



**ESPE**

**UNIVERSIDAD DE LAS FUERZAS ARMADAS**  
**INNOVACIÓN PARA LA EXCELENCIA**

**DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA**

**CARRERA DE INGENIERÍA ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL**

**TEMA: DESARROLLO DE UN SISTEMA BÁSICO DE ROBOTICA DE ENTRETENIMIENTO PERSUASIVO BASADO EN EL SISTEMA HUMANOIDE NAO**

**AUTOR: VILATUÑA SALGUERO, FABRICIO SEBASTIÁN**

**DIRECTOR: ING. IBARRA JÁCOME, OSWALDO ALEXANDER**

**SANGOLQUÍ**

**2018**



**ESPE**  
UNIVERSIDAD DE LAS FUERZAS ARMADAS  
INNOVACIÓN PARA LA EXCELENCIA

i

## DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

### CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

#### CERTIFICACIÓN

Certifico que el trabajo de titulación “**DESARROLLO DE UN SISTEMA BÁSICO DE ROBOTICA DE ENTRETENIMIENTO PERSUASIVO BASADO EN EL SISTEMA HUMANOIDE NAO**” fue realizado por el señor **Vilatuña Salguero, Fabricio Sebastián** el mismo que ha sido revisado en su totalidad, analizado por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolqui, 08 de Agosto del 2018

Ing. Alexander Ibarra J.  
CI: 1719535427



## DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

### CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

#### DECLARACIÓN DE RESPONSABILIDAD

Yo, **Vilatuña Salguero, Fabricio Sebastián**, declaro que el contenido, ideas y criterios del trabajo de titulación: **“DESARROLLO DE UN SISTEMA BÁSICO DE ROBOTICA DE ENTRETENIMIENTO PERSUASIVO BASADO EN EL SISTEMA HUMANOIDE NAO”** es de mi autoría y responsabilidad, cumpliendo con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas utilizadas.

Consecuentemente el contenido de la investigación mencionada es veraz.

Sangolqui, 08 de Agosto del 2018



---

FABRICIO SEBASTIÁN VILATUÑA SALGUERO

CI: 1722581178



## DEPARTAMENTO DE ELÉCTRICA Y ELECTRÓNICA

### CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y CONTROL

#### AUTORIZACIÓN

Yo, **Vilatuña Salguero Fabricio Sebastián**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“DESARROLLO DE UN SISTEMA BÁSICO DE ROBOTICA DE ENTRETENIMIENTO PERSUASIVO BASADO EN EL SISTEMA HUMANOIDE NAO”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolqui, 08 de Agosto del 2018

---

FABRICIO SEBASTIÁN VILATUÑA SALGUERO

CI: 1722581178

## DEDICATORIA

Este proyecto se lo dedico a las personas más importantes en mi vida, que me han sabido guiar por el mejor camino y junto a su apoyo y cariño, culminar mi formación profesional, a ellos, mis padres.

A mi madre Nancy, por ser una mujer de inspiración, gracias a su apoyo constante, amor y dedicación hacia sus hijos nos ha sabido llevar de la mano durante toda nuestras vidas, buscando siempre lo mejor para nosotros, apoyándonos y aconsejándonos en todas nuestras decisiones y brindándonos todo para salir adelante.

A mi padre Bolívar, por ser mi más grande ejemplo e inspiración, gracias a su lucha incansable, trabajo y dedicación me brindaste todo lo que estuvo a tu alcance, inculcándome valores éticos y guiándome siempre por el camino del bien, brindándome siempre su apoyo y confianza.

A mis hermanos Geovanna y Lenin, por acompañarme, educarme, guiarme y cuidarme siempre desde pequeño, por ser los mejores hermanos del mundo, a los que admiro por sus logros y metas alcanzadas, por ser excelentes profesionales y seres humanos.

Este trabajo y toda mi carrera no habrían sido posibles sin el apoyo y entrega incondicional de cada uno de ustedes, siendo mi principal fuente de inspiración y mi más grande ejemplo de sabiduría y fortaleza.

## **AGRADECIMIENTOS**

Agradezco en primer lugar a mis padres por ayudarme cada día de mi vida a crecer como persona, por sentir felicidad de las metas y sueños cumplidos o hacerlos posibles. Gracias por brindarme su afecto incondicional, por haberme dado siempre una educación de calidad e inculcarme los mejores valores.

Agradezco a mis hermanos por sus consejos y sus palabras de aliento que me ayudaron a llegar a cumplir esta meta, por ser mi ejemplo y mis cómplices, Gracias Geovy, mi hermanita querida, por brindarme tantas experiencias y permitirme conocer más allá de las fronteras de este país, por ser mi compañera de viajes y apoyarme siempre en todo lo que he necesitado.

Gracias a todos mis amigos con los que compartí esta experiencia universitaria, porque con cada uno vivimos momentos de alegría, tristeza, triunfos y fracasos, pero siempre he contado con su apoyo, gracias por tantos consejos y gratas experiencias, es un honor formar parte de su vida y es un placer que cada uno de ustedes formen parte de la mía, gracias Javi, Mari, Didi, a mis Geógrafas, las más hermosas del mundo y a la familia Pililín.

Un agradecimiento especial al Ingeniero Alexander Ibarra, por el apoyo que me brindo a lo largo del desarrollo de esta tesis, gracias por las enseñanzas, consejos y por brindarme su amistad.

## ÍNDICE DE CONTENIDOS

<b>CERTIFICADO DEL DIRECTOR</b> .....	<b>i</b>
<b>AUTORÍA DE RESPONSABILIDAD</b> .....	<b>ii</b>
<b>AUTORIZACIÓN</b> .....	<b>iii</b>
<b>DEDICATORIA</b> .....	<b>iv</b>
<b>AGRADECIMIENTOS</b> .....	<b>v</b>
<b>ÍNDICE DE CONTENIDOS</b> .....	<b>vi</b>
<b>ÍNDICE DE TABLAS</b> .....	<b>x</b>
<b>ÍNDICE DE FIGURAS</b> .....	<b>xi</b>
<b>RESUMEN</b> .....	<b>xiv</b>
<b>ABSTRACT</b> .....	<b>xv</b>
<b>CAPÍTULO I</b> .....	<b>1</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
1.1 Antecedentes .....	1
1.2 Justificación e Importancia .....	4
1.3 Alcance del Proyecto .....	6

1.5 Objetivos.....	7
1.5.1 Objetivo General.....	7
1.5.2 Objetivos Específicos .....	7
<b>CAPÍTULO II.....</b>	<b>8</b>
<b>ESTADO DEL ARTE .....</b>	<b>8</b>
2.1 Introducción.....	8
2.2 Interacción Humano-Robot (HRI).....	11
2.3 Robótica social .....	14
2.4 Robótica persuasiva .....	16
2.4.1 Estrategias de robótica persuasiva.....	18
<b>CAPÍTULO III .....</b>	<b>20</b>
<b>SISTEMA DE DESARROLLO DE APLICACIONES PARA EL ROBOT NAO.....</b>	<b>20</b>
3.1 Introducción.....	20
3.2 Estructura de NAOqi .....	25
3.3 Herramientas de ROS Y QIBUILD.....	26
3.4 Instalación de Driver NAO para los sistemas ROS y QIBUILD. ....	28
3.4.1 Ejecución de ROS y comandos utilizados.....	28
3.4.2 Ejecución de QIBUILD y comandos utilizados. ....	30
3.5 Comunicación con el robot humanoide NAO. ....	31
3.5.1 Comunicación entre NAO y ROS. ....	33

3.5.2 Comunicación entre NAO y QIBUILD.....	35
3.6 Selección del sistema.....	38
<b>CAPÍTULO IV.....</b>	<b>41</b>
<b>ALGORITMOS DE INTERACCIÓN HUMANO-ROBOT .....</b>	<b>41</b>
4.1 Expresión corporal y movimientos planificados del Robot .....	41
4.1.1 Módulo ALMotion .....	42
4.1.2 Descripción de las Articulaciones. ....	45
4.1.2 Control de las Articulaciones en C++ .....	51
4.2 Reconocimiento facial .....	54
4.2.1 Módulo ALFaceDetection. ....	56
4.2.2 Detección y reconocimiento facial en C++. ....	60
4.3 Gestor de Diálogo.....	62
4.3.1 Módulo ALDialog. ....	63
4.4 Interacción con el robot NAO .....	67
<b>CAPÍTULO V.....</b>	<b>74</b>
<b>RESULTADOS.....</b>	<b>74</b>
5.1 Evaluación .....	74
5.2 Análisis de resultados .....	78
5.2.1 Resultados de desempeño para cada rango de edades. ....	81
5.2.2 Resultados sobre la experiencia de interacción con un robot. ....	86

5.3 Trabajos futuros.....	87
<b>CAPÍTULO VI.....</b>	<b>89</b>
<b>CONCLUSIONES Y RECOMENDACIONES .....</b>	<b>89</b>
6.1 Conclusiones.....	89
6.2 Recomendaciones .....	91
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>92</b>

## ÍNDICE DE TABLAS

Tabla 1 Cantidad y tipos de sensores del robot humanoide NAO .....	22
Tabla 2. Características de los sistemas QIBuild y ROS .....	39
Tabla 3. Articulaciones disponibles de acuerdo a la extremidad del Robot NAO.....	43
Tabla 4. Ángulos límites para el movimiento de la cabeza.....	46
Tabla 5. Ángulos límites para las articulaciones del brazo derecho. ....	48
Tabla 6. Ángulos límites para las articulaciones del brazo izquierdo.....	48
Tabla 7. Ángulos límites del torso. ....	49
Tabla 8. Ángulos límites para las articulaciones de la pierna derecha.....	51
Tabla 9. Ángulos límites para las articulaciones de la pierna izquierda. ....	51
Tabla 10. Características de configuración para las cámaras del robot NAO.....	55
Tabla 11. Datos obtenidos de los usuarios de 7 a 12 años. ....	81
Tabla 12. Datos obtenidos de los usuarios de 13 a 19 años. ....	81
Tabla 13. Datos obtenidos de los usuarios de 20 a 29 años de edad.....	82
Tabla 14. Datos obtenidos de los usuarios de 30 a 59 años de edad.....	82
Tabla 15. Datos obtenidos de los usuarios de 60 a 80 años de edad.....	82
Tabla 16. Resultados mostrados por cada grupo de edad. ....	83

## ÍNDICE DE FIGURAS

Figura 1. Robot NAO .....	3
Figura 2. Interacción social con el robot ASIMO .....	9
Figura 3. Experimento de interacción entre un robot y Niños con autismo.....	11
Figura 4. Experimento de HRI con un sujeto de prueba. ....	13
Figura 5. Robot AIBO de Sony .....	15
Figura 6. Factores de persuasión para HRI .....	18
Figura 7. Estrategias de persuasión. ....	19
Figura 8. Elementos de la plataforma robótica NAO.....	20
Figura 9. Torneo Robocup realizado con robots NAO .....	21
Figura 10. Logo del programa Choregraphe. ....	23
Figura 11. Ejemplo de programación y entorno de Choregraphe .....	24
Figura 12. Estructura de NAOqi: Broker, librerías y módulos. ....	25
Figura 13. Estructura de NAOqi: Broker, Modulos y Metodos.....	26
Figura 14. Ícono del sistema operativo de Robots (ROS).....	27
Figura 15. Iconos correspondientes de Ubuntu y ROS Índigo.....	28
Figura 16. Información sobre qitoolchain. ....	31
Figura 17. Forma de conexión ordenador-robot.....	32
Figura 18. Conexión real entre el ordenador y el robot NAO.....	32
Figura 19. Conexión exitosa entre ROS y el robot NAO.....	34
Figura 20. Lista de tópicos disponibles para el robot NAO.....	35
Figura 21. Archivos creados para un proyecto de QIBUILD. ....	36

Figura 22. Configuración y construcción básica de un proyecto con mytoolchain .....	37
Figura 23. Ejecución de un programa usando QIBuild. ....	38
Figura 24. Articulaciones del robot NAO. ....	42
Figura 25. Ángulos límites para el movimiento de la cabeza. ....	46
Figura 26. Ángulos límites de las articulaciones para los brazos del robot .....	47
Figura 27. Ángulos límites del torso. ....	49
Figura 28. Ángulos límites de las articulaciones de las piernas. ....	50
Figura 29. Secuencia de movimiento realizado con el robot NAO. ....	54
Figura 30. Posicionamiento de las cámaras en el robot NAO. ....	55
Figura 31. Datos de un rostro, obtenidos con el módulo de reconocimiento facial. ....	60
Figura 32. Posicionamiento de los parlantes en el robot NAO .....	62
Figura 33. Posicionamiento de los micrófonos en el robot NAO .....	63
Figura 34. Diagrama de flujo de la función de reconocimiento facial. ....	68
Figura 35. Diagrama de flujo de la función de Diálogo 1. ....	70
Figura 36. Diagrama de flujo de la función de Diálogo 2. ....	71
Figura 37. Primera demostración grupal del funcionamiento del robot. ....	74
Figura 38. Segunda demostración grupal del funcionamiento del robot NAO .....	75
Figura 39. Interacción con usuarios entre 7 a 12 años. ....	79
Figura 40. Interacción con usuarios entre 13 a 19 años. ....	79
Figura 41. Interacción con usuarios entre 20 a 29 años .....	79
Figura 42. Interacción con usuarios entre 30 a 59 años. ....	80
Figura 43. Interacción con usuarios entre 60 a 80 años. ....	80

Figura 44. Gráfica de aceptación del parámetro de entretenimiento.....	84
Figura 45. Gráfica de aceptación del parámetro Actividades .....	84
Figura 46. Gráfica del porcentaje de aceptación las actividades propuestas por el robot.....	85
Figura 47. Gráfica de aceptación del parámetro Persuasión .....	86

## RESUMEN

El presente proyecto “Desarrollo de un sistema básico de robótica de entretenimiento persuasivo basado en el sistema humanoide NAO”, se fundamentó en la integración de algoritmos que hacen posible la interacción humano-robot (HRI), con un enfoque hacia las técnicas de persuasión con el fin de influenciar sobre una persona, para ello se plantea el uso de algoritmos de reconocimiento facial, sistemas de diálogo y expresiones corporales que se sustenta en movimientos y gestos que usará el robot para expresar de alguna manera emociones o llevar a cabo actividades programadas que estén acorde a la temática de entretenimiento, en la cual se basa este proyecto. Estos algoritmos le permiten al robot tener la capacidad de reconocer al usuario que interactúa con él, además de mantener un diálogo que vaya en relación al tema planteado, influenciando sobre el usuario con el fin de aceptar que se realicen las actividades propuestas por el robot. Inicialmente se analiza la plataforma de desarrollo de programación con la que se va a trabajar, realizando una comparación entre el sistema ROS y el sistema QIBUILD, las cuales permiten el desarrollo e implementación de algoritmos basados en un lenguaje de programación en C++, y a continuación se especifica cómo se implementa y se usan los algoritmos propuestos y al finalizar se realizan pruebas de funcionamiento del sistema que se llevan a cabo con una muestra de veinte usuarios agrupados por rangos de edades y se analiza cómo se llevó a cabo la interacción con el robot, el nivel de persuasión y entretenimiento.

### **PALABRAS CLAVE:**

- **INTERACCIÓN HUMANO-ROBOT**
- **ROBÓTICA DE ENTRETENIMIENTO**
- **SISTEMAS DE DIÁLOGO**

## **ABSTRACT**

The current project titled "Development of a basic system of persuasive entertainment robotics based on the humanoid system NAO", was based on the integration of algorithms that make possible the human-robot interaction (HRI), with a focus towards persuasion techniques in order to influence a person, for this purpose, the use of facial recognition algorithms, dialogue systems and corporal expressions are used, which is based on movements and gestures that the robot will use to express some emotions or perform programmed activities that are related to the theme of entertainment, on which this project is based. These algorithms allow the robot to have the ability to recognize the user that interacts with it, in addition to maintaining a dialogue that goes in relation to the proposed topic, influencing the user in order to accept the activities proposed by the robot. Initially the programming development platform is analyzed with which it will work, making a comparison between the ROS system and the QIBUILD system, which allow the development and implementation of algorithms based on a C ++ programming language, and then it specifies how the proposed algorithms are implemented and used, Lasty, system performance tests are carried out with a sample of twenty users grouped by age ranges and the interaction with the robot, the level of persuasion and entertainment is analyzed.

### **KEYWORDS:**

- **HUMAN-ROBOT INTERACTION**
- **ENTERTAINMENT ROBOTICS**
- **DIALOGUE SYSTEMS**

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 Antecedentes

La robótica, por el componente tecnológico y vanguardista que la define, siempre se ha considerado como un foco de atracción llamativo para el público en general. En este sentido, especialmente los robots humanoides, han encontrado un área de visibilidad en entornos en los que se busca atraer la atención del público, tales como ferias, eventos o congresos. Al margen de ello, y dentro de las amplias posibilidades que ofrece la labor investigadora, los humanoides se han instituido como unas eficaces herramientas en diferentes campos. (Robotnik, 2017).

