer.ehlo()
mailServer.login(MailNAO, Contraseña)
mailServer.sendmail(MailNAO, Destinatario, msg.as_string())
mailServer.close()
```

Figura 3.16. Programación para enviar un mail por SMTP.

Usualmente toma entre un minuto en enviar el mail.

3.1.11 Guardar Datos

La librería de Python que se utilizó para guardar datos fue pickle, la cual funciona para serializar y de-serializar objetos. En esencial se guardó diccionarios y listas como la información de cada adulto mayor y las rutinas que se ejecutan el robot para el análisis de los resultados.

Programación en Python

La programación para guardar un diccionario o una lista en un archivo llamado save.pkl es la siguiente Figura 3.17

Programación para guardar datos

```
import pickle

dict2 = {'Esteban': ['Valladares', 'mail_Adulto.com', '3/10/2000']}
list1 = [100, 100, 100]
print dict2
output = open("save.pkl", 'wb')
pickle.dump(dict1, output)
pickle.dump(list1, output)
output.close()
```

Figura 3.17. Guardar un diccionario y una lista.

La programación para extraer datos del archivo save.pkl es la siguiente Figura 3.18.

Programación para extraer datos

```
inputFile = open("save.pkl", 'rb')
person = pickle.load(inputFile)
Lista = pickle.load(inputFile)
inputFile.close()
```

Figura 3.18. Extraer el diccionario y la lista.

El archivo save.pkl se almacena en la carpeta que se corre el programa.

3.1.12 Leds

Este módulo se llama ALLeds, pertenece a los sensores de NAOqi su función es la de controlar los leds en las orejas que son azules, los del pecho, ojos y pies que son RGB. Los colores que se programan en los ojos en conjunto con los movimientos es la principal herramienta para mejorar la interacción humano robot ya que a través de estos se expresa emociones en el robot. Los leds de los oídos y de la parte superior de la cabeza del robot NAO, son de color azul, a diferencia de los leds de sus ojos, pecho, y pies que son RGB, con lo cual es posible combinarlos para obtener diferentes colores en dichas partes del robot. A si también cómo es posible controlar la intensidad de los leds entre 0 al 100%.

Programación en Python

Se necesita conectar el módulo para empezar a utilizar los métodos. Con la siguiente programación se realiza la conexión y el manejo Figura 3.19.

Conexión y manejo

```
from naoqi import ALProxy

leds = ALProxy("ALLeds", ip_robot, port_robot)
leds.post.rotateEyes(0xFF0000, 2.0, 4.0)
leds.setIntensity("RightFaceLedsRed", 0.9)
leds.setIntensity("LeftFaceLedsRed", 0.9)
```

Figura 3.19. Programación del módulo ALLeds.

Se ajusta la intensidad y se coloca el color en hexadecimal.

3.1.13 Reconocimiento de Objetos

El robot NAO tiene la capacidad de reconocer diferentes imágenes, los lados de un objeto y lugares previamente guardados. Este módulo se llama ALVisionRecognition. Este módulo basa su funcionamiento en el reconocimiento visual de puntos clave y

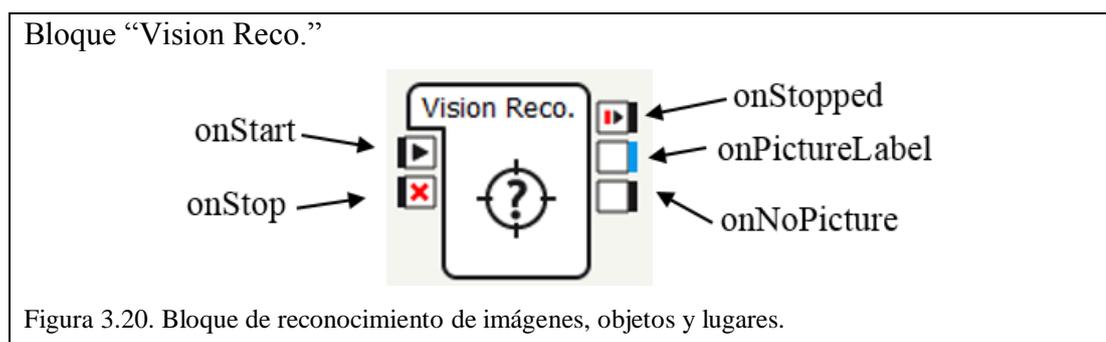
está destinado a reconocer objetos específicos que hayan sido guardados previamente. Los cuales con ayuda de Choregraphe se pueden guardar imágenes en alrededor 30 segundos. Hay q tener en cuenta que solo puede guardar imágenes o superficies que no tengan textura es decir en un plano de dos dimensiones, en su mayoría reconoce figuras.

Programación en Choregraphe

Permite reconocer imágenes, objetos y lugares que hayan sido previamente guardados en el robot a través de Video Monitor de Choregraphe.

Bloque “Vision Reco.”

Este bloque contiene dos entradas y tres salidas Figura 3.20. En la primera entrada onStart que es de tipo bang debe ingresar una señal para iniciar el proceso de reconocimiento de imágenes. En la segunda entrada la cual es onStop al momento que recibe una señal de tipo bang se detendrá el reconocimiento de imágenes. La salida onStopped envía un pulso de tipo bang cuando se termina el proceso de reconocimiento de imágenes.



En la segunda salida onPictureLabel devuelve un string con el nombre de la imagen reconocida, si es el reconocimiento se desarrolló correctamente. Y en la tercera salida onNoPicture de tipo booleana enviará una señal cuando no haya reconocido la imagen.

3.1.14 Obtener la Imagen de la Cámara

Este módulo se llama ALVideoDevice, pertenece a visión de NAOqi su función es la de proveer imágenes de las cámaras a todos los módulos de procesamiento de visión artificial. Se encarga de las configuraciones de las imágenes de acuerdo a las especificaciones técnicas de las cámaras Tabla 3.8.

Tabla 3.8.

Resoluciones, colores y frames por segundo

Resolución	Colores	Frames por segundo
4VGA (1280 * 960)	YUV422	30 fps
VGA (640 * 480)	YUV 24 bits	
QVGA (320 * 240)	Y 8 bits	
QQVGA (160 * 120)	RGB 24 bits	
QQQVGA (80 * 60)	BGR 24 bits	
QQQQVGA (40 * 30)	HSY 24 bits	

Nota: Especificaciones técnicas que soporta la cámara.

El módulo permite obtener las imágenes de la cámara en tiempo real a la computadora.

Programación en Python

Para obtener las imágenes es necesario registrarse y establecer las configuraciones Figura 3.21.

Registro y configuración

```
def _registerImageClient(self, IP, PORT):
    """
    Register our video module to the robot.
    """
    self._videoProxy = ALProxy("ALVideoDevice", IP, PORT)
    resolution = vision_definitions.kVGA # 320 * 240
    colorSpace = vision_definitions.kRGBColorSpace
    self._imgClient = self._videoProxy.subscribe("_client",
                                                resolution,
                                                colorSpace,
                                                5)

    # Select camera.
    self._videoProxy.setParam(vision_definitions.kCameraSelectID,
                              self._cameraID)
```

Figura 3.21. Configuraciones iniciales y conexión al módulo.

La segunda función es obtener la imagen Figura 3.22.

Función obtener imagen

```
def obteneri(self):|
    self._alImage = self._videoProxy.getImageRemote(self._imgClient)
    #####33333
    heigth = self._alImage[1]
    width = self._alImage[0]
    #####
    self._image = (numpy.reshape(numpy.frombuffer(self._alImage[6],
                                                    dtype='%iuint8' % self._alImage[2]),
                                (self._alImage[1],
                                 self._alImage[0],
                                 self._alImage[2])))

    return img, heigth, width
```

Figura 3.22. Función para obtener una imagen.

Para obtener una imagen es necesario observar que formato se necesita, en la aplicación se requirió un tamaño de 8 bits para los datos de la imagen. La cantidad máxima de imágenes que se puede extraer es 30 por segundo.

3.2 Planteamiento de la Aplicación

La aplicación desarrollada está enfocada en acercarse a una interacción natural entre el adulto mayor y el robot NAO, utilizando las capacidades de hardware y software del robot para entretener al adulto mayor. El sistema inicia con el registro de datos personales y rostro de los adultos mayores a través de una interfaz gráfica realizada en wxPython. Luego se procede a reconocer la expresión facial del adulto mayor para realizar actividades como frases motivacionales, reproducción de música, chistes, historias, bailes, ejercicios físicos y mentales como: reconocimiento de figuras. Estas actividades serán evaluadas verificando la expresión facial antes y después de una actividad, una rutina está compuesta de tres actividades con el adulto mayor después de una rutina se almacenan las actividades y expresiones faciales para analizar los resultados, notificándolos de forma local y por medio de correo electrónico al personal encargado.

3.3 Programación de la Aplicación

La programación de la aplicación se realiza por etapas las cuales contiene uno o más módulos de NAOqi para su funcionamiento.

3.3.1 Presentación Inicial

La aplicación inicia con una presentación de parte de los desarrolladores o personal a cargo, que colocara el NAO en una posición cercana al adulto mayor en el rango de 1.2 m de distancia, superior a esta distancia no es funcional el reconocimiento facial y de expresiones. Para esto utilizamos el behavior “follow-me” descargado de aldebaran el cual permite llevar al robot de la mano hasta el adulto mayor. Para ver el diagrama de flujo referirse al ANEXO 1. Los módulos utilizados: ALBehaviorManager.

3.3.2 Iniciación de la Rutina

Para empezar la rutina se inicia con el sensor de la cabeza MiddleTactilTouched cual en su función asociada detiene el behavior 'follow-me', posiciona al robot en posición de pie e inicia el rastreo de rostros. Los módulos utilizados: ALBehaviorManager, ALTouch, ALBasicAwareness.

3.3.3 Introducción Saludo

Cuando se encuentra un rostro se procede a rastrearlo para obtener la expresión facial, si es una expresión valida empieza la identificación del rostro. Cuando se obtiene la identificación del rostro y la expresión facial, se inicia con un saludo verbal más la fecha del día en curso, acompañado de un gesto de movimiento de mano.

De la siguiente manera:

¡Hola soy NAO! he venido a acompañarte

Hoy es “día en palabra" "día número" de "mes" de el "año", en la capital Quito Ecuador

Módulos utilizados: ALMotion, ALFaceCharacteristics, ALFaceDetection, datatime, locale y ALTextToSpeech.

Para ver el diagrama de flujo referirse al ANEXO 2.

3.3.4 Clasificación de las Actividades según las Expresiones Faciales

Cada actividad está relacionada con una expresión facial que se muestra en la Tabla 3.9.

Tabla 3.9.

Clasificación de las actividades

Expresión / Actividades	Canciones	Baile	Chiste	Reconocer	Ejercicios	Historia	Frase
Feliz	✓	✓	✓				
Sorprendido		✓		✓		✓	
Tranquilo			✓		✓	✓	
Enojado	✓	✓	✓				
Triste	✓			✓			✓

Nota: Clasificación de las actividades según la expresión fácil.

Son tres canciones, tres bailes, tres chistes, tres frases, dos historias, un ejercicio físico y un reconocimiento de figuras. Cada expresión facial que reconoce se expresa con un behavior y de forma verbal por el robot, por último se coloca al robot posición de pie para dar inicio a las actividades. Para ver el diagrama de flujo referirse al ANEXO 3. Los módulos utilizados: ALBehaviorManager y ALTextToSpeech.

3.3.5 Actividad Chistes

Los chistes empiezan y terminan con un behavior de risa, tienen temáticas de militares y de parejas los cuales necesitan la variación de la entonación de voz para lograr voz de relator, de hombre y de mujer. Para ver el diagrama de flujo referirse al ANEXO 7. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech.

3.3.6 Actividad Frases

Las frases motivacionales tienen relación con el adulto mayor. La actividad inicia con behavior de explicación y al final se coloca al robot en posición de pie. Para ver el diagrama de flujo referirse al ANEXO 8. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech.

3.3.7 Actividad Reproducción de Música

Las canciones son de la época acorde al adulto mayor la cuales se guardaron en el robot NAO a través de FileZilla. Se reproducen acompañadas del movimiento de tocar guitarra y al final se coloca al robot en posición de pie. Para ver el diagrama de flujo referirse al ANEXO 6. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech, ALMotion.

3.3.8 Actividad Historias

Las historias contienen temas nacionales de Cantuña y biografía de Eugenio Espejo de las cuales se realizan preguntas. Cada pregunta tiene como respuesta una palabra y la función de saltar la pregunta, pensando en no complicar al adulto mayor, por último se notifica cuantos aciertos se obtuvieron. Para ver el diagrama de flujo referirse al ANEXO 8. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech, ALMotion.

3.3.9 Actividad Bailes

Los bailes son thriller, funky town y macarena que están disponible en internet para Choregraphe (University of Notre Dame, 2016). Los cuales modificamos y exportamos para su uso en Python, se finaliza con una venia en agradecimiento y se coloca en posición de pie. Para ver el diagrama de flujo referirse al ANEXO 6. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech, ALMotion.

3.3.10 Actividad Reconocimiento de Figuras

Para la actividad de reconocimiento de figuras, se realizó en Choregraphe, ya que debido a su versatilidad, permite enseñarle al robot a reconocer las figuras que nosotros deseamos y guardarlas en menos de 30 segundos directamente en la base de datos del robot. Para enseñarle al robot las figuras que debe reconocer hay que abrir el Video Monitor y pulsar en el icono de enseñar .

Una vez completada la sesión de enseñanza hay que guardar la base de datos, dentro del computador para tenerla como respaldo, nuevamente eso se hace haciendo clic en el icono de “exportar base de datos de reconocimiento” .



Las figuras utilizadas en esta prueba fueron un tanto complejas y con detalles como por ejemplo las imágenes: virgen del panecillo, Mario Moreno “Cantinflas”, Ramón Valdez “Don Ramón”, la torre Eifel, símbolo de dinero y una casa, ya que si son figuras muy simples Choregraphe no permite que estas sean aprendidas por el robot porque no tiene mayor detalle como figuras geométricas, flechas y estrellas Figura 3.23. Para ver el diagrama de flujo referirse al ANEXO 7.

3.3.11 Actividad Ejercicios

Los ejercicios son creados en Choregraphe con movimientos de cabeza y brazos con una interpolación simple por ángulos. Se inicia con un behavior de vamos a mover el cuerpo y se dicta las instrucciones para realizar los ejercicios, a continuación, empiezan los ejercicios y al final se coloca al robot en posición de pie. Para ver el diagrama de flujo referirse al ANEXO 8. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech, ALMotion.

3.3.12 Notificaciones y Guardar Datos

Conforme se desarrolla la rutina se recolecta la información de las expresiones faciales, actividades, fecha e identificación del adulto mayor para guardarlas en un archivo. Al final de la rutina se analiza si el adulto mayor ha tenido más de una expresión facial negativa. Las expresiones faciales negativas son las de tristeza y enojo.

Con la información obtenida se emite dos notificaciones una vía e-mail y la otra en el robot. La notificación vía e-mail contiene la información de las expresiones faciales,

actividades, fecha y si el adulto mayor ha tenido más de una expresión facial negativa. La notificación en el robot se accede con el sensor de tacto de la cabeza MiddleTactilTouched el cual emite una advertencia si ha tenido más de una expresión facial negativa colocando los leds de los ojos con un color rojo o si ha tenido expresiones faciales positivas le indica que se encuentra bien. Para ver el diagrama de flujo referirse al ANEXO 9. Los módulos utilizados: ALBehaviorManager, ALTextToSpeech, ALMotion.

CAPÍTULO 4

ANÁLISIS DE RESULTADOS

En este capítulo se presentan las pruebas y los análisis realizados a la aplicación de asistencia al adulto mayor comprobando la efectividad en el reconocimiento de expresiones faciales, reconocimiento de voz y la expresión facial mostrada al realizar una actividad, de esta manera se verifica el óptimo funcionamiento de la aplicación de asistencia al adulto mayor.

4.1 Pruebas y análisis en el reconocimiento de expresiones faciales

El reconocimiento de expresiones faciales es el enfoque principal para realizar la aplicación de asistencia al adulto mayor, es necesario hacer pruebas de la efectividad que posee. Para esto se utiliza un set de datos que contienen imágenes de adultos mayores mostrando una expresión facial definida como: Feliz, triste, sorprendido, enojado y neutral. El set de datos fue creado para tareas de investigación, el cual contiene imágenes con las expresiones faciales de personas jóvenes, mediana edad y adultos mayores. (Max Planck Institute for Human Development, 2005). Para las pruebas se tomó a 10 adultos mayores del set de datos, el cual posee dos imágenes para cada expresión. En total se probó 20 imágenes por cada expresión facial.

Para empezar las pruebas de análisis de expresiones faciales es necesario trabajar en conjunto con el módulo de rastreo de rostros ya que una vez confirmado un rostro se procede a analizar la expresión facial. El robot analiza cada imagen presentada del set de datos y determina la expresión facial. Al final se compara la expresión facial determinada por el robot con la expresión definida en el set de datos para determinar la efectividad en el reconocimiento.

Las tablas de datos obtenidos en las pruebas están organizadas por expresiones faciales para enojado el **ANEXO 10**, feliz el **ANEXO 11**, neutral el **ANEXO 12**, asombrado el **ANEXO 13** y triste el **ANEXO 14**.

Cada tabla de datos contiene la identificación del adulto mayor, la expresión facial ya definida en el set de datos, el vector análisis con los valores para cada expresión y la expresión facial analizada.

El porcentaje de aciertos para cada expresión se calcula con la siguiente formula

$$\% \text{ Aciertos} = \frac{\text{Cantidad de Aciertos}}{\text{Numero de Pruebas}} \times 100 \quad (4-1)$$

El porcentaje de error para cada expresión se calcula con la siguiente formula

$$\% \text{ Error} = \frac{\text{Cantidad de errores}}{\text{Numero de Pruebas}} \times 100 \quad (4-2)$$

El análisis y resultados de las expresiones faciales se presentan en la Tabla 4.1.

Tabla 4.1.

Análisis de las expresiones faciales

Expresión Facial	Cantidad de Pruebas	Aciertos	Errores	Porcentaje de Aciertos	Porcentaje de error
Feliz	20	20	0	100 %	0 %
Neutral	20	18	2	90 %	10 %
Enojado	20	19	1	95 %	5 %
Triste	20	14	6	70 %	30 %
Sorprendido	20	18	2	90 %	10 %

Los datos obtenidos indican que la expresión facial feliz es fácilmente identificable ya que los rasgos difieren de las demás expresiones, a diferencia de las expresiones neutral y triste que comparten rasgos entre sí lo cual dificulta interpretar la expresión, es por esto que se obtiene un grado de acierto para triste del 70 % y para neutral del 90 %.

4.2 Pruebas y análisis del reconocimiento de voz

La capacidad que posee el robot en el reconocimiento de voz afecta directamente a la correcta ejecución de la aplicación, ya que dificulta la interacción entre el adulto mayor y el robot, debido que los comandos por voz son parte esencial para el desempeño y desarrollo del mismo al momento de seleccionar actividades.

Si una palabra no es reconocida, la aplicación no continua a la siguiente etapa, la cual es la ejecución de la actividad escogida, para las pruebas se delimitó una distancia de 0- 100 cm, divididos en secciones de 25cm cada una, haciendo pruebas de ensayo y

error, también ubicando al robot NAO a diferentes alturas como al nivel del piso (0 cm), en una superficie a 50 - 100 cm sobre el nivel del piso dependiendo de la estatura del adulto mayor, mostrado en la Figura 4.1.

Prueba de la actividad y reconocimiento de voz.



Figura 4.1. Prueba de distancias óptimas para comandos de voz.

Para las pruebas con el robot a nivel del piso (0 cm), el adulto mayor media entre 130 – 150 cm. Siendo los sujetos de prueba de mayor tamaño, se ubicó al robot a mayor altura para probar si reconocía los comandos de voz.

Pruebas de reconocimiento de voz.



Figura 4.2. Prueba de comandos de voz en el piso y sobre una tarima.

Se determinó que la posición ideal para trabajar con el robot era ubicarlo en el piso y que la persona este sentada en una silla, de este modo el tamaño de la persona influía mínimamente en el reconocimiento voz, como se muestra en la Figura 4.2.