En la actualidad robótica abarca una variedad de sub áreas de investigación, una de ellas es interacción humano-robot (HRI) es un campo multidisciplinario con contribuciones de la interacción humano-computadora, inteligencia artificial, robótica, comprensión del lenguaje natural, diseño y ciencias sociales. (Anderson & Owan, 2003)

Un requisito para HRI natural es dotar al robot con la capacidad de capturar, procesar y comprender de manera precisa y robusta las solicitudes humanas. Por lo tanto, es importante analizar las formas naturales mediante las cuales un ser humano puede interactuar y comunicarse con un robot. Un número considerable de sistemas robóticos se han desarrollado en la última década mostrando las capacidades de HRI. (Lucas, et al., 2008) (Alonso, 2014)

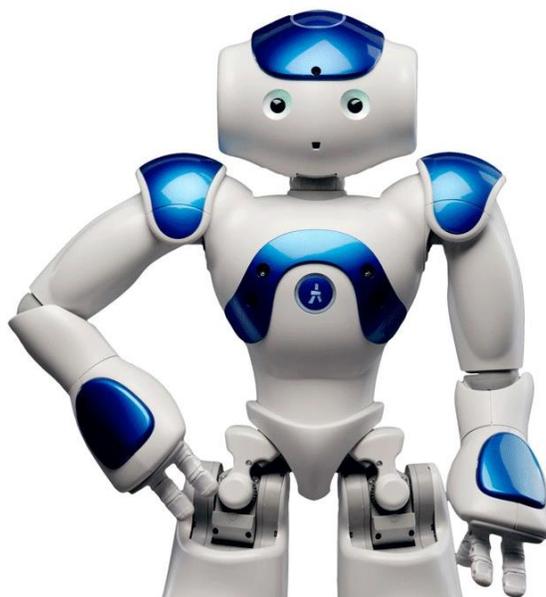
La interacción entre seres humanos y máquinas ha pasado de ser un paradigma de investigación a convertirse en la actualidad en una realidad que se da en diferentes niveles. El nivel de interacción más básico, más próximo a la máquina que al hombre, lleva décadas siendo usado (a través de dispositivos como teclados, generando comandos que la máquina debe interpretar). Sin embargo,

el campo más interesante es el desarrollo de plataformas que permitan una interacción a niveles más próximos a los que el ser humano emplea de manera intuitiva, tales como el uso de la voz o la expresión corporal. (Ismail, Shamsuddi, Yussof, & Jaafar, 2011)

Las denominadas tecnologías del habla tienen como finalidad facilitar el uso de los ordenadores, en nuestro caso de los robots, introduciendo y recibiendo información de modo oral. Para alcanzar este objetivo es necesario contar, al menos, con tres tecnologías básicas: las que posibilitan que la información escrita se convierta en voz (síntesis del habla), las que permiten que un sistema informático realice las tareas que se le solicitan verbalmente (el reconocimiento del habla) y las que facilitan la interacción oral entre una persona y un servicio (los sistemas de diálogo). (Robotronica, 2018)

Cuando se habla de interacción-humano robot, se asume que el robot tiene la capacidad de reconocer las palabras que el humano transmite y responder acorde a la temática tratada, pero para una interacción natural no es suficiente esas características, por lo que se debe abordar el tema de la “Robótica persuasiva”, la cual se aplica como un intento de cambiar las creencias o el comportamiento de otra persona, influenciar a otros permite tener una interacción social natural. Cualquier agente que desee operar sin problemas de una manera social necesitará incorporar este tipo de comportamiento humano básico. (Siegel, Breazeal, & Norton, 2009)

Entre los robots humanoides preferidos por los investigadores se encuentra el robot NAO, el cual mide aproximadamente 58 cm de alto y pesa alrededor de 4.8 Kg. Construido con policarbonato y tiene una batería de litio; Además está compuesto de un procesador Intel Atom de 1.6 GHz con Linux. 25 servos permiten controlar los 25 grados de libertad del robot. (Millán Romera, 2015)



**Figura 1.** Robot NAO  
Fuente: (Robotronica, 2018)

El robot NAO es capaz a través del desarrollo de una programación, de interactuar de forma natural, con todo tipo de público. Escucha, ve, habla y se relaciona con el medio según se le haya programado. La complejidad de sus movimientos y acciones no tiene límites, desde jugar un partido de fútbol, hacer de profesor, o promocionar un producto en un evento interactuando con los asistentes o realizando complejas coreografías, entre otras muchas actividades. (Shamsuddin, Yussof, Ismail, & Mohamed, 2012)

Una herramienta de software poderosa muy utilizada en la actualidad para investigaciones sobre robots humanoides es ROS (Robot Operating System), la cual trabaja sobre el sistema Ubuntu y permite desarrollar código específico para el robot con el cual se desee. (Zhang, Jiang, Farid, & Hossain, 2013). Además se tiene otros sistemas propios que permiten la programación del Robot NAO bajo las prestaciones que estos ofrecen, es decir, permite implementar una gran variedad de funcionalidades como es el reconocimiento facial, reconocimiento de comandos de voz y orientarse

al manejo de componentes del robot, como son motores cámaras, sensores, micrófonos, etc. Permitiendo una comunicación bidireccional con el autómata.

## **1.2 Justificación e Importancia**

Durante los últimos años, en el área de la Interacción Humano-Robot (HRI), ha sido creciente el estudio de la interacción en la que participan usuarios no entrenados tecnológicamente con sistemas robóticos. Para esta población de usuarios potenciales, es necesario utilizar técnicas de interacción que no precisen de conocimientos previos específicos. En este sentido, al usuario no se le debe presuponer ningún tipo de habilidad tecnológica: la única habilidad interactiva que se le puede presuponer al usuario es la que le permite interactuar con otros humanos. (Robotronica, 2018) Las técnicas desarrolladas y expuestas en este trabajo tienen como finalidad, por un lado que el sistema/robot se exprese de modo y manera que los usuarios puedan comprenderlo, sin necesidad de hacer un esfuerzo extra con respecto a la interacción con personas. Por otro lado, que el sistema/robot interprete lo que esos usuarios expresen sin que tengan que hacerlo de modo distinto a como lo harían para comunicarse con otra persona. En definitiva, se persigue imitar a los seres humanos en su manera de interactuar enfocándose en un tema de entretenimiento donde se podrá contar con varias actividades o temas a realizar como: bailar, cantar, relatar cuentos o realizar chistes.

En el presente año, la Universidad de las Fuerzas Armadas ESPE, realizó la adquisición de un conjunto de robots NAO, para ser desarrollado en campo de la HRI en la Universidad, siendo importante para la ejecución de proyectos que permitan ampliar el nivel de “expertice” para el desarrollo en el ámbito de la robótica y aprovechar de estos nuevos recursos tecnológicos.

El Área de Automatización y Control es pionera en el uso de robots, sin embargo tiene poca experiencia en plataformas de tipo humanoide, es por ello que este proyecto aporta al desarrollo del campo HRI enfocándose en esta fase de la robótica social y de entretenimiento.

Para alcanzar los resultados planteado es necesario la realización de un conjunto básico de funciones que permitan la interacción entre el robot y el usuario. El sistema completo consta de diversos módulos (reconocimiento facial, comandos de voz, etc.), los cuales a través del desarrollo de algoritmos de programación personalizada permitirán trabajar de una manera coordinada y complementaria, trata de alcanzar los objetivos de interacción natural deseados; Sin olvidar que se convive dentro de una arquitectura de control robótica micro-procesada a través de un procesador Atom de 1.6 GHz que se comunica con el resto de sistemas que la componen, como son los subsistemas de: movimiento, expresiones corporales, reconocimiento, haciendo uso de los 25 actuadores y 18 sensores propios del robot NAO como son los motores de las articulaciones, cámaras, micrófonos, sensores infrarrojos, sensores táctiles y botones. Con los cuales se pretende ingresar al ámbito del entretenimiento

Debido a la complejidad de la interacción natural entre humanos y robots sociales, que involucra muchos campos de investigación resulta difícil encontrar un sistema general que cubra todos los aspectos que conciernen a la interacción humano-robot. Es a este desafío donde esta tesis pretende dar una respuesta, creando un sistema que facilita la interacción entre ambas partes sin la necesidad de ayuda externa.

El sistema humanoide NAO permite el desarrollo de bloques de funciones para realizar acciones sobre el NAO, para el desarrollo de estas funciones se requiere un nivel avanzado de programación basada en el lenguaje de programación C++, que permita integrar las herramientas disponibles en función del desarrollo del presente proyecto.

### 1.3 Alcance del Proyecto

Un robot humanoide de entretenimiento como “*compañero ideal con el que divertirse y aprender en casa*”, se trata de una programación desarrollada en el entorno académico de la Universidad de las Fuerzas Armadas “ESPE” basado en lenguaje de programación en C++ y que se incorpora de forma exclusiva en los robots NAO para permitir a los usuarios interactuar con el robot en nuestro propio idioma y ejecutar un conjunto básico de funciones.

En el modo entretenimiento el objetivo es que el robot puede suscitar emociones y motivar la realización de tareas, con una disposición e interacción amigable.

El sistema consiste en la implementación de una estrategia de la robótica social persuasiva mediante un conjunto de algoritmos de software, interacción con el medio a través de la adquisición de datos de sensores y actuadores. El robot humanoide NAO maneja información relevante mediante un algoritmo de comunicación interactiva con el fin de realizar propuestas de actividades de entretenimiento acorde a los gustos del individuo.

El proyecto tiene como finalidad desarrollar un sistema básico que permita llevar a cabo una interacción natural entre el humano y el robot humanoide NAO mediante el uso de algoritmos de reconocimiento facial, reconocimiento de comandos de voz, expresiones corporales desarrolladas para en entorno nacional.

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Desarrollar un conjunto básico de funciones que permitan la interacción humano-robot persuasivo e interactivo con un robot humanoide NAO

### **1.5.2 Objetivos Específicos**

- Desarrollar un sistema de interacción entretenida mediante lenguaje natural entre las personas y el robot humanoide NAO.
- Usar estrategias de robótica persuasiva para la interacción humano-robot.
- Implementar algoritmos de programación que le permita al robot NAO mantener una comunicación en lenguaje natural junto con la expresión corporal.

## **CAPÍTULO II**

### **ESTADO DEL ARTE**

#### **2.1 Introducción**

La robótica a través de los años ha tenido aplicaciones en áreas tan diversas y críticas como la medicina, la exploración planetaria y submarina, automatización de procesos industriales, seguridad, agricultura, transportes y servicios. (González E & Jiménez B, 2009). Siempre usándolos como apoyo para los humanos, viéndolos como una herramienta que cumple tareas específicas sin ninguna intención de interacción social, sin duda esto ha cambiado en los últimos años debido a diversas investigaciones que son realizadas por universidades y por empresas dedicadas a la tecnología, desarrollando avances en temas como la inteligencia artificial, robótica social, asistentes virtuales, robótica de interacción educativa y medicinal, que cambian la perspectiva del uso de los robots como una herramienta e incursionando en la comprensión de interacciones sociales, desarrollos cognitivos y psicopedagógicos.

El objetivo de las nuevas investigaciones con robots se enfoca en romper fronteras sociales y convertirlos en productos que entren masivamente en los hogares. (Gemignani, y otros, 2016) Brindando una visión de los robots como agentes inteligentes que son capaces de interactuar sin inconvenientes con humanos y operar en entornos reales, como lo es el caso del Robot ASIMO de la empresa Honda, el cual es un robot desarrollado para trabajar con un nivel de autonomía avanzada y basta movilidad, enfocándolo hacia la relación social con humanos.



**Figura 2.** Interacción social con el robot ASIMO  
Fuente: (Honda Motor Co, 2018)

Los nuevos avances tecnológicos abren caminos a la interacción mediante el uso de nuevos comandos de reconocimiento de voz, reconocimiento facial y dispositivos que mejoran su percepción para una mejor interacción con su entorno, pero aún existen diferencias entre las expectativas de los usuarios y la funcionalidad del robot, debido a la falta de adquisición de conocimiento y conciencia del robot, lo cual es un tema muy complejo aún, que se desarrolla en grandes centros de investigación tecnológica con resultados prometedores a futuros próximos. Nuevos estudio pretenden minimizar las diferencias entre robots y humanos, haciéndolos cada día más reales, con movimientos y gestos que involucran un gran nivel de ingeniería, pero además también, se centran en generar una comunicación y una interacción más natural, convincente y persuasiva.

Si se juntan características de interacción avanzadas se tiene un robot con un alto nivel de inteligencia considerado para la actualidad , debido a que pretende interactuar con el individuo de la manera más natural posible, pero además, conjuntamente se deben llevar a cabo ordenes o actividades de acuerdo al enfoque en el cual se encuentre su programación, es decir, que pueden

tener actividades en el ámbito educativo, de exploración o reconocimiento del entorno, culinarias, actividades diarias, que van desde ordenes simples, hasta ordenes como las de bailar o cantar proponiendo dinámicas entretenidas con una proyección hacia los más pequeños o el uso terapéuticos.

La robótica de entretenimiento se da a partir de grandes industrias con fines de lucro, como Disney con la fabricación de robots animados para usarlos en sus famosos parques de diversiones, otro ejemplo claro es Sony con la creación de Aibo para venderlo en forma masiva, el cual era un robot de entretenimiento en forma de perro y con funcionalidades muy avanzadas y precios para esa época accesibles para muy pocos; Pero eso es tema es más comercial, si se habla de robots de entretenimiento en el ámbito de investigación se puede mencionar varios trabajos que van desde enfoques sociales hasta los médicos, como el trabajo de B. Robins, K. Dautenhahn, R. Boekhorst, y A. Billard (2005), de la universidad de Hertfordshire U.K, los cuales analizan el impacto que tiene el exponer a cuatro niños con autismo de entre 5 a 10 años de edad, a una interacción social prolongada usando un robot en forma de muñeco, el cual habla, escucha y se mueve, creando un vínculo emocional que permitió que sean más espontáneos, proactivos y tengan una interacción divertida, ayudando así al niño a mantener relaciones sociales más duraderas.



**Figura 3.** Experimento de interacción entre un robot y Niños con autismo.  
Fuente: (Robins, Dautenhahn, Boekhorst, & Billard, 2005)

## 2.2 Interacción Humano-Robot (HRI)

La comprensión del lenguaje natural hablado en robots en contextos HRI tiene dos componentes críticos que lo distinguen de otras aplicaciones donde no necesita comprender el contexto de la comunicación sino solo la sintaxis, debido a la forma en que los humanos interactúan en lenguaje natural; Como primera parte los hablantes humanos esperan que el interlocutor exprese su comprensión durante el enunciado, por ejemplo, en forma de mirada fija, agradecimientos verbales como "bien" o "mhm", o mediante acciones como asentir con la cabeza, enviando una retroalimentación al emisor de su entendimiento ante el mensaje proporcionado; Como segunda parte cuando los oyentes y los hablantes comparten el mismo entorno, los hablantes esperan que integren de forma rápida e incremental el contexto perceptual como los cambios en tonos de voz o expresiones al hablar que pueden cambiar totalmente el contexto de una oración, por ello, si bien no existe un sistema robótico actual que incluso se acerque a la naturaleza HRI, varios esfuerzos

recientes han abordado diferentes aspectos de estos desafíos y avanzado el estado del arte en las interacciones del lenguaje natural de los robots con los humanos.

El campo de la interacción humano-robot (HRI) aborda el diseño, comprensión y evaluación de sistemas robóticos, que involucran a humanos y robots que interactúan a través de la comunicación. (Goodrich & Schultz, 2007). El tipo de interacción con un robot se las puede resumir en dos categorías: La interacción remota y la que se da por proximidad; El primer tipo hace referencia a la teleoperación, control supervisado, telemanipulación, mientras que la interacción por proximidad incluye ámbitos sociales, emotivas y aspectos cognitivos que tienen inmerso además comunicación verbal o corporal que incentiven o marquen pautas para la ejecución de acciones, actividades o trabajos para el cual el robot haya sido creado.

Según Goodrich & Schultz (2007) en su trabajo sobre interacción Humano-Robot definen cinco atributos que pueden afectar una interacción natural entre humanos y robots:

- Nivel de autonomía y comportamiento
- Intercambio natural de información
- Mecanismo de interacción
- Adaptación, aprendizaje y entrenamiento de las personas y los robots
- Capacidad de cumplir tareas.

Estos atributos confluyen con el objetivo de lograr una comprensión más estable, mayor relación y conocimiento de las funciones que están disponibles para una interacción más natural entre humanos y robots, junto con ello cabe considerar los componentes verbales y no verbales que le dan sentido y contexto a la comunicación, puesto que un dialogo seco y e inexpressivo no genera los suficientes niveles de empatía y confianza para continuar con la interrelación que se mantiene, llegándolo a postergar o a una ruptura, terminando la interacción.