En la Tabla 4.2, se puede observar el error promedio obtenido de cada una de las pruebas realizadas a los adultos mayores usando la fórmula de la ecuación 4.2, conjuntamente con los comandos de voz dictados por ellos para seleccionar cada una de las actividades, teniendo como resultado más alto un error del 34,97% en el

comando “Canción”. Para mirar los resultados obtenido en las pruebas y sus respectivos errores con cada adulto mayor dirigirse a los **ANEXO 15 - ANEXO 25**.

Tabla 4.2.

Tabla de porcentaje del Error Promedio

Frase	% Error Promedio	Frase	% Error Promedio
Baile	9,29	Historia	9,52
Canción	34,97	Luzbel	16,67
Chiste	19,72	Quito	20,00
Ejercicio	25,00	Reconocer Figuras	30,21
Emancipadoras	33,33		
Frase	28,45	San Francisco	16,67

4.3 Prueba de la aplicación

Se realizó las pruebas de la aplicación con diez adultos mayores con los cuales se demostró la capacidad de interacción alcanzada con el robot. La aplicación inicia con guardar el rostro en la base de datos del robot a través de la aplicación creada en WxPython como se muestra en la Figura 4.3.



En la aplicación se guarda el nombre, el apellido, fecha de nacimiento y el correo electrónico de contacto. Se continúa con la aplicación que consiste en las actividades desarrolladas las cuales se almacenan en un archivo con formato pickle.

Por último, se finaliza con el envío del correo electrónico a la persona de contacto mostrado en la Figura 4.4.



Figura 4.4. Envío de un correo electrónico con los datos de la aplicación.

La persona de contacto es un familiar el cual se informa de las actividades y expresión facial del adulto mayor.

CONCLUSIONES

- Con la información investigada sobre las necesidades del adulto mayor se determinó, que no solo se busca alcanzar mejoras en un sentido médico, sino lograr un sentimiento de bienestar a través de cualquier actividad que les haga sentir útiles y autosuficientes, que alivie la tensión psicológica que genera la vejez
- Se logró obtener el análisis de expresiones faciales a través de la programación en lenguaje Python con el funcionamiento de los módulos de rastreo de personas y análisis de características, se obtuvo el vector de análisis de expresiones el cual contiene cinco valores que sumados dan un total de uno, el cual se procesó con un algoritmo que determina el tipo de expresión según el valor más alto entre ellos.
- Para cubrir las necesidades del adulto mayor se programó y diseñó actividades que se basan en terapias de reminiscencia, orientación a la realidad y lúdicas. Las actividades desarrolladas son bailes, canciones, saludo inicial, chistes, reconocimiento de figuras e historias. Con esto se demostró que se pueden realizar terapias con las capacidades del robot NAO.
- Para notificar el estado del adulto mayor se desarrolló un algoritmo que permite guardar datos personales, expresiones faciales y las actividades desarrolladas con el robot, se crea un archivo de texto plano y se envía por correo electrónico al titular o persona encargada del adulto. Además, se informan los resultados al tocar la cabeza del robot.
- Para saber la efectividad del reconocimiento de las expresiones faciales de tristeza, felicidad, asombro, enojo y tranquilidad, fue necesario recurrir a una base de datos de imágenes ya establecida, se utilizó un set de datos del Instituto Max Planck Para El Desarrollo Humano, que contiene imágenes de adultos mayores con expresiones faciales.

- Con los datos analizados de las pruebas de efectividad del reconocimiento de expresiones faciales se determinó que la felicidad y el enojo son fácilmente identificables con un 100% y 95% de aciertos respectivamente, ya que los rasgos del rostro difieren notoriamente de las demás expresiones y son claramente diferentes una de la otra. Para el caso de sorprendido y neutral comparten rasgos comunes que dificultan su reconocimiento alcanzando un 90% de aciertos ambos casos. Analizando el caso de tristeza posee diferencias muy sutiles con la de enojo por lo cual confunde estas expresiones mostrando un 70% de aciertos.
- Se propuso realizar la interfaz gráfica que permitía guardar rostros y datos personales del adulto mayor en PyQt, pero debido a complicaciones al crear ventanas y al exportar video de las cámaras se optó por utilizar wxPython, librería gráfica con la cual se pudo manejar óptimamente ventanas y cuadros de diálogos, además que se consiguió exportar exitosamente el video de las cámaras del robot e incluso aplicar algunas librerías de OpenCV para identificar rostros.
- Al exportar movimientos desde Choregraphe a Python, se determinó que para mover una o dos articulaciones se lo puede realizar por interpolación de ángulos simple, pero para movimientos de todo el cuerpo es necesaria la curva de Bezier que permite que el robot mueva sus articulaciones por todos los puntos trazados hasta llegar de una forma precisa a su posición final. Si se realiza un movimiento completo por interpolación simple no se llega al punto deseado, y se puede producir un bloqueo de las articulaciones, por movimientos de contorsión no analizados en la trayectoria.

RECOMENDACIONES

- Para realizar aplicaciones que contengan reconocimiento de voz es recomendable que la carga mínima sea 80% de la batería. Caso contrario el reconocimiento de las palabras no es exitoso y en el peor de los casos ningún comando de voz es reconocido por el robot, debido a que la carga restante de las baterías tiene prioridad para el procesador, cámara y los motores en general.
- Al desarrollar aplicaciones que necesiten exportar video en tiempo real de la cámara del robot al ordenador, es recomendable tener en cuenta el tamaño de la imagen y el medio que se va transmitir sea cableada o inalámbrica necesitaran diferentes tamaños de imagen para que sea fluida, además es necesario tener en cuenta que el formato de las imágenes sea compatible con la librería que se utiliza. Para la aplicación se utilizó una calidad KVGA de 640 x 480 por un medio cableado.
- Para aplicaciones complejas o extensas con el robot NAO es recomendable ejecutarlas desde la computadora, debido a que estas consumen muchos recursos de procesamiento e incrementan el consumo de la batería.
- Para instalar correctamente el software Python, hay que cerciorarse que sea la versión 2.7 de 32 bits, la cual es compatible con las bibliotecas disponibles del robot NAO no una versión posterior, debido que NAO corre en una plataforma Linux, es mejor instalar todos los softwares para su programación en un computador con Linux.

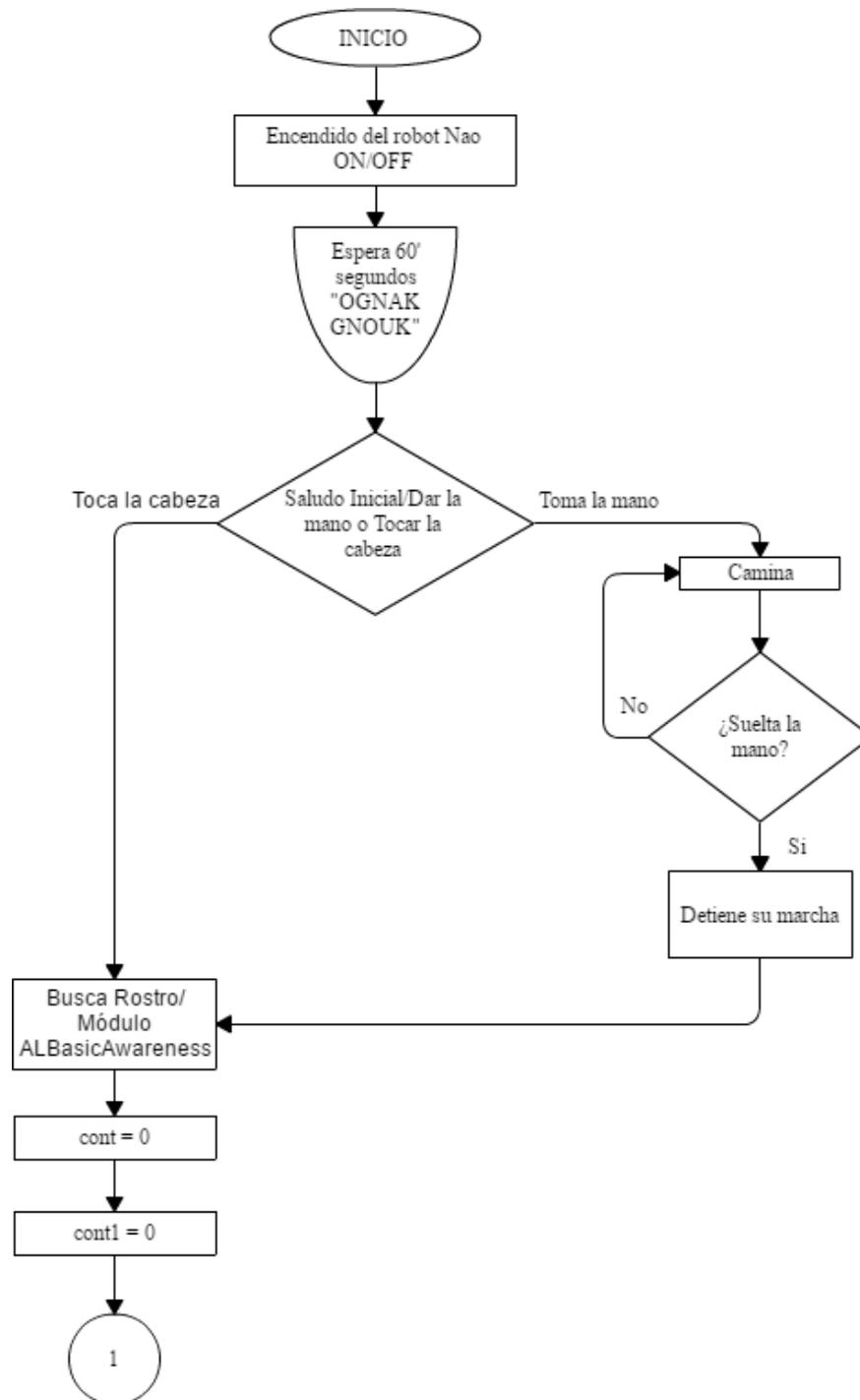
LISTA DE REFERENCIAS

- Baynes, S., Saxby, P., & Ehlert, K. (1987). Reality Orientation and reminiscence Therapy. A controlled cross-over study of elderly confused People. *British Journal of psychiatry*, 151:222-132.
- Black, M., & Yacoob, Y. (1997). Recognizing facial expressions in image sequences using local parameterized models of image motion. *International Journal of Computer Vision*, 23-48.
- Borel, B. (2010). A Ping-Pong-Playing Terminator. *Popular Science*.
- Chicaiza, C. P. (2011). *La terapia ocupacional y el estado de ánimo del adulto mayor en el asilo de ancianos del hospital provincial docente Ambato*. Ambato, Ecuador.
- Cootes, T., Edwards, G., & Taylor, C. (1998). Active Appearance Models. *European conference on computer vision*, 484-498.
- Cortés, F. R. (2011). *Robótica Control de robots manipuladores*. MARCOMBO S.A.
- Darwin, C. (1872). *the expression of the emotions in man and animals*. Londres: John Murray.
- Ekamn, P., & Friesen, W. (1971). Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 124-129.
- German Aerospace Center. (2007). German Aerospace Center. Alemania.
- Henkemans, B., BP, B., & Bosch, V. (2013). Using a robot to personalise health education for children with diabetes type 1: a pilot study. *US National Library of Medicine*, 174-181.
- Honda. (09 de 09 de 2016). World Honda. Tokio, Japón.
- learning, T. (2000). Aistudy.
- Max Planck Institute for Human Development. (2005). FACES. Berlin, Alemania.
- Mehrabian, A. (1972). *Nonverbal Communication*. New Jersey: Aldine.

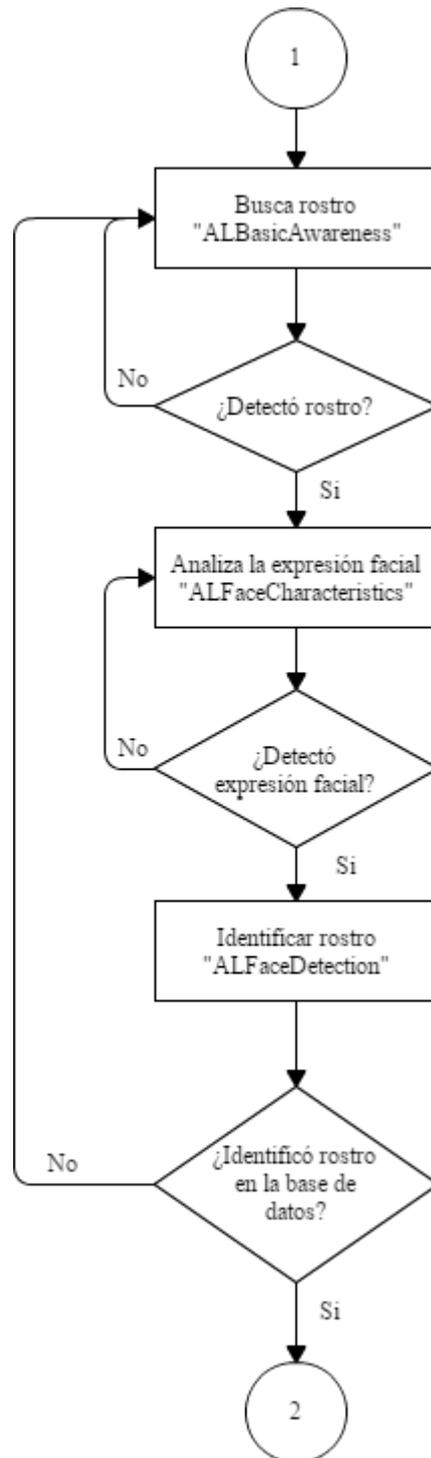
- Pérez, V. (2010). *Solidaridad y calidad de vida en las personas mayores*. Madrid: Dykinson.
- Ribeiro, T., Pereira, A., Deshmukh, A., Aylett, R., & Paiva, A. (2011). I'm the mayor: A Robot Tutor in Enercities-2. *University of Lisbon Protugal*, 1675-1676.
- Riccillo, M. (2012). Robótica. *Serie vida cotidiana y tecnología*, 6.
- Robotics Today HOAP 3. (2005). Robotics Today.
- Robotics Today SDR 3X. (2000). Robotics Today.
- Robotics, A. (27 de 08 de 2015). *Aldebaran Documentation NAO*. Paris.
- Robotics, E. (31 de 03 de 2015). Emotion Robotics.
- Ruiz Munera, A. J. (2005). La Educación Física en la Vejez. *Ponencia presentada en el V congreso Internacional de Pedagogía de la Habana*, 1-12.
- Shamsuddin, S., Yussof, H., Ismail, L., Hanapiah, F., Mohamed, S., & Piah, H. A. (2012). Initial response of Autistic Children in Human-Robot Interaction therapy with Humanoid Robot NAO. *International Colloquium on Signal Processing and its Application*, 188-192.
- Soft Bank Robotics. (2014). Inside Retail. Australia.
- Tecnologías, C. (19 de 05 de 2014). Cuatrorios Tecnologías.
- thefutureofthings. (11 de 09 de 2016). thefutureofthings. Japón.
- University of Notre Dame. (2016). University of Notre Dame.
- University, C. R. (2000). onlineinspeeration.

ANEXOS

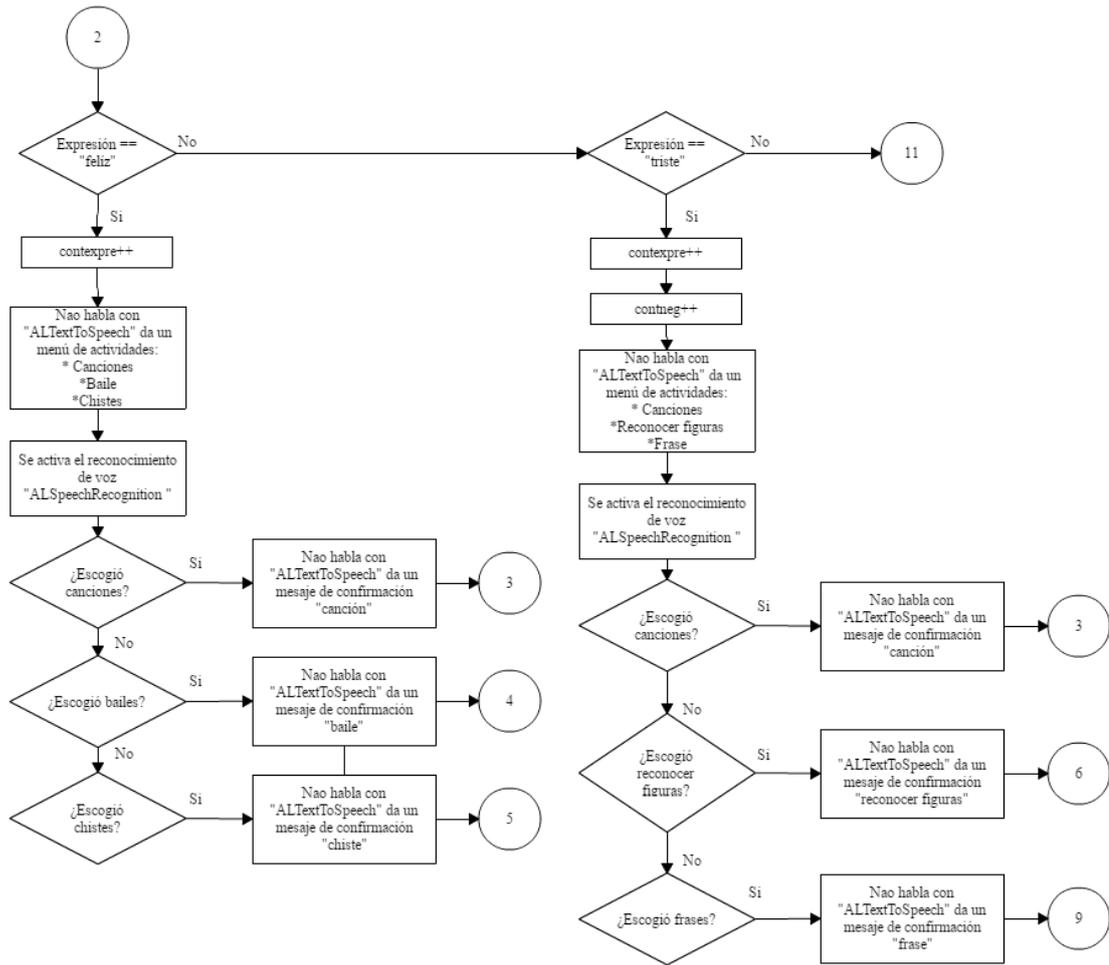
ANEXO 1: Diagrama de flujo inicio aplicación



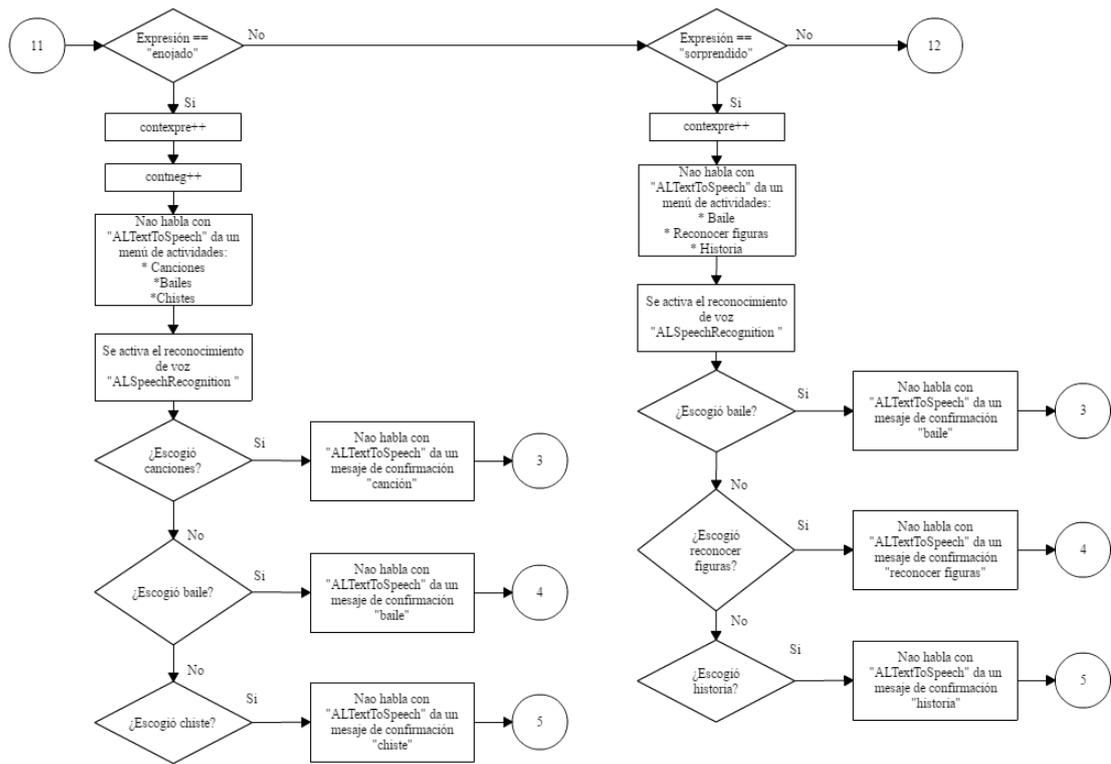
ANEXO 2: Introducción saludo



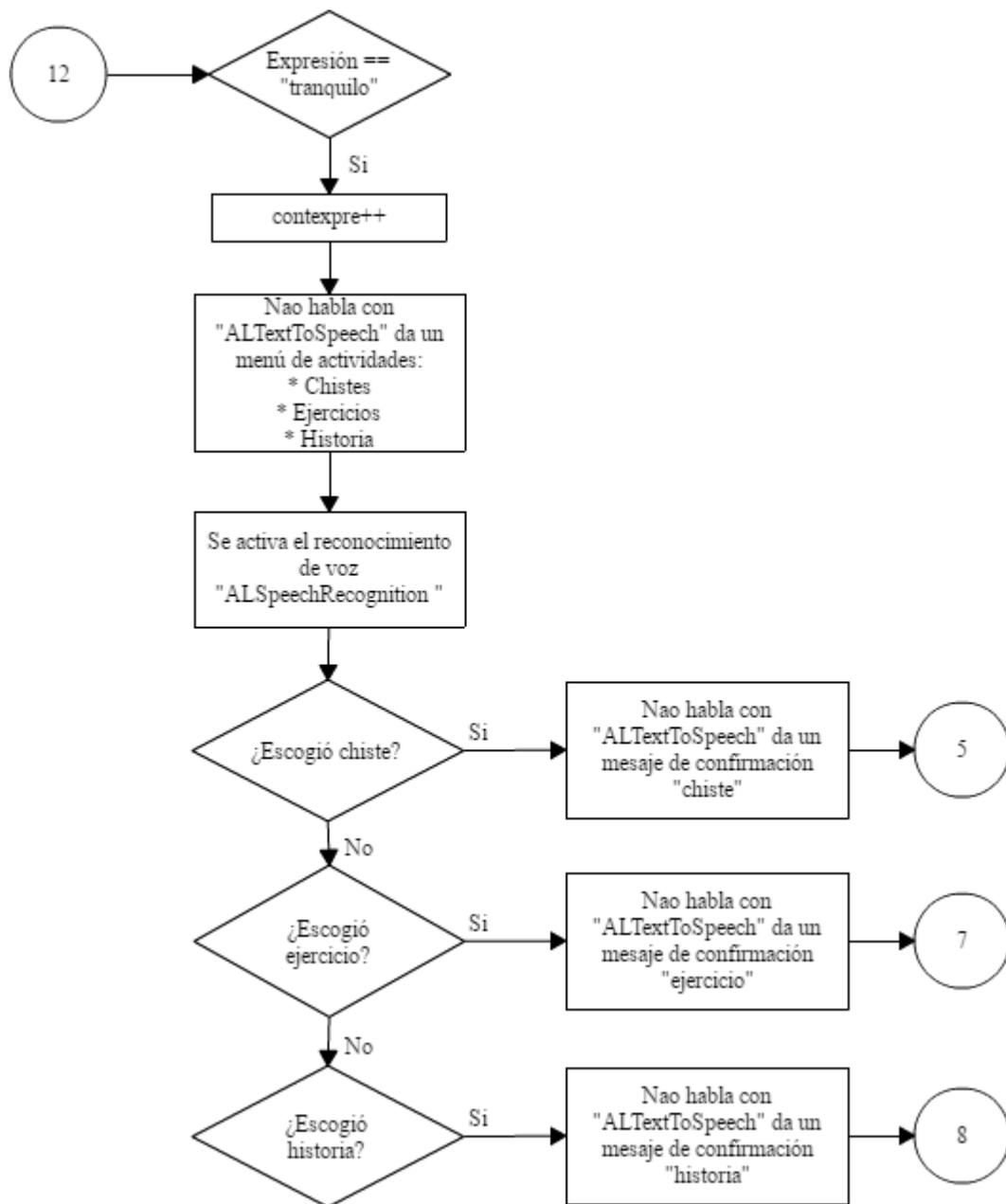
ANEXO 3: Clasificación de las expresiones



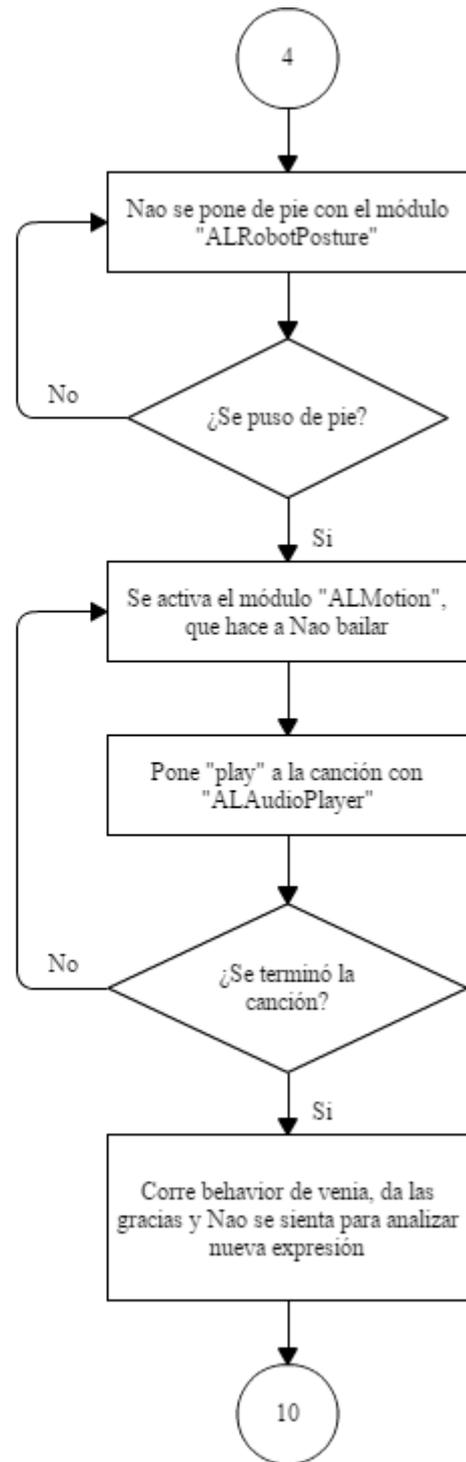
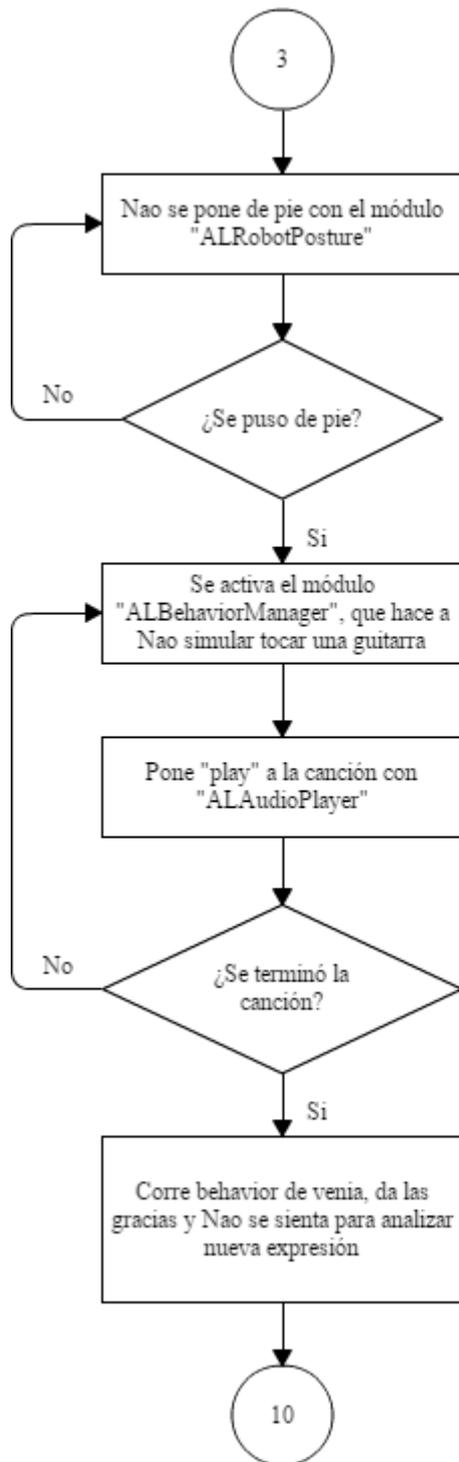
ANEXO 4: Clasificación de las actividades enojado y sorprendido



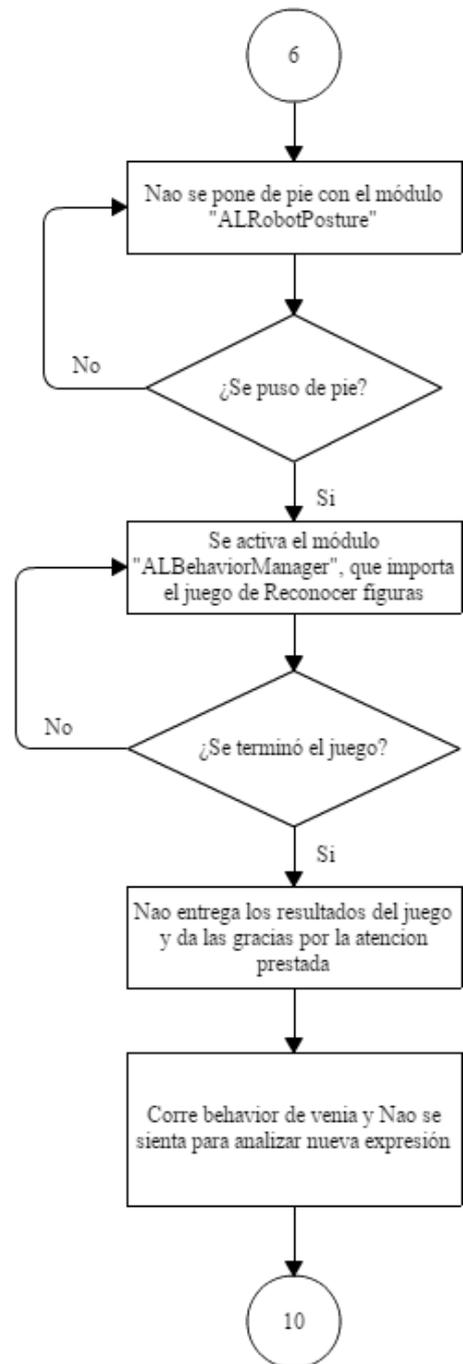
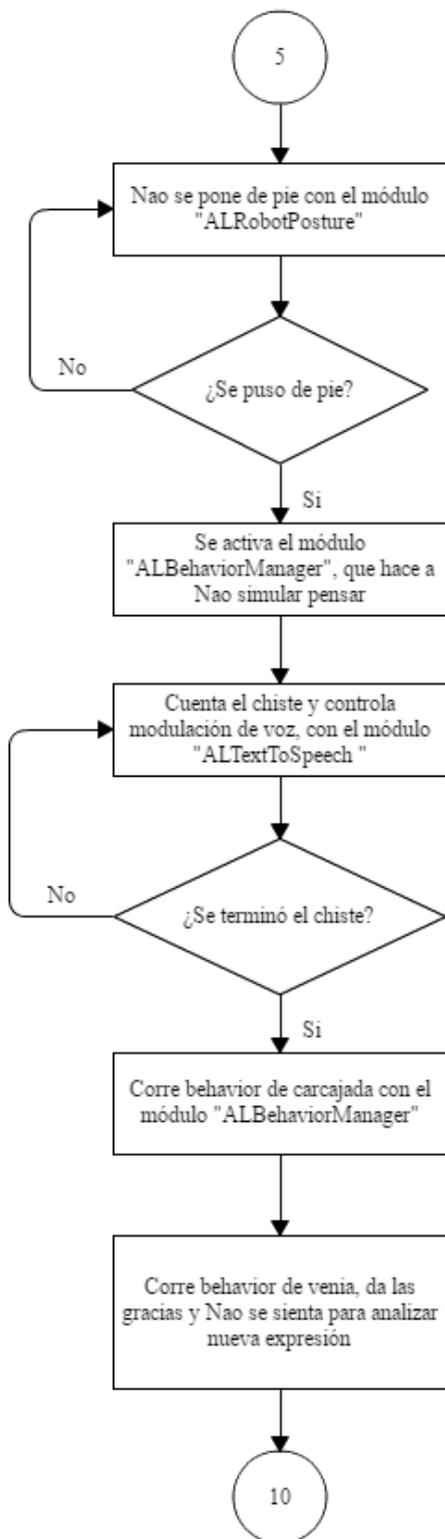
ANEXO 5: Clasificación de actividades tranquilo



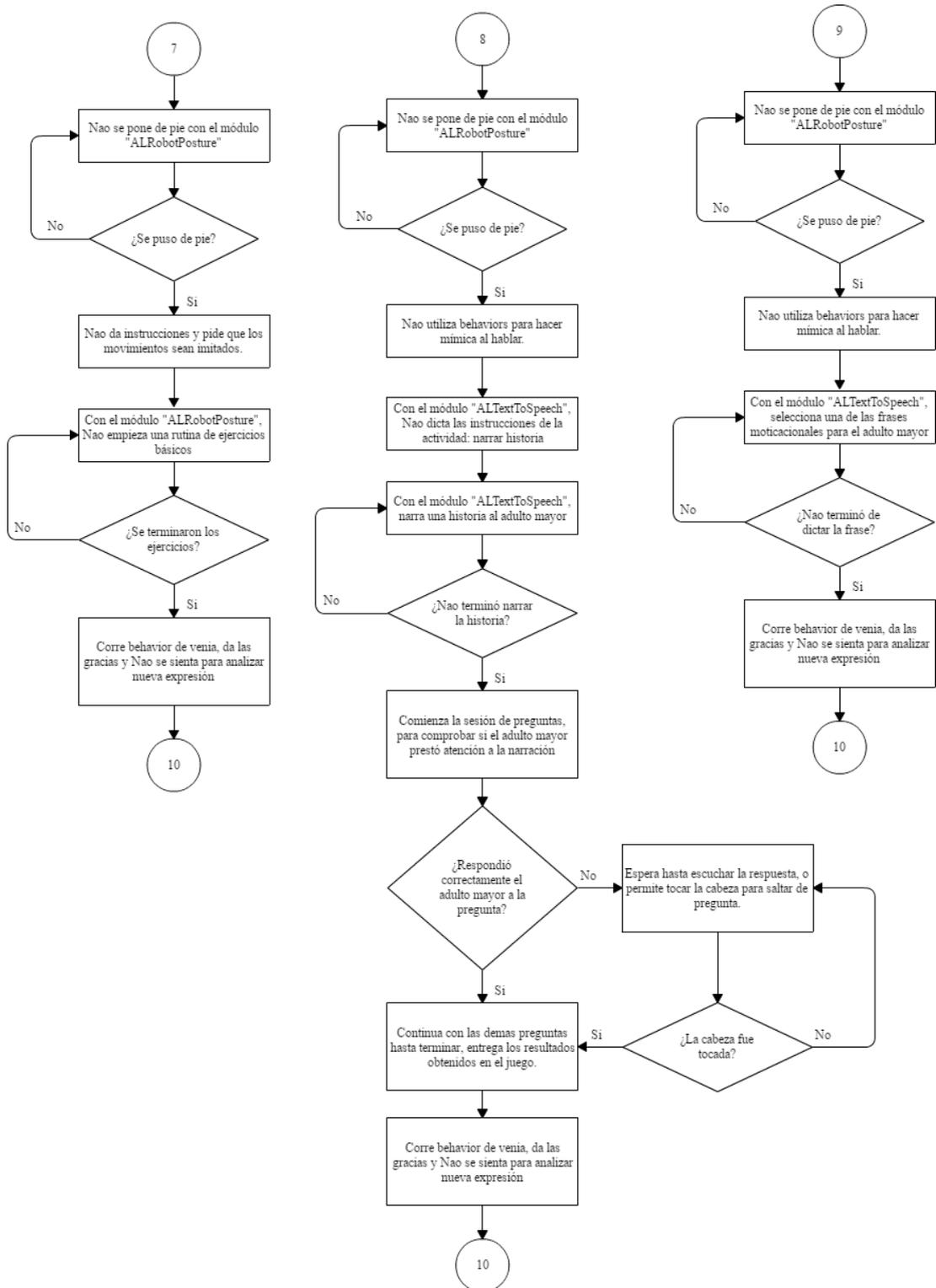
ANEXO 6: Actividad canciones y bailes



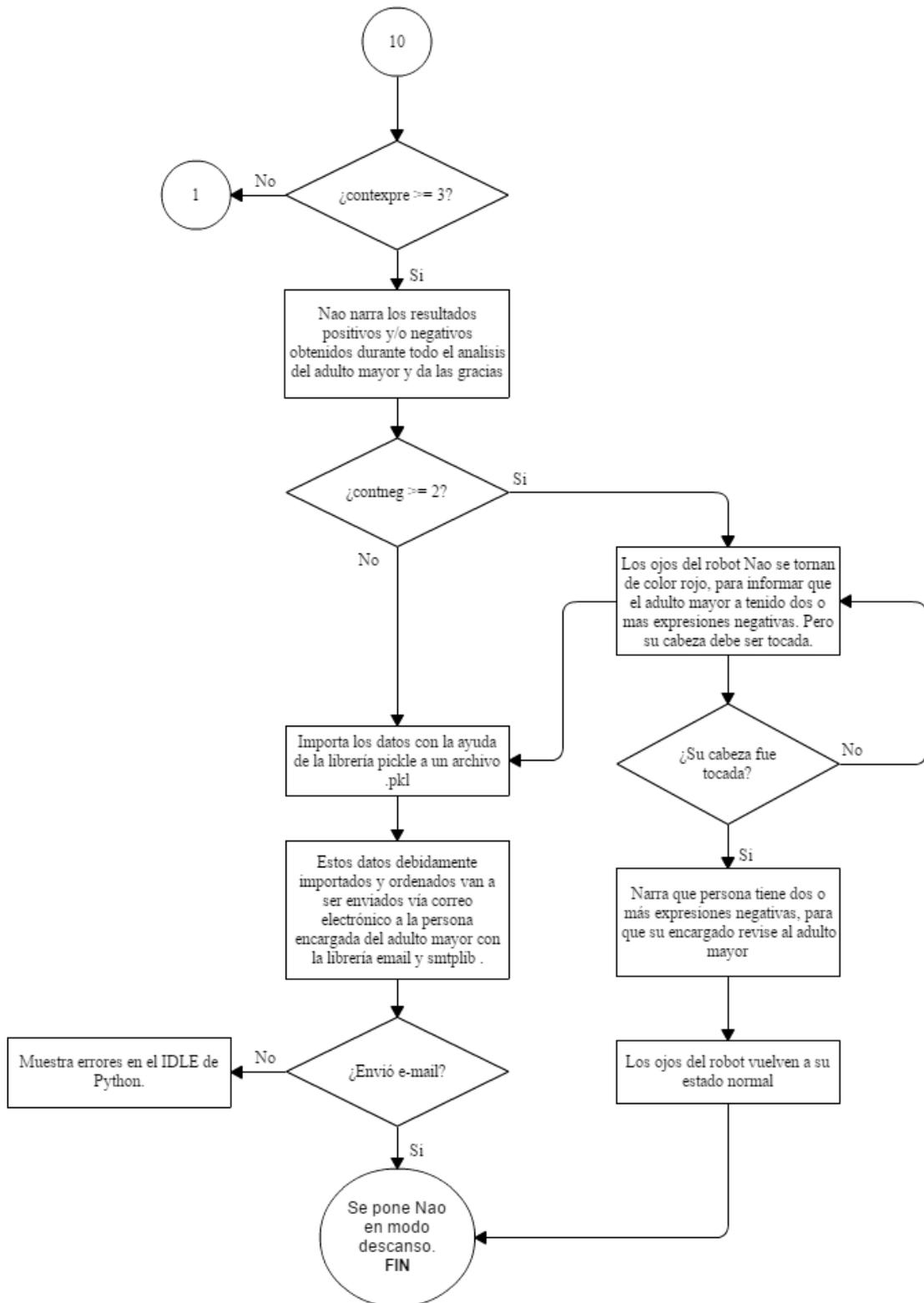
ANEXO 7: Actividad chistes y reconocer figuras



ANEXO 8: Actividades ejercicios, historias y frases



ANEXO 9: Resultados



ANEXO 10: Pruebas de reconocimiento de expresión enojado

#	P1	Neutral	Feliz	Asombrado	Enojado	Triste	Valor alto
1	PA1	0.11999999731779099	0.0	0.0	0.8199999928474426	0.05999999865889549	enojado