En estudios realizados por Brenna D & Aude G (2010) se analiza el tipo de interacción humano robot que se tiene al incluir el contacto físico, donde se ve las conductas del humano ante las reacciones del robot al incluir el sentido del tacto, de acuerdo a ello se generan movimientos dependiendo de los datos captados por los sensores los cuales están distribuidos en gran parte de su estructura, además, se hace un análisis de la mejor manera en que se pueden colocar dichos sensores en la estructura del autómeta, dependiendo del recubrimiento que este tenga: carcasa dura ,sustrato flexible o sin recubrimiento analizando los cambios de datos y del comportamiento humano ante el contacto con el material.



**Figura 4.** Experimento de HRI con un sujeto de prueba.

Fuente: (Brenna & Aude, 2010)

En investigaciones sobre HRI realizados en la Universidad de Indiana por Rehj Cantrell, Matthias Scheutz, Paul Schermerhorn & Xuan Wu (2010) se describe el análisis efectuado al desarrollar una arquitectura integrada de comprensión del lenguaje natural para HRI, usando datos obtenidos de experimentos de la interacción humano-humano, demostrando las capacidades novedosas que presenta este modelo robótico, que integra elementos como el reconocimiento de

voz , procesamiento de imágenes y se demuestra su funcionalidad y rendimiento basados en datos de diálogo recopilados previamente con los experimentos entre humanos.

Las interacciones humano robot, no se dan únicamente mediante la comunicación verbal como se mencionó anteriormente, un claro ejemplo de ello es un estudio realizado en la Universidad Campinas de Brasil, por Eric Fujiwara, Danilo Miyatake, Murilo dos Santos, Carlos Kenichi (2013), donde se desarrolla un sensor que usa la tecnología de fibra óptica debido a sus características como ligereza, flexibilidad e inmunidad al ruido o interferencias, siendo muy adecuados para la implementación en esquemas HRI, enfocándose en una interacción a distancia o por teleoperación, creando un guante que mediante los movimientos de la mano envía señales al robot para realizar tareas específicas.

Otra investigación relacionada, realizada por Valiallah Monajjemi, Jens Wawerla, Richard Vaughan & Greg Mori (2013), presentan sistemas multi-robots basada en visión en tiempo real para crear, modificar y comandar vehículos aéreos no tripulados por un humano no preparado en la rama, donde mediante gestos de la mano basado en movimiento el humano lo puede agregar o retirar del equipo, otro gesto ordena al equipo ejecutar la tarea y donde los robots se comunican entre sí mediante una red inalámbrica para compartir datos.

### **2.3 Robótica social**

En el siglo veintiuno se ve grandes avances en la inclusión de robots sociales. Desde robots de limpieza (como el Roomba), robots de entretenimiento (como Aibo), mascotas robot (como KittyCat), muñecas robot (como Baby Alive), robots de terapia (como Paro) y muchos otros robots sociales, están encontrando rápidamente aplicaciones en hogares y en entornos de atención para personas mayores. (Scheutz, 2009).



**Figura 5.** Robot AIBO de Sony

Fuente: (Sony Corporation, 2018)

Esto nos hace ver que en algún momento y más pronto de lo que se espera se podrá contar con robots sociales en cada casa y se convertirán en una parte integral de las sociedades humanas muy similar a las computadoras y al internet, las cuales antes no se los encontraba mucho por su costo o debido a la falta de prestaciones que estas ofrecían, pero ahora son herramientas indispensables en el diario vivir de millones de personas; Por ello la robótica social seguirá una trayectoria similar ingresando en nuestras vidas personales y una vez que hayan sido aceptados en su totalidad por la sociedad la vida sin estas máquinas será inconcebible, lo que nos hace plantearnos cómo será la comunicación y la interacción con estos robots, si podrán en algún momento llegar a expresar emociones o empatía o simplemente serán herramientas deslingadas de cualquier tipo de nivel emocional o afectivo y el concepto de robótica social se quede únicamente en la capacidad que estos tengan de cumplir tareas enfocadas al beneficio humano.

Un modelo realizado por Rachel Kirby, Jodi Forlizzi & Reid Simmons (2010), sobre el comportamiento afectivo en robots sociales, donde dicho modelo intenta a crear una afectación natural similar a la humana e incluye distinciones entre las respuestas proporcionadas de acuerdo al estado de ánimo del robot el cual se muestra de forma aleatoria y que intenta imitar las emociones

que muestran las personas de la forma más natural posible y está diseñado para una interacción de largos periodos de tiempo con el objetivo de modelar los aspectos e interacciones a largo plazo entre las emociones, los estados de ánimo y las actitudes y como estas pueden influir significativamente en como las personas interactúan con el robot.

En otro estudio realizado en Japón por Takayuki Kanda, Takayuki Hirano & Daniel Eaton (2004) ,se examinó la proposición de que los robots podrían formar relaciones con los niños y que los niños podrían aprender de los robots a medida que aprenden de otros niños, usando la enseñanza de un idioma como base para el estudio; Los Robots usan un vocabulario pregrabado de alrededor de 300 oraciones y 50 palabras que se podían reconocer en inglés, el interés mostrado por el aprendizaje de este idioma creció entre los estudiantes, los cuales interactuaban de manera constante con los robots, aunque ese interés fue decayendo en el tiempo y la interacción con los robots se mantuvo con los niños que verdaderamente se encontraban interesados por el conocimiento del idioma Inglés, además se observó mejoras en su aprendizaje del idioma, lo cual sugiere que se proporcionó una herramienta fiable para la enseñanza.

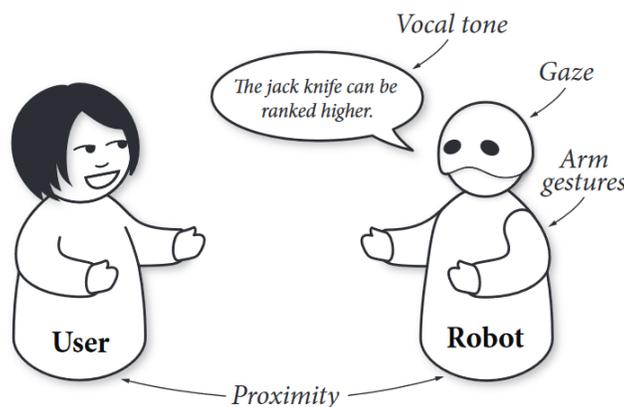
## **2.4 Robótica persuasiva**

La persuasión se puede definir generalmente como un intento de cambiar las creencias o el comportamiento de otra persona. El acto de influenciar a los demás es fundamental para casi cualquier tipo de interacción social. Cualquier agente que desee operar sin problemas de una manera social necesitará incorporar este tipo básico de comportamiento humano. Al igual que en la interacción humana, la apariencia y el comportamiento de un robot humanoide pueden alterar significativamente su capacidad de persuasión. (Siegel, Breazeal, & Norton, 2009).

Tanto la forma del robot como las características de movimiento que este posea lo hará más o menos apto para que pueda ser persuasivo, además del tipo de comunicación que se establezca entre el humano y el robot marcara las pautas necesarias para una interacción más afín y además si se añaden el uso de técnicas visuales se tendrá mayor éxito en el tipo de influencia que se tenga sobre un humano, estas técnicas o mejor llamadas estrategias de persuasión están muy relacionadas tanto con los gestos como con la mirada, debido a que estos factores infieren sobre el comportamiento o la toma de decisiones de los seres humanos, como en el estudio realizado por Jaap Ham, Raymond Cuijpers & John Cabibihan (2015), donde se midió el nivel de persuasión de un robot narrador, el cual tenía como tarea contar una historia persuasiva clásica sobre las consecuencias de mentir a cuarenta y ocho participantes, y se programó para usar gestos (persuasivos) (y no) y mirar (o no). Después cada participante evaluó el personaje de la historia de acuerdo al nivel de persuasión del robot, donde se obtuvo como resultado que la persuasión de un robot aumenta cuando se usa la observación. Cuando el robot usaba la mirada, su persuasión solo aumentaba cuando también esta venia acompañada de gestos. Cuando el robot no usaba gestos, disminuía el nivel de persuasión.

La comunicación también es un factor clave en el desarrollo de la robótica persuasiva, debido a que el habla es una forma de transmitir basta información y que influye sobre el comportamiento de las personas, en un estudio experimental sobre robótica persuasiva realizado por la universidad de Wisconsin Estados Unidos, por Chidambaram V, Chiang Y y Mutlu B (2012) se investiga los efectos de la toma de decisiones sobre treinta y dos participantes, usando un robot el cual mediante movimientos, gestos y expresiones tanto vocales como corporales, emite indicaciones para realizar acciones sobre un juego de supervivencia, se analiza el comportamiento del participante, sus

decisiones y acciones tomadas ante las indicaciones del robot, comparando que tipo de acción persuasiva genera mayor o menor impacto sobre los participantes.



**Figura 6.** Factores de persuasión para HRI

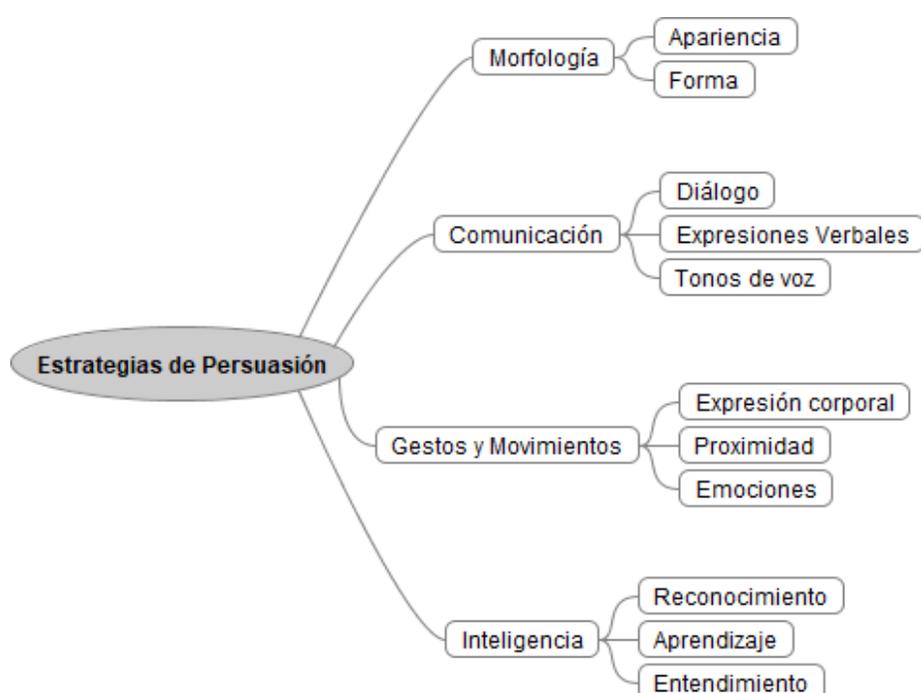
Fuente: (Chidambaram, Chiang, & Mutlu, 2012)

### 2.4.1 Estrategias de robótica persuasiva

La investigación en comunicación humana explica que la inmediatez no verbal de un individuo es un factor clave para persuadir a otros, es decir, que las expresiones corporales juegan un papel fundamental sobre los atributos conductuales de los individuos de acuerdo al grado de cercanía corporal y psicológica percibida, dentro de las expresiones se incluyen, particularmente señales corporales como la proximidad, la mirada, los gestos, la postura, las expresiones faciales, el tacto y las claves vocales, como el tono y las expresiones vocales. (Richmond, Gorham, & McCroskey, 1987).

De acuerdo a las diversas investigaciones que se han expuesto en este capítulo, se realiza un diagrama que muestra de manera resumida las estrategias de persuasión mencionadas, como se observa en la Figura 7, se destacan cuatro características fundamentales que son: la morfología del robot, debido a que la forma y apariencia influye sobre la capacidad de interacción en una persona, la forma de comunicación, al momento de establecer un diálogo, los gestos y movimientos usados

que puedan expresar indicaciones, señas o emociones y el nivel de inteligencia, es decir aprender, entender y reconocer aspectos o gustos del usuario, son características que influyen en gran medida a que se establezca una comunicación persuasiva, por lo tanto serán consideradas en este trabajo, a excepción del tipo de persuasión por modulación en tonos de voz, la cual por su complejidad de reconocer el contexto del diálogo para modular el tono de voz no será considerado.



*Figura 7.* Estrategias de persuasión.

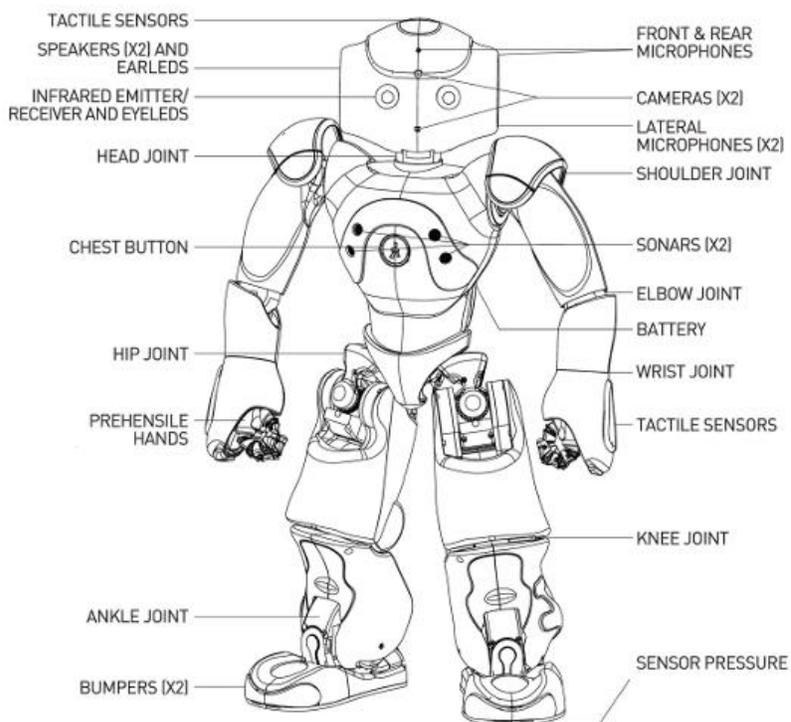
## CAPÍTULO III

### SISTEMA DE DESARROLLO DE APLICACIONES PARA EL ROBOT

#### NAO

##### 3.1 Introducción

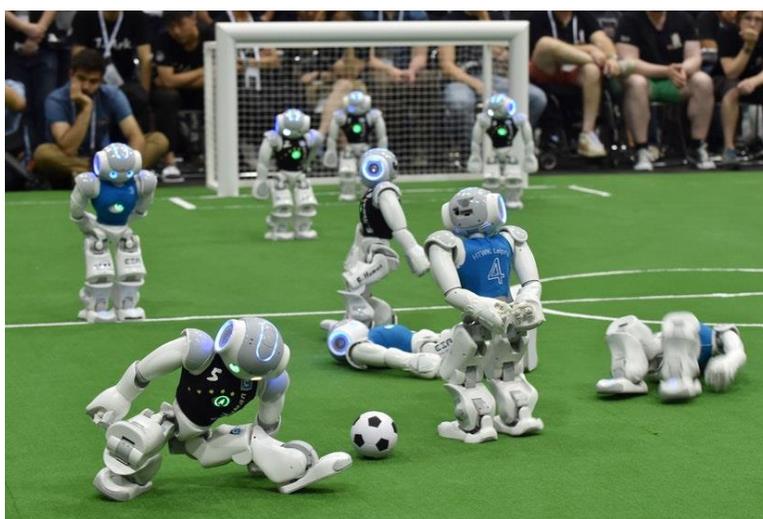
El robot humanoide NAO, es un robot interactivo que es totalmente programable y que su evolución data desde el año 2008, desarrollado por la empresa francesa Aldebarán Robotics ahora conocida como SoftBank Robotics, este robot se ha venido desarrollando durante varios años, aumentando mejoras y el nivel de procesamiento, hasta llegar a la versión actual v5, su forma y elementos se muestran en la Figura 8.



**Figura 8.** Elementos de la plataforma robótica NAO

Fuente: (Aldebaran Robotics, 2018)

Con la creación del Robot NAO, robots como AIBO de Sony (véase en la Figura 5), fueron desplazados de sus ramas de investigación y se marcó como el sucesor en la liga de fútbol Robocup como una plataforma estándar; Esta competencia universitaria que organiza partidos de fútbol entre robots tiene el objetivo de tener un equipo que pueda competir directamente contra humanos en el año 2050.