2	PA2	0.0	0.0	0.0	0.949999988079071	0.04999999701976776	enojado
3	PA3	0.1599999964237213	0.0	0.009999999776482582	0.8299999833106995	0.0	enojado
4	PA4	0.10999999940395355	0.0	0.0	0.6699999570846558	0.2199999988079071	enojado
5	PA5	0.029999999329447746	0.0	0.019999999552965164	0.8499999642372131	0.09999999403953552	enojado
6	PA6	0.07000000029802322	0.0	0.0	0.8299999833106995	0.09999999403953552	enojado
7	PA7	0.0	0.0	0.0	0.2199999988079071	0.7799999713897705	triste
8	PA8	0.3100000023841858	0.0	0.0	0.35999998450279236	0.32999998331069946	enojado
9	PA9	0.0	0.0	0.0	0.9899999499320984	0.009999999776482582	enojado
10	PA10	0.3999999761581421	0.0	0.019999999552965164	0.5799999833106995	0.0	enojado
11	PA1	0.0	0.0	0.0	0.9899999499320984	0.009999999776482582	enojado
12	PA2	0.0	0.0	0.0	1.0	0.0	enojado
13	PA3	0.17999999225139618	0.0	0.0	0.7899999618530273	0.029999999329447746	enojado
14	PA4	0.009999999776482582	0.0	0.0	0.9799999594688416	0.009999999776482582	enojado
15	PA5	0.17000000178813934	0.14999999105930328	0.0	0.6499999761581421	0.029999999329447746	enojado
16	PA6	0.25	0.0	0.0	0.6299999952316284	0.11999999731779099	enojado
17	PA7	0.0	0.0	0.0	0.8299999833106995	0.17000000178813934	enojado
18	PA8	0.23999999463558197	0.0	0.0	0.38999998569488525	0.3700000047683716	enojado
19	PA9	0.04999999701976776			0.9199999570846558	0.029999999329447746	enojado
20	PA10	0.28999999165534973	0.0	0.019999999552965164	0.6499999761581421	0.03999999910593033	enojado

ANEXO 11: Pruebas de reconocimiento de expresión feliz

#	P1	Neutral	Feliz	Asombrado	Enojado	Triste	Valor alto
1	PA1	0.0	1.0	0.0	0.0	0.0	feliz
2	PA2	0.0	1.0	0.0	0.0	0.0	feliz
3	PA3	0.019999999552965164	0.97999999594688416	0.0	0.0	0.0	feliz
4	PA4	0.029999999329447746	0.4699999988079071	0.019999999552965164	0.029999999329447746	0.449999998807907104	feliz
5	PA5	0.0	0.91999999570846558	0.0	0.0	0.07999999821186066	feliz
6	PA6	0.009999999776482582	0.9899999499320984	0.0	0.0	0.0	feliz
7	PA7	0.0	0.859999995470047	0.0	0.0	0.14000000059604645	feliz
8	PA8	0.0	1.0	0.0	0.0	0.0	feliz
9	PA9	0.05999999865889549	0.72999999594688416	0.0	0.04999999701976776	0.1599999964237213	feliz
10	PA10	0.0	0.89999999761581421	0.009999999776482582	0.0	0.08999999612569809	feliz
11	PA1	0.0	1.0	0.0	0.0	0.0	feliz
12	PA2	0.0	1.0	0.0	0.0	0.0	feliz
13	PA3	0.03999999910593033	0.95999999785423279	0.0	0.0	0.0	feliz
14	PA4	0.1599999964237213	0.449999998807907104	0.0	0.23999999463558197	0.14999999105930328	feliz
15	PA5	0.009999999776482582	0.95999999785423279	0.009999999776482582	0.0	0.019999999552965164	feliz
16	PA6	0.0	0.97999999594688416	0.0	0.0	0.019999999552965164	feliz
17	PA7	0.0	0.9899999499320984	0.0	0.0	0.009999999776482582	feliz
18	PA8	0.0	1.0	0.0	0.0	0.0	feliz
19	PA9	0.009999999776482582	0.8199999928474426	0.0	0.03999999910593033	0.12999999523162842	feliz
20	PA10	0.0	1.0	0.0	0.0	0.0	feliz

ANEXO 12:Pruebas de reconocimiento de expresión neutral

#	P1	Neutral	Feliz	Asombrado	Enojado	Triste	Valor alto
1	PA1	0.5399999618530273	0.0	0.0	0.029999999329447746	0.429999977350235	tranquilo
2	PA2	0.5600000023841858	0.0	0.22999998927116394	0.20999999344348907	0.0	tranquilo
3	PA3	0.6699999570846558	0.0	0.0	0.10999999940395355	0.2199999988079071	tranquilo
4	PA4	0.029999999329447746	0.0	0.17999999225139618	0.09999999403953552	0.6899999976158142	triste
5	PA5	0.9199999570846558	0.0	0.07000000029802322	0.0	0.009999999776482582	tranquilo
6	PA6	0.7400000095367432	0.0	0.0	0.25	0.009999999776482582	tranquilo