**Figura 9.** Torneo Robocup realizado con robots NAO

Fuente: (RoboCup Federation, 2018)

El Robot NAO permite un extenso rango de programación que admite la personalización de funciones del robot y adaptarlo a cualquier entorno, ya sea para el ámbito educativo, motivando a la participación de los estudiantes en la realización de tareas o a niveles más avanzados para incursionar en la programación, siendo de gran ayuda para los profesores también al ayudarlos a dinamizar la clase, aumentando los niveles de interés y concentración, en ámbitos de entretenimiento, con demostraciones de baile y diversas coreografías y el uso de su lenguaje corporal para llamar la atención y establecer formas de comunicación más natural con los Humanos; Otro campo muy explotado es el que usan las empresas que obtienen este tipo de robot humanoide, debido a que lo usan en proyectos de Marketing y comunicación institucional

atrayendo las miras de todo el público por su forma innovadora y sus capacidades interactivas; Además no se puede dejar atrás que el robot NAO se destaca en el ámbito científico y de investigación para el desarrollo de nuevas aplicaciones prácticas, experimentos técnicos y sociales donde incursiona la robótica, siendo muy famoso entre las universidades de investigación a nivel mundial.

El robot NAO cuenta con varios sensores que le permiten interactuar de diversas formas con su entorno, la cantidad de los sensores y equipos que dispone el robot se describen en la Tabla 1.

**Tabla 1**

*Cantidad y tipos de sensores del robot humanoide NAO*

<b>Cantidad</b>	<b>Elemento</b>
2	Cámaras de alta definición
4	Micrófonos
8	Sensores de presión
9	Sensores táctiles
2	Sensores Ultrasónicos
1	Acelerómetro
1	Giroscopio
53	Leds RGB
2	Altavoces

Fuente: (Aldebaran Robotics, 2018)

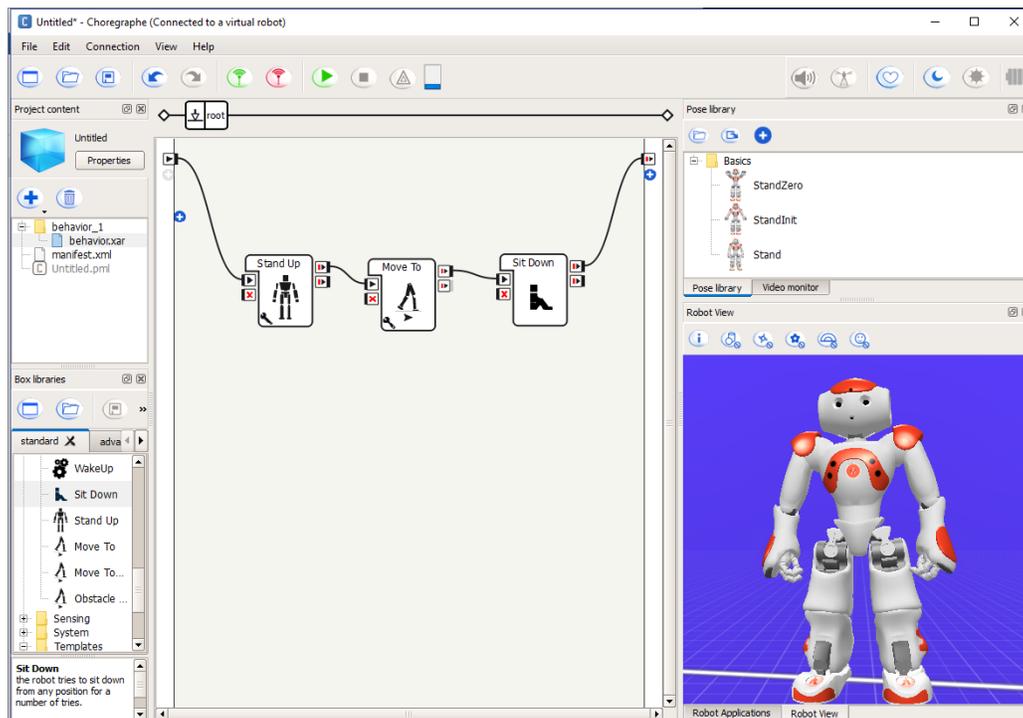
Además de los sensores y equipos detallados, se puede destacar más elementos característicos del Robot NAO, el cual cuenta con un procesador Intel Atom de 1.6 GHz en el cual se corre un Kernel de Linux y el middleware de Aldebarán llamado NAOqi, el cual es el sistema operativo del

robot, el cual es capaz de cumplir necesidades avanzadas de programación como es el paralelismo, manejo de múltiples recursos, sincronización y eventos, comparte funciones y librerías para que sea compatible con su programación, de entre las cuales se pueden mencionar el Software gráfico de programación propia de la empresa llamado Choregraphe cuyo logo se muestra en la Figura 10, el cual permite programarlo sin tener conocimientos propios de un lenguaje de programación al ser muy intuitivo y sencillo de usar debido a su nivel de programación gráfica por bloques, donde cada bloque contiene una librería desarrollada que realiza una función en específico, pero que se puede conectar a otros bloques formando diagramas de flujo complejos que ejecuten varias actividades del robot a la vez, su entorno de programación se puede apreciar en la Figura 11, donde se muestra un ejemplo sencillo de la forma y los bloques de programación. Y para los usuarios que se permitan contribuir con nuevas características en investigación incluye un conjunto completo para el desarrollo de software, que permite usar distintos lenguajes como C++, Python, JAVA.NET y Matlab.



**Figura 10.** Logo del programa Choregraphe.

Fuente: (Aldebaran Robotics, 2018)



*Figura 11.* Ejemplo de programación y entorno de Choregraphe

Una herramienta muy conocida para la escritura de software de robots es ROS (Robot Operating System), el cual es compatible para trabajar con el robot humanoide NAO, debido al amplio desarrollo que este robot tiene de su driver NAOqi, mediante ROS se puede acceder a cualquier elemento, sensor o actuador del robot, funciona mediante “**nodos**” que se comunican entre sí compartiendo información usando un modelo de mensajería publicador/suscriptor, mediante comandos propios para cada tipo, además se puede emplear lenguajes de programación conocidos como C++ y Python de acuerdo a las necesidades e investigaciones que se realicen.

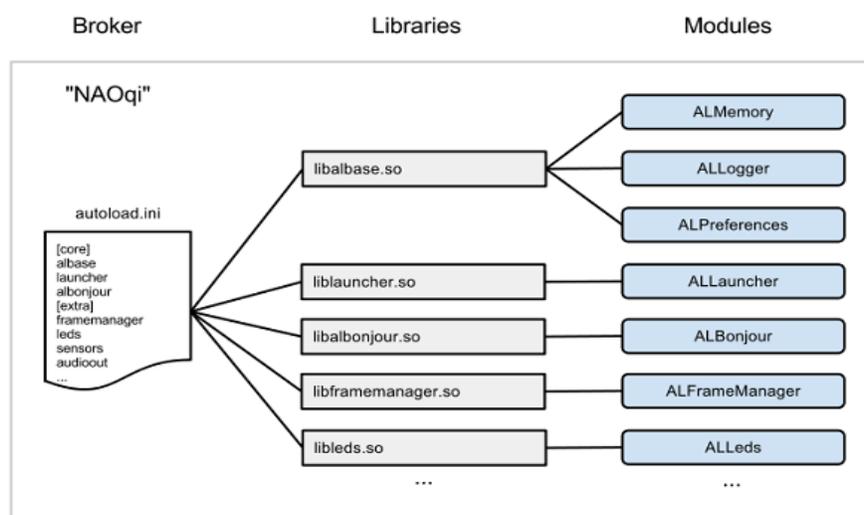
Otra herramienta que permite la escritura de software diseñada para los Robots de Aldebaran Robotics es QIBuild, la cual es compatible al cien por ciento para trabajar con NAOqi, debido a que se lo desarrollo en base a este sistema, además de permitir desarrollos e investigaciones nuevas usando librerías de plataformas de código como CMake, gcc o Visual Studio y siendo compatible con herramientas como OpenCV, facilitando el trabajo para investigadores al momento de acceder

a los componentes, sensores y actuadores de los robots, al momento de incluir librerías externas o trabajar con software de terceros.

### 3.2 Estructura de NAOqi

NAOqi es software principal que corre sobre el robot, permite su funcionamiento y control, además sirve como driver para ser ejecutado en un ordenador y realizar pruebas que se puedan hacer sin la necesidad del robot mediante el uso de algún simulador.

El software NAOqi, permite buscar las librerías que se deben cargar cuando se ejecuta un programa o cuando se inicia el robot, a este ejecutable que corre en el robot se lo conoce como “broker”. Dentro de todo el sistema existe una gran biblioteca que contiene todos los módulos del robot a los que se puede acceder y que permiten buscar las funciones específicas del robot, y su estructura se muestra en la *Figura 12*.

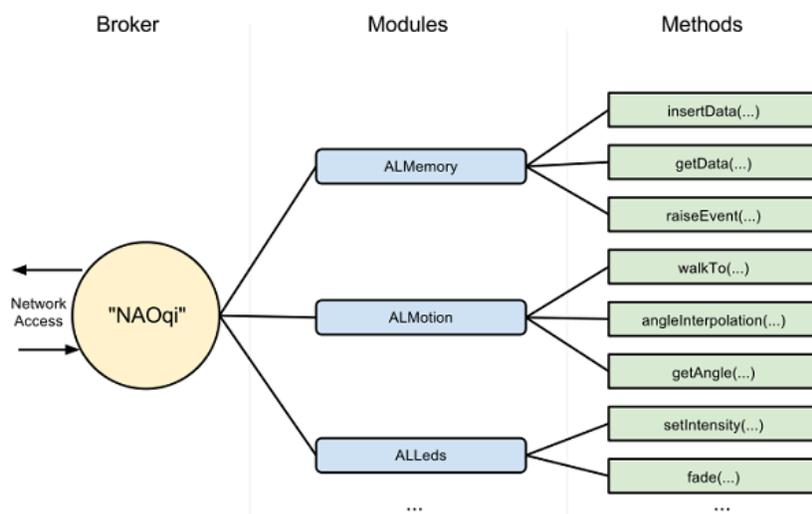


**Figura 12.** Estructura de NAOqi: Broker, librerías y módulos.

Fuente: (Aldebaran Robotics, 2018)

Cada módulo que contiene NAOqi está desarrollado de tal forma que permita un acceso a todo el sistema del robot y puede ser desde los datos de los sensores hasta funciones ya desarrolladas

como el reconocimiento de voz que desentendiendo la característica del módulo se puede acceder a los métodos para poder modificar las características de cada módulo de acuerdo a las necesidades del desarrollador y un ejemplo de la estructura se puede apreciar en la **Figura 13**.



**Figura 13.** Estructura de NAOqi: Broker, Módulos y Métodos.

Fuente: (Aldebaran Robotics, 2018)

### 3.3 Herramientas de ROS Y QIBUILD

Como ya se mencionó los dos sistemas me permiten trabajar con el driver y sistema propio del Robot NAO, haciéndolos muy compatibles para desarrollo e investigación, cumpliendo funciones similares, con el hecho de que QIBuild se lo creo específicamente para los robots que tengan el sistema NAOqi y no como ROS que es un sistema universal para cualquier tipo de robots que tenga un driver de desarrollo.

El sistema de ROS es de código abierto y no tiene preferencias por algún lenguaje de programación, por tanto, se pueden generar códigos en C++, Python u otros sin problemas pues dispone de las herramientas necesarias para la compilación y ejecución de estos, una de sus filosofías es la reutilización de códigos de manera que se pueden adicionar fácilmente librerías de

los repositorios (Serrano, 2011), el funcionamiento de ROS está basado en la ejecución de un conjunto de “nodos”, los cuales se comunican entre ellos con el fin de compartir información con un modelo de mensajería publicador/suscriptor, para ello debe existir un master el cual comunica los nodos y proporciona la información de la conexión que existen entre estos para permitir la transmisión de los mensajes mediante los “tópicos”, el cual funciona como canal de envío de información, de esta manera un nodo puede publicar o suscribirse a un tópico sin importar que existan otros nodos conectados a él pero de forma unidireccional, de lo contrario para tener una comunicación síncrona bidireccional se debe recurrir a los “servicios” que son los que permite la petición y respuesta de alguna información.



**Figura 14.** Ícono del sistema operativo de Robots (ROS).

Fuente: (ROS (Robot Operating System), 2018)

QIBuild también es un Sistema bastante desarrollado que ayuda a la gestión de proyectos y sus dependencias, trabaja con CMake el cual es una herramienta diseñada para construir, probar y empaquetar el software, le ayuda con la compilación y la búsqueda de errores en el código; Para empezar con QIBuild se debe conocer cómo trabaja, para empezar realiza un proyecto con un árbol de trabajo o “worktree” que crea una lista de archivos, que son objetos simples y necesarios para la ejecución del proyecto, en el cual se incluye el archivo de tipo “.cpp” y el archivo “CMakeList.txt”; El primero es el que contendrá todo el código que va a ser ejecutado del programa principal o subprogramas que se llamen de un programa principal y el segundo me permite agregar rutas, librerías y archivos extras que deseen ser partícipes del proyecto o sean necesarios para su

ejecución, QIBuild trabaja extrayendo la información de las rutas especificadas en el CMakeList que pueden conducir a las librerías o paquetes externos.

Para trabajar con cualquiera de estos dos sistemas se utilizó como sistema operativo Ubuntu 14.04 LTS de 64 bits, debido a su compatibilidad con ROS Índigo y este a su vez con NAOqi 2.1.4.13, esta integración evita fallas en el sistema y la compilación del código, otro aspecto es que Ubuntu 14.04 también es compatible para la instalación, manejo y ejecución de QIBuild.



*Figura 15.* Iconos correspondientes de Ubuntu y ROS Índigo.  
Fuente: (Ubuntu homepage, 2018) (ROS (Robot Operating System), 2018)

### **3.4 Instalación de Driver NAO para los sistemas ROS y QIBUILD.**

#### **3.4.1 Ejecución de ROS y comandos utilizados.**

Para la ejecución de los comandos de ROS se debe establecer un puente de comunicación entre este sistema y NAOqi, es decir, el driver propio del robot NAO, el cual contiene los tópicos al cual se podrá tener acceso en un futuro y poder ejecutar acciones para el robot.

Una vez instalado el sistema ROS, se procede a instalar el driver del robot NAO, al estar trabajando en un entorno de Ubuntu, la ejecución de los comandos para dicha instalación se llevan a cabo por medio del “Terminal”, ahora se procede a ejecutar el siguiente comando:

```
$ sudo apt-get install ros-.*-naoqi-driver
```

Otra forma existente de instalar el driver de NAO para ROS, muy usada para desarrolladores que requieran versiones anteriores al driver establecido, es mediante repositorios “github” de terceros, para ello cabe recalcar que al momento de instalar ROS se debe trabajar con un generador de código, el más común se lo denomina “Catkin”, contiene su propio espacio de trabajo y me ayuda a compilar, revisar y generar el código con el que se esté trabajando. Antes de instalar el driver es necesario establecer una carpeta que contendrá el espacio de trabajo de Catkin, para ello se puede ejecutar los siguientes comandos:

```
# Crear un espacio de trabajo para catkin
$ mkdir -p ~/catkin_ws/src
# Ingresar al espacio de trabajo creado
$ cd ~/catkin_ws/src
```

Una vez establecido el espacio de trabajo del constructor de código, se puede instalar el driver desde un repositorio github, como se muestra a continuación:

```
# Instalar el driver desde el repositorio elegido
$ git clone https://github.com/ros-naoqi/naoqi_driver.git
# Instalar las dependencias del driver
$ rosdep install -i -y --from-paths ./naoqi_driver
# Construir y ejecutar los cambios
$ source /opt/ros/indigo/setup.sh
$ cd ../ && catkin_make
```

### 3.4.2 Ejecución de QIBUILD y comandos utilizados.

Para una correcta instalación del driver NAOqi para QIBuild es necesario tener instalados herramientas que sirven de base como Python en la versión 2.7 y como ya se mencionó CMake, los cuales brindan paquetes y librerías necesarias para la compilación de código. Una vez preparado eso se procede con la instalación del driver mediante el siguiente comando.

```
$ pip install qibuild
```

QIBuild se encarga también de compilar, revisar y generar el código por lo que al igual que Catkin debe contar con un espacio de trabajo en el cual se creen todos los proyectos que necesiten de esta herramienta para su ejecución.

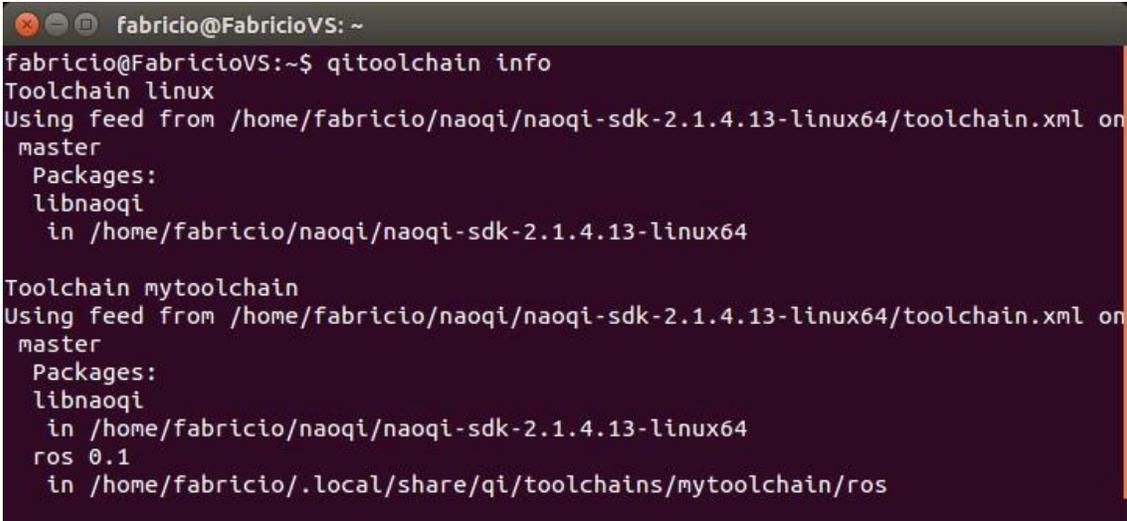
```
# Crear un espacio de trabajo para qibuild
$ mkdir ~/qibuild_ws
# Ingresar al espacio de trabajo creado
$ cd ~/qibuild_ws
#Inicializar como espacio de trabajo
$ qibuild init
```

QIBuild cuando se compila, busca dependencias dentro de una cadena de herramientas llamadas “qitoolchain” que usa paquetes pre-compilados que pueden ser de Aldebarán, propios o de terceros, pudiendo así también, usar paquetes propios de ROS y fusionarlos si se requiere para alguna aplicación o desarrollo de nuevas investigaciones.

Para configurar qitoolchain usando las herramientas por defecto que se han desarrollado, se debe buscar dentro del driver NAOqi el archivo “toolchain.xml” y dentro de la carpeta en la que se creó el espacio de trabajo de QIBuild proceder con los siguientes comandos:

```
$ qibuild init --interactive
$ qitoolchain create linux <path/to/cpp/sdk>/toolchain.xml
$ qitoolchain create mytoolchain <path/to/cpp/sdk>/toolchain.xml
$ qibuild add-config mytoolchain -t mytoolchain
$ qitoolchain info
```

Con la línea de comando final se podrá verificar la correcta instalación de la cadena de herramientas, la versión en la que se encuentra y las carpetas de dependencias como se muestra en la Figura 16.



```
fabricio@FabricioVS: ~
fabricio@FabricioVS:~$ qitoolchain info
Toolchain linux
Using feed from /home/fabricio/naoqi/naoqi-sdk-2.1.4.13-linux64/toolchain.xml on
master
Packages:
libnaoqi
in /home/fabricio/naoqi/naoqi-sdk-2.1.4.13-linux64

Toolchain mytoolchain
Using feed from /home/fabricio/naoqi/naoqi-sdk-2.1.4.13-linux64/toolchain.xml on
master
Packages:
libnaoqi
in /home/fabricio/naoqi/naoqi-sdk-2.1.4.13-linux64
ros 0.1
in /home/fabricio/.local/share/qi/toolchains/mytoolchain/ros
```

*Figura 16.* Información sobre qitoolchain.

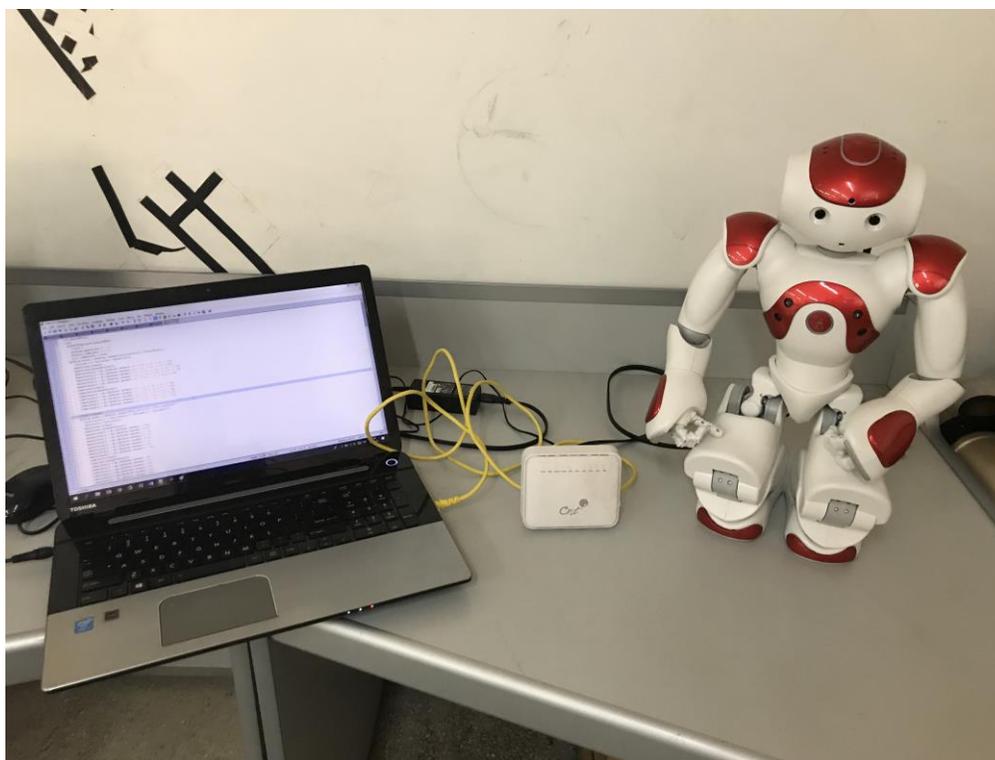
### 3.5 Comunicación con el robot humanoide NAO.

El robot NAO cuenta con una tarjeta de red que le permite la conexión mediante Ethernet, wlan o VPN, las cuales se pueden usar para transmitir datos desde el ordenador al robot y viceversa. En

primera instancia el robot debe conectarse a una red establecida mediante un punto de acceso, con el fin poder acceder a sus funciones básicas de conexión y observar las redes disponibles y unirse alguna de ellas, como se observa en la Figura 17, donde se representa el diagrama de conexión para el robot NAO.



**Figura 17.** Forma de conexión ordenador-robot.



**Figura 18.** Conexión real entre el ordenador y el robot NAO.

### 3.5.1 Comunicación entre NAO y ROS.

Para establecer la comunicación con el Robot NAO se debe especificar la interfaz de red en la que se desee trabajar, ya sea Ethernet, wlan o VPN e involucrar el ordenador que contiene ROS a una misma red, conociendo las direcciones IP establecidas se procede a ejecutar cualquier de los siguientes comandos, dependiendo de la interfaz de red.

```
$ Roslaunch naoqi_driver naoqi_driver.launch nao_ip:= <Robot_IP> roscore_ip:=  
<roscore_ip> network_interface:= <eth0 | wlan0 | tethering | vpn ...>
```

Un ejemplo del comando ejecutado, usando una interfaz de red Ethernet, se muestra a continuación, donde se establece nao\_ip con la dirección IP del robot y roscore\_ip con la dirección ip del ordenador; El resultado obtenido ante un evento de conexión exitosa se muestra en la Figura 19.

```
$ roslaunch naoqi_driver naoqi_driver.launch nao_ip:=130.130.130.203  
roscore_ip:=130.130.130.202 network_interface:=eth0
```

```

/home/fabricio/catkin_ws/src/naoqi_driver/launch/naoqi_driver.launch http://localhost:113
fabricio@FabricioVS:~$ roslaunch naoqi_driver naoqi_driver.launch nao_ip:=130.13
0.130.202 roscore_ip:=130.130.130.203 network_interface:=eth0
... logging to /home/fabricio/.ros/log/f1b95746-6f12-11e8-9ab4-089e01c65bd6/rosl
aunch-FabricioVS-3158.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://FabricioVS:38869/

SUMMARY
=====

PARAMETERS
* /rosdistro: indigo
* /rosversion: 1.11.21

NODES
/
  naoqi_driver (naoqi_driver/naoqi_driver_node)

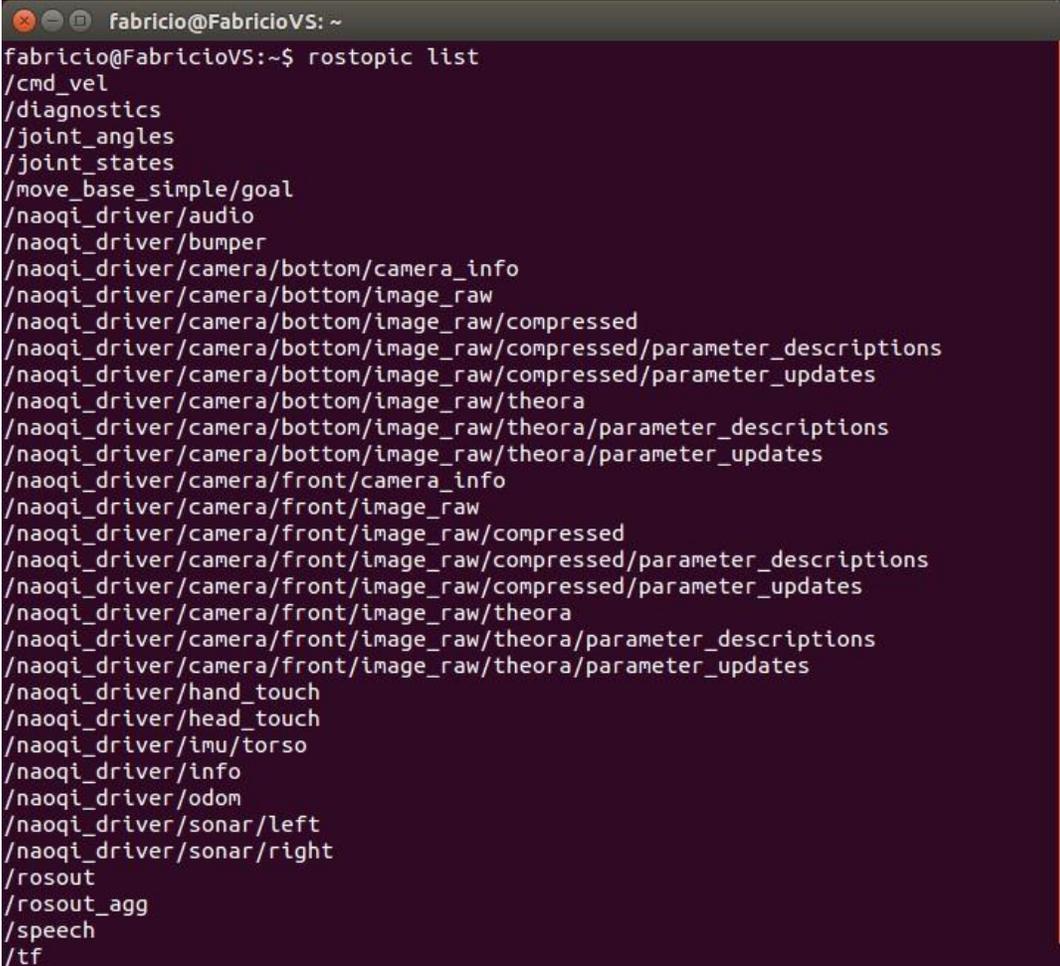
auto-starting new master
process[rosmaster]: started with pid [3170]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to f1b95746-6f12-11e8-9ab4-089e01c65bd6
process[rosout-1]: started with pid [3183]
started core service [/rosout]
process[naoqi_driver-2]: started with pid [3187]
Receiving information about robot model
[W] 3187 qimessaging.remoteobject: Return signature might be incorrect depending
on the value, from m to s

```

*Figura 19.* Conexión exitosa entre ROS y el robot NAO

Una vez establecido la conexión es posible comprobar la lista de tópicos que están disponibles para el robot NAO como se puede observar en la Figura 20.



```
fabricio@FabricioVS: ~
fabricio@FabricioVS:~$ rostopic list
/cmd_vel
/diagnostics
/joint_angles
/joint_states
/move_base_simple/goal
/naoqi_driver/audio
/naoqi_driver/bumper
/naoqi_driver/camera/bottom/camera_info
/naoqi_driver/camera/bottom/image_raw
/naoqi_driver/camera/bottom/image_raw/compressed
/naoqi_driver/camera/bottom/image_raw/compressed/parameter_descriptions
/naoqi_driver/camera/bottom/image_raw/compressed/parameter_updates
/naoqi_driver/camera/bottom/image_raw/theora
/naoqi_driver/camera/bottom/image_raw/theora/parameter_descriptions
/naoqi_driver/camera/bottom/image_raw/theora/parameter_updates
/naoqi_driver/camera/front/camera_info
/naoqi_driver/camera/front/image_raw
/naoqi_driver/camera/front/image_raw/compressed
/naoqi_driver/camera/front/image_raw/compressed/parameter_descriptions
/naoqi_driver/camera/front/image_raw/compressed/parameter_updates
/naoqi_driver/camera/front/image_raw/theora
/naoqi_driver/camera/front/image_raw/theora/parameter_descriptions
/naoqi_driver/camera/front/image_raw/theora/parameter_updates
/naoqi_driver/hand_touch
/naoqi_driver/head_touch
/naoqi_driver/imu/torso
/naoqi_driver/info
/naoqi_driver/odom
/naoqi_driver/sonar/left
/naoqi_driver/sonar/right
/rosout
/rosout_agg
/speech
/tf
```

*Figura 20.* Lista de tópicos disponibles para el robot NAO.

### 3.5.2 Comunicación entre NAO y QIBUILD.

En este caso para establecer la conexión con el robot, se parte de la creación de un nuevo proyecto, que debe estar contenido dentro del espacio de trabajo antes creado para QIBuild y ejecutar el siguiente comando el cual crea un árbol de trabajo, con los archivos necesarios para su compilación, como se puede observar en la Figura 21.

```
$ qisrc create test1
```



**Figura 21.** Archivos creados para un proyecto de QIBUILD.

Seguidamente se procede a configurar y construir el proyecto, con el fin de crear los archivos necesarios para su ejecución, en el caso de no existir problemas se tendrá algo similar a lo que se muestra en la Figura 22.

```
$ qibuild configure test1  
$ qibuild make test1
```

```

fabricio@FabricioVS:~/naoqi/qibuild_ws/test1$ qibuild configure -c mytoolchain
Current build worktree: /home/fabricio
Using toolchain: mytoolchain
* (1/1) Configuring test1
-- The C compiler identification is GNU 4.8.4
-- The CXX compiler identification is GNU 4.8.4
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Using qibuild 3.11.17
-- Binary: test1
-- Binary: test_test1
-- Configuring done
-- Generating done
-- Build files have been written to: /home/fabricio/naoqi/qibuild_ws/test1/build-mytoolchain

fabricio@FabricioVS:~/naoqi/qibuild_ws/test1$ qibuild make -c mytoolchain
Current build worktree: /home/fabricio
Using toolchain: mytoolchain
* (1/1) Building test1 in Debug
Scanning dependencies of target test1
[ 50%] Building CXX object CMakeFiles/test1.dir/main.cpp.o
Linking CXX executable sdk/bin/test1
[ 50%] Built target test1
Scanning dependencies of target test_test1
[100%] Building CXX object CMakeFiles/test_test1.dir/test.cpp.o
Linking CXX executable sdk/bin/test_test1
[100%] Built target test_test1
fabricio@FabricioVS:~/naoqi/qibuild_ws/test1$ █

```

*Figura 22.* Configuración y construcción básica de un proyecto con mytoolchain

Una vez configurado el proyecto se crea una nueva carpeta llamada “build-mytoolchain” que contiene el archivo ejecutable dentro de la carpeta “bin” la cual puede ser llamada desde el terminal junto con la dirección IP del robot y el puerto de comunicación el cual por defecto está establecido el “9559”, obteniendo como resultado una salida como se muestra en la Figura 23.

```

fabricio@FabricioVS: ~/naoqi/qibuild_ws/test1/build-mytoolchain/sdk/bin
fabricio@FabricioVS:~/naoqi/qibuild_ws/test1/build-mytoolchain/sdk/bin$ ./test1
130.130.130.202 9559
[W] 4830 alcommon.alproxy: ALProxy(ModuleName, IP, Port) is DEPRECATED. Create a
broker yourself and ask it for a proxy.
[I] 4830 qimessaging.session: Session listener created on tcp://0.0.0.0:0
[I] 4830 qimessaging.transportserver: TransportServer will listen on: tcp://130.
130.130.203:39873
[I] 4830 qimessaging.transportserver: TransportServer will listen on: tcp://127.
0.0.1:39873
[I] 4830 qimessaging.transportserver: TransportServer will listen on: tcp://130.
130.130.109:39873
Connection Successful
fabricio@FabricioVS:~/naoqi/qibuild_ws/test1/build-mytoolchain/sdk/bin$

```

*Figura 23.* Ejecución de un programa usando QIBuild.

### 3.6 Selección del sistema.

Para continuar con el trabajo se realiza una pequeña comparación entre el uso del sistema ROS y QIBuild, con el fin de determinar la mejor vía para el desarrollo del presente proyecto, teniendo en cuenta los niveles de complejidad y aplicaciones a implementar.