7	PA7	0.4699999988079071	0.0	0.0	0.3499999940395355	0.17999999225139618	tranquilo
8	PA8	0.6100000143051147	0.04999999701976776	0.019999999552965164	0.029999999329447746	0.28999999165534973	tranquilo
9	PA9	0.6399999856948853		0.009999999776482582	0.2199999988079071	0.12999999523162842	tranquilo
10	PA10	0.5899999737739563	0.1899999976158142	0.20999999344348907	0.0	0.009999999776482582	tranquilo
11	PA1	0.6699999570846558	0.0	0.0	0.14000000059604645	0.1899999976158142	tranquilo
12	PA2	0.4899999797344208	0.009999999776482582	0.17999999225139618	0.20999999344348907	0.10999999940395355	tranquilo
13	PA3	0.75	0.0	0.0	0.009999999776482582	0.23999999463558197	tranquilo
14	PA4	0.019999999552965164	0.0	0.11999999731779099	0.38999998569488525	0.4699999988079071	triste
15	PA5	0.7099999785423279	0.0	0.2800000011920929	0.0	0.009999999776482582	tranquilo
16	PA6	0.6599999666213989	0.0	0.23999999463558197	0.029999999329447746	0.07000000029802322	tranquilo
17	PA7	0.5600000023841858	0.03999999910593033	0.0	0.1599999964237213	0.23999999463558197	tranquilo
18	PA8	0.8100000023841858	0.0	0.0	0.0	0.1899999976158142	tranquilo
19	PA9	0.8499999642372131	0.0	0.0	0.009999999776482582	0.14000000059604645	tranquilo
20	PA10	0.6100000143051147	0.28999999165534973	0.08999999612569809	0.0	0.009999999776482582	tranquilo

ANEXO 13: Pruebas de reconocimiento de expresión asombrado

#	P1	Neutral	Feliz	Asombrado	Enojado	Triste	Valor alto
1	PA1	0.0	0.0	1.0	0.0	0.0	sorprendido
2	PA2	0.0	0.0	0.85999995470047	0.01999999552965164	0.1199999731779099	sorprendido
3	PA3	0.5399999618530273	0.0	0.0499999701976776	0.319999928474426	0.0899999612569809	tranquilo
4	PA4	0.0	0.0	0.9799999594688416	0.01999999552965164	0.0	sorprendido
5	PA5	0.0	0.0	0.5	0.2800000011920929	0.219999988079071	sorprendido
6	PA6	0.409999964237213	0.0	0.41999998688697815	0.0	0.17000000178813934	sorprendido
7	PA7	0.0	0.26999998092651367	0.7299999594688416	0.0	0.0	sorprendido
8	PA8	0.00999999776482582	0.00999999776482582	0.6499999761581421	0.0	0.32999998331069946	sorprendido
9	PA9	0	0	0.9799999594688416	0.0	0.01999999552965164	sorprendido
10	PA10	0.00999999776482582	0	0.7599999904632568	0	0.22999998927116394	sorprendido
11	PA1	0.0	0.0	1.0	0.0	0.0	sorprendido
12	PA2	0.05999999865889549	0.0	0.8999999761581421	0.0	0.03999999910593033	sorprendido
13	PA3	0.5099999904632568	0.0	0.02999999329447746	0.429999977350235	0.02999999329447746	tranquilo
14	PA4	0.0	0.0	0.5699999928474426	0.409999964237213	0.01999999552965164	sorprendido
15	PA5	0.0	0.0	0.5600000023841858	0.29999998211860657	0.14000000059604645	sorprendido
16	PA6	0.3700000047683716	0.0	0.5299999713897705	0.0	0.09999999403953552	sorprendido
17	PA7	0.08999999612569809	0.0	0.8999999761581421	0.00999999776482582	0.0	sorprendido
18	PA8	0.02999999329447746	0.0	0.9599999785423279	0.0	0.00999999776482582	sorprendido
19	PA9	0	0.01999999552965164	0.9599999785423279	0	0.01999999552965164	sorprendido
20	PA10	0.14999999105930328	0	0.7199999690055847	0.12999999523162842	0	sorprendido

ANEXO 14: Pruebas de reconocimiento de expresión triste

#	P1	Neutral	Feliz	Asombrado	Enojado	Triste	Valor alto
1	PA9	0.0	0.25999999046325684	0.0	0.14000000059604645	0.5999999642372131	triste
2	PA10	0.10999999940395355	0.0	0.04999999701976776	0.47999998927116394	0.35999998450279236	enojado
3	PA1	0.09999999403953552	0.0	0.0	0.85999995470047	0.03999999910593033	enojado
4	PA2	0.029999999329447746	0.0	0.03999999910593033	0.1899999976158142	0.7400000095367432	triste
5	PA3	0.38999998569488525	0.0	0.0	0.029999999329447746	0.5799999833106995	triste