Como se describe el sistema ROS permite acceder a cualquier tipo de robot y programarlo en distintos lenguajes de programación dependiendo el tipo de aplicación que se requiera, usar librerías y código de terceros que ayuden al desarrollo e implementación de algún tipo de proyecto sobre cualquier plataforma robótica al cual se le ha creado en base a un driver de desarrollo, su tipo de comunicación es por Nodos siguiendo un modelo de mensajería publicador/suscriptor, en el cual es indispensable conocer a que tópico es necesario acceder y la forma de hacerlo.

Por otro lado tenemos al sistema desarrollado específicamente para los robots de Aldebarán, el cual cumple con la misma función para la cual fue desarrollada ROS, desde QIBuild se permite el acceso al sistema, actuadores, sensores y controles del robot de una forma más sencilla, sin tener que preocuparse de la comunicación con los elementos, debido a que es posible enlazarlos

únicamente mediante las librerías y usar los módulos y métodos ya desarrollado que son de gran utilidad para el desarrollo en este proyecto.

Con el fin de mostrar las características existentes entre los dos sistemas se realiza una comparativa para destacar al sistema que mejor se adapte al desarrollo del presente proyecto.

**Tabla 2.**

*Características de los sistemas QIBuild y ROS*

Características	QIBuild	ROS
Comunicación Ethernet, wlan y vpn.	Si	Si
Creación de paquetes redistribuibles y pre-compilados	Si	Si
Compilación cruzada con el robot.	Si	Con el uso de librerías extras de CMake y paquetes adicionales.
Uso de dependencias de terceros	Si	Si
Soporte de varios lenguajes de programación	Si	Si
Constructor de Código	Qibuild (qitoolchain)	Catkin
Búsqueda de librerías	Automáticamente	Especificando el tipo de librería
Módulos avanzados (reconocimiento de voz, rostro, etc.)	Incluido	Debe ser implementado basado en el programa del desarrollador.

Como se observa en la Tabla 2, las características de los dos sistemas planteados para usarlo como plataforma de desarrollo con el fin de programar al robot NAO, muestran las ventajas de cada uno de estos, pero se puede observar que ROS, es un sistema muy global que se diseñó para trabajar sobre cualquier robot que posea un driver de desarrollo, funcionando mediante un modelo en el cual es necesario un master como “roscore” que siempre debe estar ejecutándose para que los

nodos puedan comunicarse entre ellos, mientras QIBUILD brinda una mayor comodidad para trabajar con los robots creados por SoftBank Robotics, debido a la facilidad de comunicación que se da entre el programa y los componentes del robot, a los cuales se los puede acceder de manera rápida sin tener un servicio master de por medio, además contiene algoritmos modulares y librerías propias para este tipo de robots, permitiendo optimizar los recursos de software y facilitando la comunicación con el hardware, por ello se considera a QIBUILD como la plataforma de desarrollo para el presente proyecto.

## CAPÍTULO IV

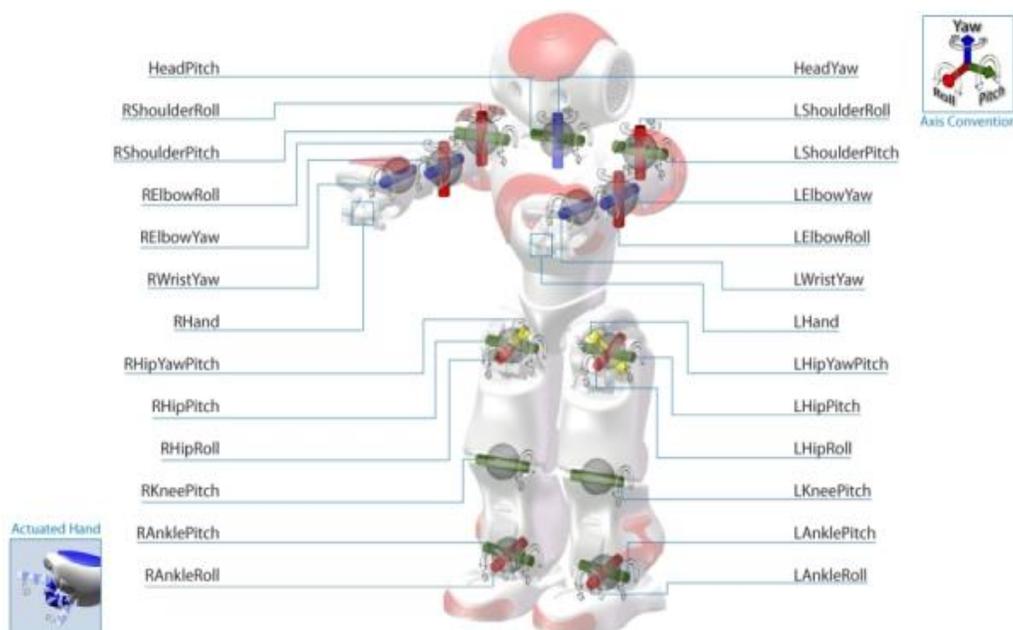
### ALGORITMOS DE INTERACCIÓN HUMANO-ROBOT

#### 4.1 Expresión corporal y movimientos planificados del Robot

En esta sección se va a presentar los módulos que se usan para el desarrollo del movimiento del robot orientados al lenguaje de programación C++, como se conoce el robot humanoide NAO puede ejecutar programas en este tipo de lenguaje de programación, y cabe destacar que es posible hacerlo de forma local o remota, siempre y cuando la comunicación sea por red, por ello es necesario conocer la dirección IP del robot NAO que se va a utilizar remotamente y estar enlazados en la misma red o en caso de que sea local usar la dirección IP 127.0.0.1.

Para generar el movimiento en el robot NAO, se debe tener en cuenta cada ángulo de rotación de los motores que se encuentran distribuidos sobre todo el robot, siendo como sus articulaciones y dando los grados de libertad que posee el NAO, de acuerdo a los ángulos de rotación y la a función de cada una de las articulaciones, estas pueden tener el complemento Yaw, Pitch y Roll.

Todos los nombres de las articulaciones disponibles para el robot NAO se pueden observar en la Figura 24.



**Figura 24.** Articulaciones del robot NAO.

Fuente: (Aldebaran Robotics, 2018)

#### 4.1.1 Módulo ALMotion

El módulo encargado de todo el movimiento del robot NAO es llamado ALMotion, el cual permite el control y manipulación de todos los motores que forman parte de cada una de las articulaciones del robot, trabajando a una frecuencia de 50 Hz o de 20 ms. Siempre que se solicite realizar un movimiento a través de este módulo, es creada una tarea llamada “*motion.task*”, encargada de controlar y realizar en manejo de la posición designada para cada motor. Con el fin de evitar inconsistencia en los comando ante la ejecución de varios movimientos que se den al mismo tiempo, los movimientos se deben declarar en una matriz que servirá para encapsular la tarea a ejecutar.

Este módulo proporciona cuatro grupos de métodos que permiten controlar diferentes aspectos relacionados con los movimientos del robot o el estado de los motores.

- Rigidez en las articulaciones. Este método controla el par en los motores, para definir qué tan

firmes deben estar de acuerdo al nivel en el que se los haya definido. Si la rigidez de las articulaciones se establece en 0.0, no se controla nada y la articulación es libre, por el contrario si se define un valor de 1.0 la articulación utiliza la energía de torsión completa con el fin de llegar a una posición dada y es posible utilizar valores intermedios de acuerdo al nivel de rigidez con la que se trabaja pero teniendo en cuenta que entre más bajo es el valor no llegará a la posición deseada.

- Posición de las articulaciones. Cada articulación puede ser controlada individualmente o paralelamente con otras articulaciones, estas formas de control de las articulaciones se las puede realizar de dos maneras diferentes, la primera se realiza mediante métodos de animación mediante la interpolación de ángulos y en un determinado tiempo, pudiendo así variar la velocidad de acuerdo al tiempo que se debe demorar en llegar a su posición y la segunda mediante métodos reactivos, donde el ángulo de movimiento se puede cambiar en cada ciclo del método ALMotion, donde la trayectoria es suave en su posición y continua en velocidad, por ejemplo para generar el movimiento de un brazo se debe tener en cuenta el conjunto de articulaciones disponibles para este como se puede observar en la **Tabla 3**.

**Tabla 3.**

*Articulaciones disponibles de acuerdo a la extremidad del Robot NAO*

<b>Cadena</b>	<b>Head</b>	<b>LLeg</b>	<b>RLeg</b>	<b>LArm</b>	<b>RArm</b>
<b>Motores</b>	HeadYaw	LHipYawPitch	RHipYawPitch	LShoulderPitch	RShoulderPitch
	HeadPitch	LHipPitch	RHipPitch	LShoulderRoll	RShoulderRoll
		LHipRoll	RHipRoll	LElbowYaw	RElbowYaw
		LKneePitch	RKneePitch	LElbowRoll	RElbowRoll
		LAnklePitch	RAnklePitch	LWristYaw	RWristYaw
		LAnkleRoll	RAnkleRoll	LHand	RHand

- Caminar. Le permite al robot generar el control para trasladarse en diferentes direcciones, es posible modificar la configuración por defecto de los pies del robot con el fin de proporcionar mejor exactitud en los pasos, aunque este método usa la retroalimentación de los sensores de presión con el objetivo de disminuir oscilaciones en el torso y poder caminar sobre diversas superficies debido a la corrección del equilibrio en los planos frontal y lateral. Este módulo utiliza un modelo dinámico simple o más conocido como el modelo de péndulo invertido, utilizando programación cuadrática enfocándose en el apoyo sobre una pierna cuando exista a intención de generar movimiento.
- Control cartesiano. Dedicado para controlar los *effectors* del robot en un plano cartesiano usando el método de cinemática inversa, donde a los *effectors* se los puede definir como un punto de tres dimensiones predefinido en el robot, generalmente ubicado al final de una cadena; El control mediante la cinemática inversa se puede dar de dos maneras, la primera que utiliza solo las articulaciones de una cadena para lograr un objetivo y la segunda, también conocida como: control de cuerpo entero, utiliza todas las articulaciones del robot para lograr su meta.

A parte de los módulos de ALMotion mencionados, este método ejecuta conjuntamente con los métodos de movimientos, llamados reflejos de seguridad, estas son funciones de auto protección del robot los cuales se mencionan a continuación:

- Evasión de auto-colisiones. Con este reflejo se puede evitar golpes propios que puede suscitarse por parte del mismo robot, puesto que le permite conocer autónomamente al robot la ubicación propia de cada articulación, de su cabeza y su torso, independientemente del movimiento que desee realizar, para llevarlo a cabo se crean esferas y cilindros acorde a la morfología del robot con el fin de realizar la detección de colisiones entre esferas, esferas y cilindros o entre cilindros de una forma más rápida y sencilla.

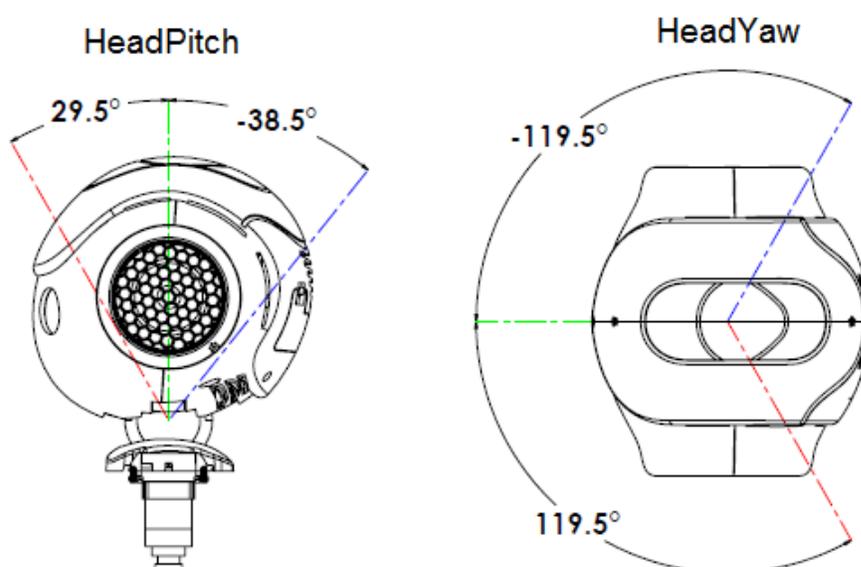
- Evasión de colisiones externas. A partir de este reflejo se activan los sensores para poder detectar cuando existe un obstáculo que comprometa su trayectoria y así evitar chocar con algo o alguien.
- Gestor de caída. Es uno de los reflejos del robot más importantes, debido a que protege la integridad del mismo de la manera más óptima posible, ante alguna eventual caída el robot se protegerá usando los brazos para amortiguar el golpe, esto se detecta mediante los sensores de presión ubicados en sus pies, si los dos sensores dan lecturas que indiquen falta de presión ante un movimiento no planificado se activará inmediatamente el sistema de protección de caídas y posteriormente procede a levantarse y continuar con su actividad.
- Rigidez inteligente. Esta función permite la adaptación de la potencia de los motores de acuerdo a la posición del robot con el fin de optimizar recursos y evitar sobrecalentamientos, atascamientos o fallos en movimientos cuando se está en superficies que no permitan un deslizamiento suave en el robot.
- Diagnostico. Es una función que me permite gestionar y analizar lo que sucede dentro del robot, gestiona otros reflejos con el fin de proteger la integridad del mismo, bloqueando algunos controles o desactivándolos, por ejemplo, si el control de posicionamiento se encuentra en falla, este reflejo desactiva la rigidez inteligente para evitar falsas lecturas y acciones no deseadas. Además ante alguna falla crítica, pone al robot a descansar y evita su activación hasta que el problema o la falla este solucionada.

#### **4.1.2 Descripción de las Articulaciones.**

Para el control de las articulaciones del robot NAO, se debe indicar mediante un conjunto de métodos la posición en la que debe encontrarse cada servomotor, definiendo el ángulo que estará representado en radianes, cada articulación consta de uno a tres servomotores con sus ángulos

máximos y mínimos de movimientos para no atentar contra la morfología humanoide del robot, en el caso de exceder el valor de estos ángulos, el servomotor tomara el limite más cercano.

Para el movimiento de la cabeza se observa en la figura los ángulos límites a los cuales se puede llegar alcanzar tanto en las coordenadas “Y” o llamada “HeadPitch” que corresponden al movimiento frontal y hacia atrás de la cabeza y a la coordenada “Z” o descrita como “HeadYaw” que relaciona el giro de la misma.



**Figura 25.** Ángulos límites para el movimiento de la cabeza.

Fuente: (Aldebaran Robotics, 2018)

Los ángulos límites que se tiene para el movimiento de la cabeza se especifica en la Tabla 4 , donde se especifica tanto el movimiento que realiza, ángulos límites en grados y en radianes.

**Tabla 4.**

*Ángulos límites para el movimiento de la cabeza.*

Ángulos	Límite ángulos (grados)	Límite ángulos (radianes)
<b>Articulación</b>		
<b>HeadPitch</b>	-38,5 a 29,5	-0.6720 a -0.5149

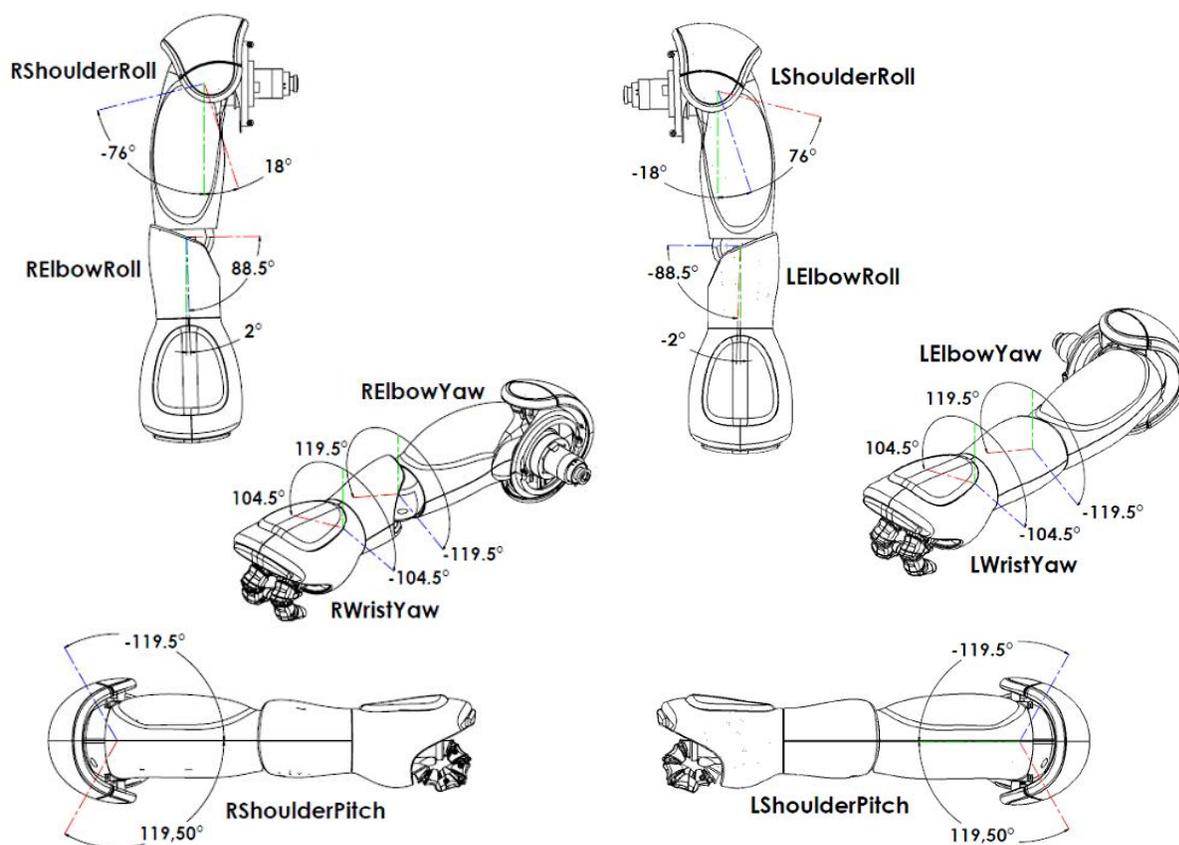
CONTINÚA 

HeadYaw

-119,5 a 119,5

-2,0857 a 2,0857

Para el brazo derecho e izquierdo se tienen nombres que son similares a diferencia de la primera letra que distingue si pertenece al lado derecho (R) o lado izquierdo (L), como se puede observar en la Figura 26, además se muestran los movimientos límites para cada articulación, lo que sirve para saber cómo se debe realizar el movimiento al momento de la programación, teniendo en cuenta cada uno de los cinco grados de libertad que tiene cada brazo que se da mediante la unión del movimiento de todas las articulaciones del mismo.



**Figura 26.** Ángulos límites de las articulaciones para los brazos del robot

Fuente: (Aldebaran Robotics, 2018)

Como se observa en la Figura 26, el límite los ángulos en algunos casos es similar, pero en otros es de signo contrario, debido a la posición de la articulación; Como se muestra en la *Tabla 5* y

*Tabla 6*, se especifican los límites de cada articulación tanto en grados como en radianes para cada brazo.

**Tabla 5.**

*Ángulos límites para las articulaciones del brazo derecho.*

Articulación	Ángulos	Límite ángulos (grados)	Límite ángulos (radianes)
<b>RShoulDrPitch</b>		-119,5 a 119,5	-2,0857 a 2,0857
<b>RShoulderRoll</b>		-76 a 18	-1,3265 a 0,3142
<b>RElbowYaw</b>		-119,5 a 119,5	-2,0857 a 2,0857
<b>RElbowRoll</b>		2 a 88,5	0,0349 a 1,5446
<b>RWristYaw</b>		-104,5 a 104,5	-1,8238 a 1,8238

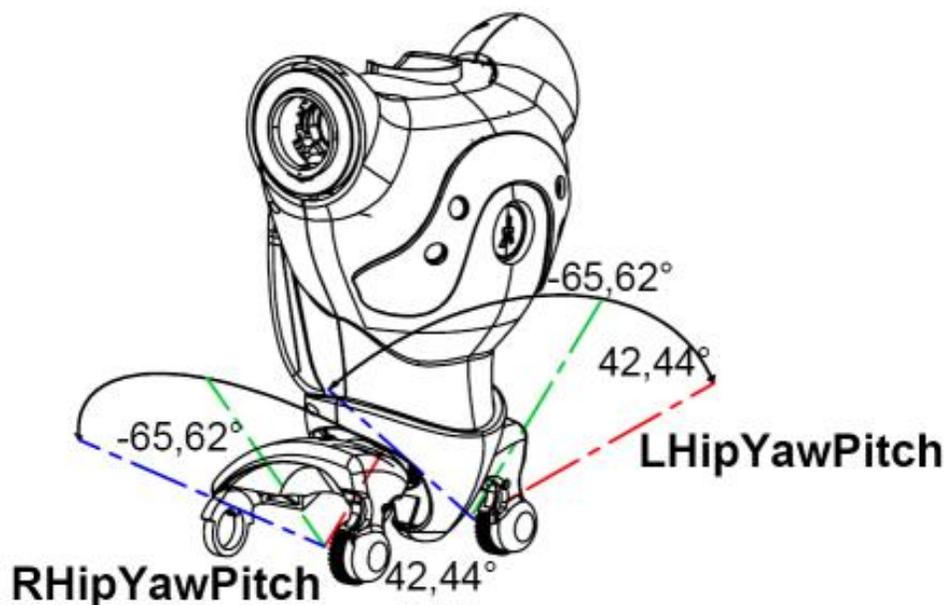
**Tabla 6.**

*Ángulos límites para las articulaciones del brazo izquierdo.*

Articulación	Ángulos	Límite ángulos (grados)	Límite ángulos (radianes)
<b>LSHoulDrPitch</b>		-119,5 a 119,5	-2,0857 a 2,0857
<b>LShoulderRoll</b>		-18 a 76	-0,3142 a 1,3265
<b>LElbowYaw</b>		-119,5 a 119,5	-2,0857 a 2,0857
<b>LElbowRoll</b>		88,5 a -2	-1,5446 a -0,0349
<b>LWristYaw</b>		-104,5 a 104,5	-1,8238 a 1,8238

Para controlar la inclinación del robot o giros del torso se cuenta con un servomotor que permite realizar dichos movimientos, como se muestra en la , LHipYawPitch y RHipYawPitch son

físicamente un solo motor para que no puedan ser controlados de forma independiente. En caso de órdenes contradictorias, LHipYawPitch siempre tiene la prioridad.



**Figura 27.** Ángulos límites del torso.

Fuente: (Aldebaran Robotics, 2018)

Los ángulos límites que se generan a partir de los movimientos del torso se especifican en la

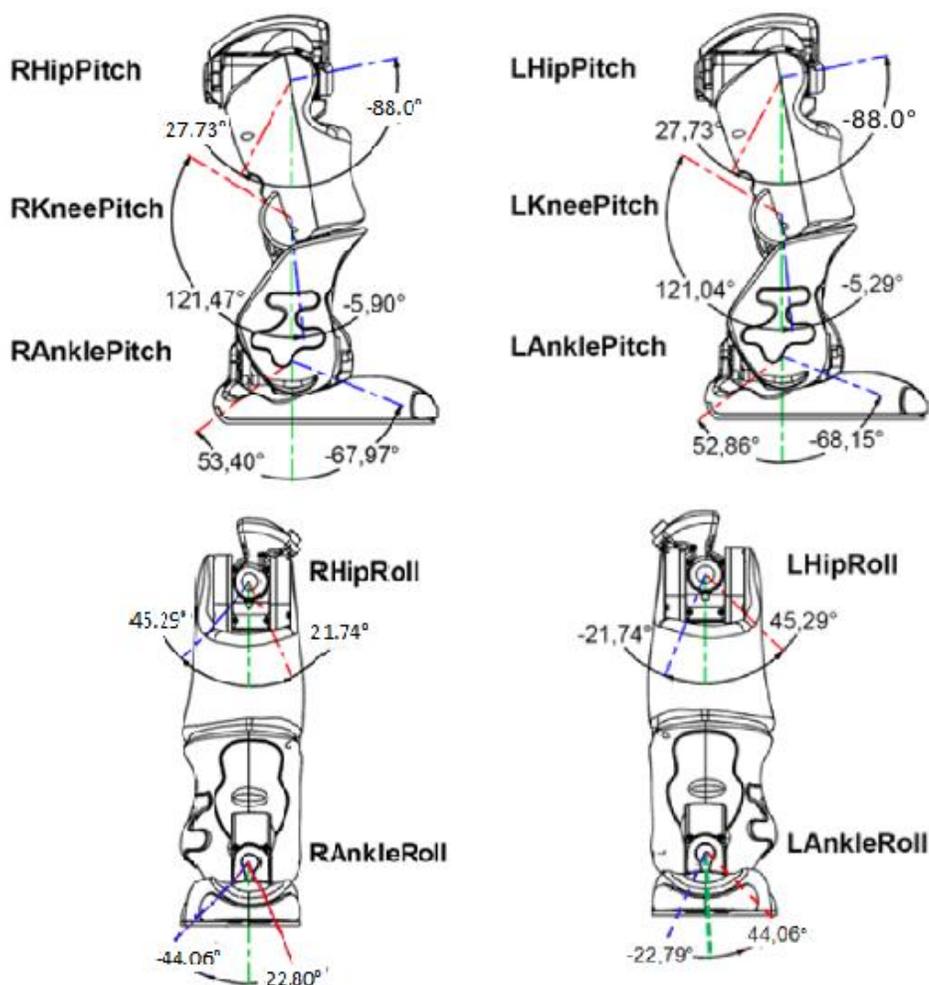
Tabla 7.

**Tabla 7.**

*Ángulos límites del torso.*

Ángulos	Límite ángulos (grados)	Límites ángulos (radianes)
<b>Articulación</b>		
<b>LHipYawPitch</b>	-65,62 a 42,44	-1,145303 a 0,740810
<b>RHipYawPitch</b>	-65,62 a 42,44	-1,145303 a 0,740810

Al igual que con lo ocurrido con las extremidades superiores, en las piernas se tienen nombres similares diferenciados por la primera letra para izquierda y derecha, los ángulos límites, su nombre y posición para cada articulación de las piernas se puede observar en la Figura 28.



**Figura 28.** Ángulos límites de las articulaciones de las piernas.

Fuente: (Aldebaran Robotics, 2018)

Para representar los límites angulares de movimiento en las articulaciones de las piernas se muestra la Tabla 8 y Tabla 9, donde se especifica el rango de movimiento en grados y en radianes.

**Tabla 8.**

*Ángulos límites para las articulaciones de la pierna derecha.*

Articulación	Ángulos	Límite ángulos (grados)	Límites ángulos (radianes)
<b>RHipRoll</b>		-45,29 a 21,74	-0,790477 a 0,379472
<b>RHipPitch</b>		-88,00 a 27,73	-1,535889 a 0,484090
<b>RKneePitch</b>		-5,90 a 121,47	-0,103083 a 2,120198
<b>RAnklePitch</b>		-67,97 a 53,40	-1,186448 a 0,932056
<b>RAnkleRoll</b>		-44,06 a 22,80	-0,768992 a 0,397935

**Tabla 9.**

*Ángulos límites para las articulaciones de la pierna izquierda.*

Articulación	Ángulos	Límite ángulos (grados)	Límites ángulos (radianes)
<b>LHipRoll</b>		-21,74 a 45,29	- 0,379472 a 0,790477
<b>LHipPitch</b>		-88,00 a 27,73	-1,535889 a 0,484090
<b>LKneePitch</b>		-5,29 a 121,04	-0,092346 a 2,112528
<b>LAnklePitch</b>		-68,15 a 52,86	-1,189516 a 0,922747
<b>LAnkleRoll</b>		-22,79 a 44,06	0,397880 a 0,769001

#### 4.1.2 Control de las Articulaciones en C++

Para empezar con la programación de los movimientos en lenguaje de C++ es necesario declarar el *proxy* del módulo del movimiento en este caso de *ALMotion*, asignando un nombre que se usará para enlazar a las demás funciones que contiene dicho módulo; Además, se debe especificar la dirección IP del robot NAO que se encuentre conectado a la red y el puerto de comunicación, que como ya se mencionó anteriormente por defecto es “9559”.

Una vez declarado el proxy del módulo que se va a utilizar, es recomendable establecer la rigidez de los motores en su máximo valor, es decir en “1”, con el fin de lograr alcanzar el ángulo deseado, esto se logra mediante el método `ALMotion::setStiffnesses` y se lo puede declarar para una articulación específica o para todo el cuerpo.

El siguiente paso corresponde al uso del método `ALMotion::angleInterpolation`, la cual se usará debido a que específicamente este método permite controlar el tiempo en el que se generan los movimientos y consta de cuatro etapas, la primera consiste en declarar la articulación o articulaciones que se desean mover, segundo especificar el ángulo o la lista de ángulos con su valor en radianes, tercero indicar el tiempo que tardará en generarse el movimiento y por último declarar si el movimiento es descrito en ángulos absolutos o relativos, es decir si se declara como ángulos absolutos el ángulo subsiguiente se generará a partir de la posición inicial en la que se encontraba la articulación.

```
1 motion.setStiffnesses("Body", 1.0f);
2
3 //MOVIMIENTO DE CABEZA
4 //Articulación a mover
5 AL::ALValue pose1 = "HeadYaw";
6 //Angulo de movimiento en radianes
7 AL::ALValue angleLists1 = 1.0f;
8 //Tiempo para generar el movimiento
9 AL::ALValue timeLists1 = 1.0f;
10 //Ángulos absolutos
11 bool isAbsolute = true;
12 //Ejecución del movimiento uno
13 motion.angleInterpolation(pose1, angleLists1, timeLists1, isAbsolute);
14
15 qi::os::sleep(1.0f);
16
17 //BRAZOS ABIERTOS
18 //Arreglo de articulaciones a mover
19 ALValue pose2 = ALValue::array("LShoulderPitch", "LShoulderRoll",
20 "RShoulderPitch", "RShoulderRoll", "LElbowYaw", "RElbowYaw");
```

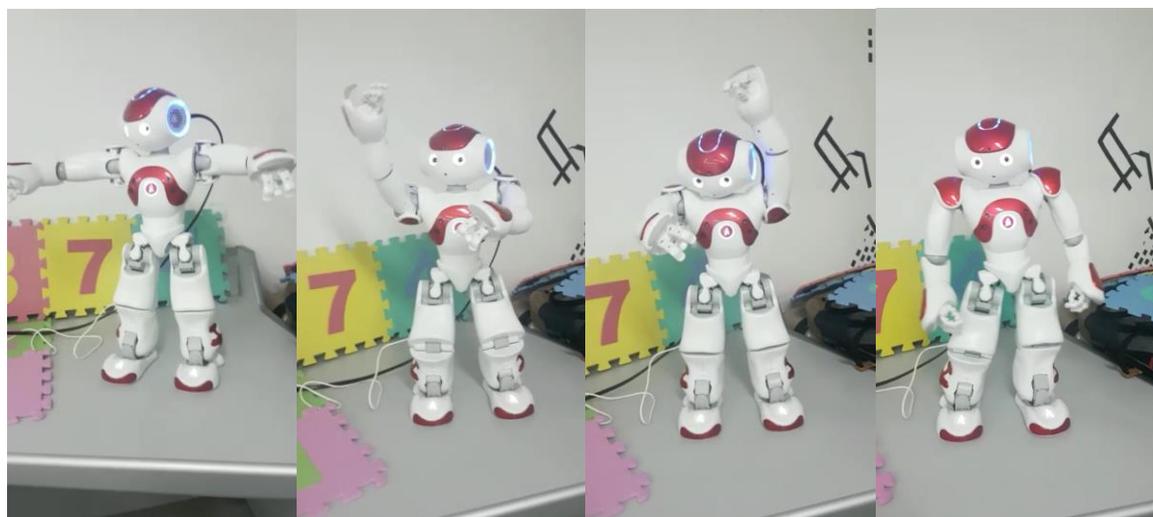
```

21 //Ángulos de cada articulación
22 angleLists2.arraySetSize(6);
23 angleLists2[0] = AL::ALValue::array(0.0f);
24 angleLists2[1] = AL::ALValue::array(1.0f);
25 angleLists2[2] = AL::ALValue::array(0.0f);
26 angleLists2[3] = AL::ALValue::array(-1.0f);
27 angleLists2[4] = AL::ALValue::array(0.0f);
28 angleLists2[5] = AL::ALValue::array(0.0f);
29 //Tiempos de movimiento
30 timeLists2.arraySetSize(6);
31 timeLists2[0] = AL::ALValue::array(1.0f);
32 timeLists2[1] = AL::ALValue::array(1.0f);
33 timeLists2[2] = AL::ALValue::array(1.0f);
34 timeLists2[3] = AL::ALValue::array(1.0f);
35 timeLists2[4] = AL::ALValue::array(1.0f);
36 timeLists2[5] = AL::ALValue::array(1.0f);
37 //Ejecución del movimiento dos
38 motion.angleInterpolation(pose2, angleLists2, timeLists2, isAbsolute);
39
40 qi::os::sleep(1.0f);
41
42 //MOVIMIENTO MULTIPLE
43 //Arreglo de articulaciones a mover
44 ALValue pose3 = ALValue::array("HeadPitch", "RShoulderPitch",
45 "LShoulderPitch", "RElbowYaw", "RKneePitch", "RHipPitch", "LKneePitch",
46 "LHipPitch", "RAnklePitch", "LAnklePitch");
47 //Ángulos de cada articulación
48 angleLists3.arraySetSize(10);
49 angleLists3[0] = AL::ALValue::array(-0.17f, 0.4f, -0.17f, 0.4f);
50 angleLists3[1] = AL::ALValue::array(-0.8f, -1.22f, -0.8f, -1.22f);
51 angleLists3[2] = AL::ALValue::array(0.17f, 0.26f, 0.17f, 0.26f);
52 angleLists3[3] = AL::ALValue::array(1.40f, 0.35, 1.40f, 0.35);
53 angleLists3[4] = AL::ALValue::array(0.7f, -0.087f, 0.7f, -0.087f);
54 angleLists3[5] = AL::ALValue::array(-0.35f, 0.13f, -0.35f, 0.13f);
55 angleLists3[6] = AL::ALValue::array(0.7f, -0.087f, 0.7f, -0.087f);
56 angleLists3[7] = AL::ALValue::array(-0.35f, 0.13f, -0.35f, 0.13f);
57 angleLists3[8] = AL::ALValue::array(-0.26f, 0.09f, -0.26f, 0.09f);
58 angleLists3[9] = AL::ALValue::array(-0.26f, 0.09f, -0.26f, 0.09f);
59 //Tiempos de movimiento
60 timeLists3.arraySetSize(10);
61 timeLists3[0] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
62 timeLists3[1] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
63 timeLists3[2] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
64 timeLists3[3] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
65 timeLists3[4] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
66 timeLists3[5] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
67 timeLists3[6] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
68 timeLists3[7] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
69 timeLists3[8] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
70 timeLists3[9] = AL::ALValue::array(1.0f, 1.3f, 1.6f, 1.9f);
71 //Ejecución del movimiento tres
72 motion.angleInterpolation(pose3, angleLists3, timeLists3, isAbsolute);

```

Como se observa en el ejemplo del código que se usa para generar el movimiento de las articulaciones, se muestra el movimiento de la cabeza, de la manera más simple para una sola articulación, para movimientos más complejos se realiza un arreglo de articulaciones, donde se define para cada una de estas el ángulo al que debe llegar y el tiempo requerido, pero si dicha articulación requiere ser movida múltiples veces se realiza una secuencia de ángulos y los tiempos requeridos para efectuar el movimiento, considerando que el arreglo de las mismas no puede exceder los diez elementos o se generarán errores de compilación.