6	PA4	0.07000000029802322	0.03999999910593033	0.0	0.35999998450279236	0.5299999713897705	triste
7	PA5	0.08999999612569809	0.0	0.0	0.2800000011920929	0.6299999952316284	triste
8	PA6	0.05999999865889549	0.0	0.0	0.9300000071525574	0.009999999776482582	enojado
9	PA7	0.05999999865889549	0.0	0.0	0.019999999552965164	0.9199999570846558	triste
10	PA8	0.26999998092651367	0.0	0.0	0.0	0.7299999594688416	triste
11	PA9	0.10999999940395355	0	0.32999998331069946	0	0.5600000023841858	triste
12	PA10	0.14000000059604645	0.0	0.019999999552965164	0.7899999618530273	0.04999999701976776	enojado
13	PA1	0.0	0.0	0.0	0.8899999856948853	0.10999999940395355	enojado
14	PA2	0.009999999776482582	0.12999999523162842	0.019999999552965164	0.20999999344348907	0.6299999952316284	triste
15	PA3	0.05999999865889549	0.0	0.0	0.03999999910593033	0.8999999761581421	triste
16	PA4	0.0	0.0	0.009999999776482582	0.41999998688697815	0.5699999928474426	triste
17	PA5	0.0	0.0	0.029999999329447746	0.0	0.9699999690055847	triste
18	PA6	0.029999999329447746	0.0	0.0	0.8499999642372131	0.11999999731779099	enojado
19	PA7	0.23999999463558197	0.0	0.009999999776482582	0.29999998211860657	0.44999998807907104	triste
20	PA8	0.17999999225139618	0.0	0.009999999776482582	0.009999999776482582	0.7999999523162842	triste

ANEXO 15: Pruebas de reconocimiento de comandos de voz con del participante 1 OS (Oswaldo Romero).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
1 OS	Canción	3	4	50	160-170	Sentado	100	Canción	10	14	71,4
	Chiste	0	1					Chiste	0	1	0,0
	Frase	6	7	Frase		6	7	85,7			
	Historia	0	1	Historia		0	1	0,0			
	Quito	0	1	Quito		0	1	0,0			
	Emancipadoras	1	2	Emancipadoras		1	2	50,0			
	Canción	5	6	Baile		0	1	0,0			
	Canción	1	2	Canción		1	2	50,0			
	Baile	0	1	Reconocer Figuras		3	4	75,0			
	Canción	1	2	100		Sentado	50				
Reconocer Figuras	3	4									

ANEXO 16: Pruebas de reconocimiento de comandos de voz con del participante 2 ET (Erlinda Troya).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
2 ET	Canción	3	4	50 - 75	140 - 150	Sentado	50	Canción	8	11	72,7
	Baile	0	1					Chiste	1	3	33,3
	Chiste	1	2					Baile	2	5	40,0
	Canción	3	4	50 - 75		Sentado	100	Reconocer Figuras	0	1	0,0
	Chiste	0	1					Canción	2	3	50 - 75
	Baile	0	1					Baile	2	3	
	Reconocer Figuras	0	1	50 - 75		Sentado	0				
	Canción	2	3								
	Baile	2	3								

ANEXO 17: Pruebas de reconocimiento de comandos de voz con del participante 3 BM (Beatriz Morales).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
3 BM	Canción	8	9	75	140 - 150	Sentado	0	Canción	8	9	88,9
	Historia	2	3					Chiste	16	18	88,9
	Quito	4	5					Frase	17	20	85,0
	Frase	3	4					Historia	2	3	66,7
	Reconocer figuras	2	3	75		Sentado	0	Quito	4	5	80,0
	Chiste	11	12					Baile	0	2	0,0
	Frase	7	8					Baile	0	2	0,0
	Baile	0	1	<50		De pie	100	Reconocer Figuras	2	3	66,7
	Frase	4	5								
	Chiste	5	6								

ANEXO 18: Pruebas de reconocimiento de comandos de voz con del participante 4 MC (Mariana Calvopiña).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
4 MC	Reconocer figuras	0	1	35	165 - 175	De pie	100	Canción	0	2	0,0
	Frase	0	1					Frase	0	4	0,0
	Canción	0	1	Reconocer Figuras		0	3	0,0			
	Frase	0	1								
	Reconocer figuras	0	1	35		De pie	100				
	Frase	0	1								
	Canción	0	1								
	Reconocer figuras	0	1	35		De pie	100				
Frase	0	1									

ANEXO 19: Pruebas de reconocimiento de comandos de voz con del participante 5 JV (Julio Valladares).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
5 JV	Reconocer figuras	0	1	50	160 - 170	De pie	100	Canción	0	2	0,0
	Canción	0	1					Chiste	0	1	0,0
	Chiste	0	1	75		De pie	100	San Francisco	0	2	0,0
	Canción	0	1					Historia	0	2	0,0
	Reconocer figuras	0	1	75		De pie	100	Piedra	0	2	0,0
	Frase	0	1					Baile	0	1	0,0
	Historia	0	1		Reconocer Figuras			0	2	0,0	
	San Francisco	0	1		Luzbel			0	1	0,0	
	Piedra	0	1		Frase			0	1	0,0	
	Baile	0	1								
	Historia	0	1								
	San Francisco	0	1								
	Luzbel	0	1								
	Piedra	0	1								

ANEXO 20: Pruebas de reconocimiento de comandos de voz con del participante 6 AH (Adela Herrera).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)
6 AH	Chiste	1	2	75	150 - 160	De pie	100
	Canción	1	2				
	Baile	0	1				
	Chiste	0	1	75		De pie	100
	Baile	1	2				
	Canción	0	1				
	Canción	0	1	75		De pie	100
	Chiste	0	1				
	Baile	0	1				

Frase	Número de Errores	Número de pruebas	% Error
Canción	1	4	25,0
Chiste	1	4	25,0
Baile	1	4	25,0

ANEXO 21: Pruebas de reconocimiento de comandos de voz con del participante 7 LR (Leonor Rosero).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
7 LR	Canción	0	1	60	150 - 160	De pie	100	Canción	0	3	0,0
	Baile	1	2					Chiste	1	2	50,0
	Canción	0	1					San Francisco	0	1	0,0
	Canción	0	1	60		De pie	100	Historia	0	1	0,0
	Reconocer figuras	0	1					Piedra	0	1	0,0
	Historia	0	1					Baile	1	3	33,3
	San Francisco	0	1					Reconocer Figuras	0	1	0,0
	Luzbel	0	1	70		De pie	100	Luzbel	0	1	0,0
	Piedra	0	1					Ejercicio	1	2	50,0
	Chiste	1	2					Baile	0	1	

ANEXO 22: Pruebas de reconocimiento de comandos de voz con del participante 8 GA (Gloria Arequipa).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error	
8 GA	Historia	0	1	80	150 - 160	Sentado	0	Canción	1	4	25,0	
	Quito	0	1					Chiste	0	2	0,0	
	Emancipadoras	1	2					Historia	0	1	0,0	
	Baile	0	1					Quito	0	1	0,0	
	Chiste	0	1					Emancipadoras	1	2	50,0	
	Canción	1	2	70		Sentado	0	Baile	0	3	0,0	
	Baile	0	1					Canción	0	1		
	Chiste	0	1					Canción	0	1		
		Canción	0	1		70	Sentado	0	Baile	0	1	
		Canción	0	1								
	Baile	0	1									

ANEXO 23: Pruebas de reconocimiento de comandos de voz con del participante 9 GJ (Gloria Jurado).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
9 GJ	Chiste	0	1	100	140 - 150	Sentado	0	Canción	1	3	33,3
	Historia	0	1					Chiste	0	2	0,0
	San Francisco	0	1					San Francisco	0	1	0,0
	Piedra	0	1					Historia	0	1	0,0
	Canción	0	1					Piedra	0	1	0,0
	Canción	1	2	100		Sentado	0	Reconocer Figuras	1	2	50,0
	Reconocer figuras	1	2								
Chiste	0	1									

ANEXO 24: Pruebas de reconocimiento de comandos de voz con del participante 10 GT (Gonzalo Tafur).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)
10 GT	Frase	0	1	80	150 - 160	Sentado	0
	Baile	0	1				
	Chiste	0	1				
	Canción	1	2	80		Sentado	0
	Chiste	0	1				
	Baile	0	1				
	Chiste	0	1	80		Sentado	0
	Canción	0	1				
	Baile	0	1				

Frase	Número de Errores	Número de pruebas	% Error
Canción	1	3	33,3
Chiste	0	3	0,0
Baile	0	3	0,0
Frase	0	1	0,0

ANEXO 25: Pruebas de reconocimiento de comandos de voz con del participante 11 SC (Salomón Calvopiña).

ID	Frase	Errores	Número de veces que se dictó el comando	Distancia entre persona y robot (cm)	Tamaño de la persona (cm)	Estado de la persona	Estado del robot sobre el piso (cm)	Frase	Número de Errores	Número de pruebas	% Error
11 SC	Historia	0	1	100	170 - 175	Sentado	0	Chiste	0	1	0,0
	San Francisco	1	2					San Francisco	1	2	50,0
	Luzbel	1	2					Historia	0	1	0,0
	Piedra	0	1					Piedra	0	1	0,0
	Frase	0	1					Frase	0	1	0,0
	Reconocer figuras	1	2	Reconocer Figuras		1	2	50,0			
	Ejercicios	0	1	100		Sentado	0	Luzbel	1	2	50,0
	Chiste	0	1					Ejercicios	0	1	0,0
	Historia	0	1					Quito	0	1	0,0
	Quito	0	1					Emancipadoras	0	1	0,0
Emancipadoras	0	1									