En la Figura 29 se observa cómo se lleva a cabo la secuencia de movimientos que se ejecutan para el robot NAO.

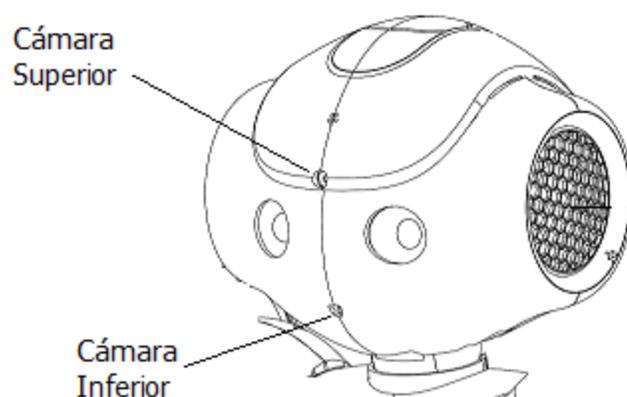


*Figura 29.* Secuencia de movimiento realizado con el robot NAO.

## 4.2 Reconocimiento facial

El robot NAO cuenta con dos cámaras idénticas ubicadas en su cabeza como se muestra en la Figura 30, una en la parte frontal sobre sus ojos, la cual muestra todo lo que pasa en la parte frontal superior del robot, es comúnmente usado para identificar objetos a mayores distancias o rostros, la segunda cámara se encuentra en la boca y permite observar la parte inferior y es usada para

identificar objetos que se encuentren en el suelo u obstáculos que impidan su movilidad o causar caídas. Las cámaras del robot NAO son de alta definición con una resolución de 1288x968 píxeles que trabajan a 30 cuadros por segundo, presentando el formato de datos en un espacio de color YUV422, enviando 4 bytes por cada 2 píxeles, aunque todas estas características se pueden cambiar y configurar de acuerdo a la Tabla 10, trabajan con un ángulo de visión de 47,64 grados verticalmente y 60,97 grados de forma horizontal.



**Figura 30.** Posicionamiento de las cámaras en el robot NAO

**Tabla 10.**

*Características de configuración para las cámaras del robot NAO.*

Resolución	Características	Espacio de color	Cuadros por segundo
4VGA (1288x968)		YUV422 (Por defecto)	Máximo 30 fps
VGA (640x480)		YUV (24 bits)	
QVGA (320x240)		Y (8 bits)	
QQVGA (160x120)		RGB (24 bits)	
		BGR (24 bits)	
		HYS (24 bits)	

#### **4.2.1 Módulo ALFaceDetection.**

Es un módulo encargado de la detección y reconocimientos de rostros que se encuentran frente al robot, proporcionando una matriz que especifica la posición en la que se encuentra el rostro y todas las coordenadas angulares de las características de la cara, es decir, la posición de la nariz, ojos y boca, con el fin de reconocer a la persona que se encuentra en su campo de visión siempre y cuando su rostro haya sido previamente guardado en la memoria del robot; Proceso que se describirá posteriormente.

La función de reconocimiento facial tiene características configurables que describe cómo funciona la información que se maneja, dentro de estas características se encuentran:

- **Umbral.** ALFaceDetection se lo puede configurar para ser más o menos estricto en su nivel de certeza al momento de detectar un rostro conocido, es posible establecer un valor de puntuación que vaya de 0 a 1, mientras más alta sea la puntuación configurada, se tendrá una mayor certeza al momento del reconocimiento, reduciendo errores, pero sería más difícil obtener un resultado, puesto que solo se verían aquellos cuya puntuación este por encima del umbral considerado.
- **Filtro Temporal.** Una vez realizado el reconocimiento de una persona se activa un filtro temporal que evita reconocer una y otra vez el mismo rostro, con el fin de no activar otros módulos repetidamente, por ejemplo, al detectar un rostro conocido y se haya programado un saludo, el robot evitara saludar cada segundo, debido que se coloca en una memoria de corto plazo el nombre de la persona detectada, en cuanto pase más de cuatro segundos sin detectar una cara, dicha memoria se reinicia y puede volver a detectar cualquier persona.

Una vez que se ejecute el módulo ALFaceDetection, con la suscripción a un evento llamado FaceDetected(), el cual es una llamada de retorno ante el cumplimiento de una función, que por lo general, obtiene datos de la memoria, una vez detectado el rostro, dicho evento retorna valores en forma matriz que se los describirá individualmente para su explicación como se observa a continuación:

```
FaceDetected =  
[  
  Marca_de_tiempo, [Información_del_rostro [N], Filtro_de_Información],  
  Posición_de_la_cámara_en_relación_al_torso,  
  Posición_de_la_cámara_en_relación_al_robot,  
  Camara_Id  
]
```

Marca de tiempo. Es el tiempo que se tomó para realizar la detección del rostro y está formada por dos componentes dados en segundos y en microsegundos como se muestra:

```
Marca de Tiempo =  
[  
  Marca_de_tiempo_Segundos,  
  Marca_de_tiempo_Microsegundos  
]
```

Información del rostro. Una vez realizada la detección facial, la información de esta se presenta en dos parámetros.

```
Información_del_rostro =  
[  
  Información_forma,  
  Información_extra  
]
```

Donde la información de la forma muestra datos que se los representa como Alfa, Beta, X y Y, donde los dos primero representan la ubicación de la cara en términos del ángulo de la cámara; X y Y constituyen el tamaño de la cara en función del ángulo de la cámara.

```
Información_forma =  
[  
  0,  
  Alfa,  
  Beta,  
  X,  
  Y  
]
```

La información extra muestra el contenido de todos los puntos de interés del rostro en siete parámetros, el primero en mostrar es Rostro\_Id, el cual es un número momentáneo de identificación que se asigna para la cara de la persona detectada, como segundo parámetro se tiene un puntaje que entre más alto sea, indicará que mejor fue el reconocimiento, el tercer elemento se muestra como una etiqueta del rostro, la cual puede almacenar el nombre o identificación de la persona detectada si previamente se encontraba en la memoria del robot, a partir de aquí se muestran la posición de los puntos de interés del rostro, como son los de cada ojo, nariz y boca.

```
Información_extra =  
[  
  Rostro_Id,  
  Puntuación,  
  Etiqueta_rostro,  
  Puntos_ojo_izq,  
  Puntos_ojo_der,  
  -,  
  -,  
  Puntos_nariz,  
  Puntos_boca  
]
```

La posición de la cámara en relación al torso y la posición de la cámara en relación al robot muestran datos de la posición de la cámara al momento de realizar el reconocimiento, con respecto a la referencia espacial que se tiene del torso del robot y la segunda a partir del eje creado por los pies del robot alrededor de un eje z vertical.

El último parámetro en mostrar se lo representó como Cámara\_Id y es aquel que muestra cuál de las dos cámaras del robot NAO se utilizó, “0” para la cámara superior y “1” para la cámara inferior.

El evento FaceDetected como se detalló, proporciona todos los datos del rostro de una persona, dichos datos deben ser adquiridos de la memoria del robot mediante el módulo ALMemory::getData(), obteniendo un arreglo de tipo ALValue, siempre y cuando se haya detectado un rostro como se muestra en la Figura 31, caso contrario proporcionará un arreglo vacío.



almacena en la base de datos con el nombre especificado, caso contrario si la etiqueta del rostro contiene un nombre o identificación se procede a emitir un saludo.

```

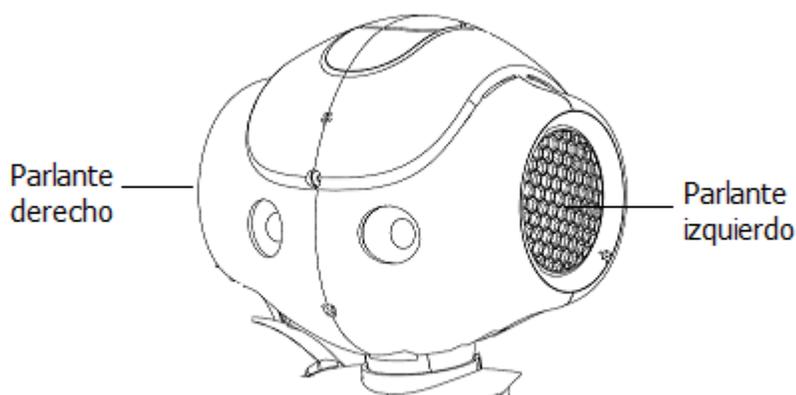
1  ALFaceDetectionProxy facedetection(naoip, pport);
2  facedetection.subscribe("FaceDetected");
3  //Declaración de variables
4  ALValue fFaces;
5  ALValue FFaceInfoArray;
6  try{
7      //Inicio búsqueda de rostros
8      fFaces = memproxy.getData("FaceDetected");
9      //Mientras no encuentre caras sigue buscando
10     while (fFaces.getSize()<2){
11         fFaces = memproxy.getData("FaceDetected");
12     }
13     while (fFaces.getSize()>=2)
14     {
15         FFaceInfoArray = fFaces[1];
16         //Lectura de informacion de la cara
17         for (int j = 0; j < FFaceInfoArray.getSize()-1; j++)
18         {
19             ALValue faceInfo = FFaceInfoArray[j];
20             ALValue faceExtraInfo = faceInfo[1];
21             string faceName = faceExtraInfo[2];
22             //Rostro desconocido
23             if (faceName == "")
24             {
25                 tts.say("Aprendiendo rostro nuevo");
26                 bool face = facedetection.learnFace("Fabricio");
27                 qi::os::sleep(2.0f);
28                 NaoDialogo2();
29             }
30             //
31             else
32             {
33                 qiLogInfo("module.name")<< "Cara detectada"<< endl;
34                 string fraseNombre = "Hola conocido";
35                 NaoDialogo2();
36             }
37         }
38     }
39 }
40 catch (const AL::ALError& e) {
41     qiLogError("module.name")<< e.what() << endl;
42 }

```

### 4.3 Gestor de Diálogo

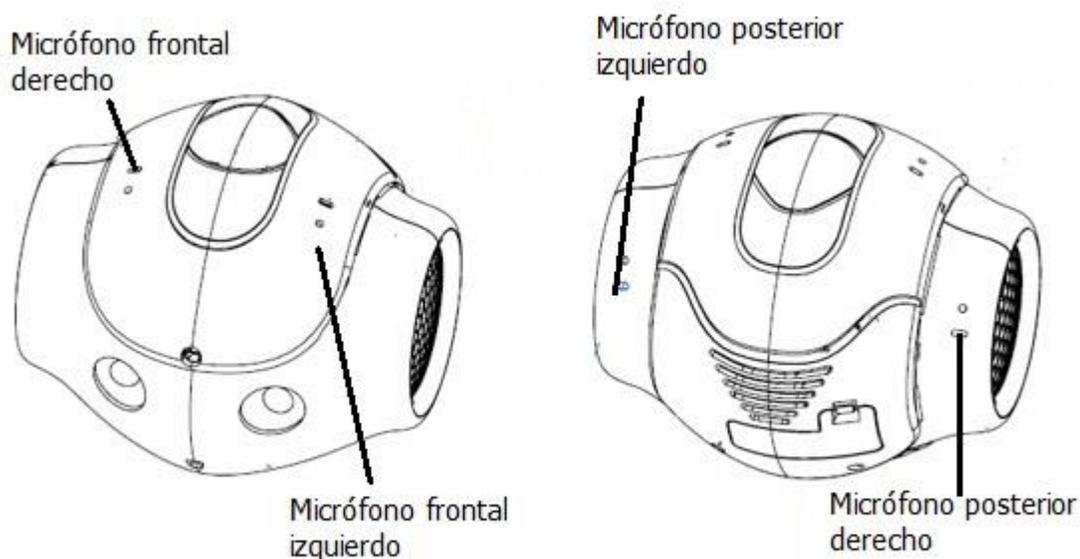
Para los sistemas de diálogo hablado intervienen parámetros que interactúan para dar respuestas a preguntas realizadas por los usuarios. Es considerado como un sistema modular, donde intervienen tantas funciones para el reconocimiento del habla, comprensión en la pronunciación, y el gestor de diálogo. Estos módulos no están exentos de errores, debido a que la fase de comprensión debe tratar con los errores que se producen en la frase de reconocimiento y el gestor de diálogo con los generados en la fase de comprensión; Además, se tienen dificultades al momento de reconocer la pronunciación, debido a defectos propios del habla o a errores producidos por el ruido ambiental.

Los gestores de diálogo se han desarrollado reduciendo al máximo los errores que se producen para sistemas de diálogo haciéndolos más fiables para las etapas de reconocimiento y comprensión. El robot humanoide NAO cuenta con herramientas para que sea posible implementar estos sistemas, debido a que posee un sistema de emisión estéreo compuesto por dos altavoces que se ubican en sus oídos como se muestra en la Figura 32.



**Figura 32.** Posicionamiento de los parlantes en el robot NAO

Además, cuenta con cuatro micrófonos que trabajan en el rango de 150 Hz a 12 Khz y se encuentran ubicados en la cabeza como se muestra en la Figura 33.



**Figura 33.** Posicionamiento de los micrófonos en el robot NAO

#### 4.3.1 Módulo ALDialog.

Es un módulo que le permite al robot NAO mantener conversaciones simples mediante el uso de reglas que deben ser escritas previamente, además de ser categorizadas de acuerdo al ámbito de la conversación.

Los gestores de diálogo dependen de reglas escritas que gestionan la forma en que se debe mantener una conversación, de esta manera ALDialog maneja dos tipos de reglas, las reglas del usuario y las reglas propuestas.

- Reglas de Usuario. Permite relacionar específicamente una entrada de usuario a una salida del robot, es decir, identificar lo que un usuario dice y ofrecer una respuesta por parte del robot.

- Reglas de propósito. Permiten tener una salida específica del robot sin ninguna intervención del usuario, se produce automáticamente.

```
#Regla de usuario
u:(Hola) Hola, es un gusto saludarte
u:(Es hora de dormir) si necesitamos descansar

#Regla de propósito
proposal: Pasaste bien el día de hoy?
  u1:(si) Eso es genial, me alegro mucho
  u1:(no) Bueno habrá días mejores
```

Para estructurar un sistema de diálogo se pueden usar varias herramientas que permiten mejorar la comprensión que tendrá el robot y manejar de una manera más simple los datos.

Como primer punto para la creación de un sistema de diálogo, es necesario asignar un tema específico y el lenguaje en el cual se mantendrá la charla, con el fin de cargar las reglas en el motor del reconocimiento de voz que posee el robot.

Los temas deben contener únicamente una palabra que la identificará y no contener caracteres especiales debido a que pueden existir errores de interpretación. Existe gran variedad de lenguajes que puede soportar la herramienta de reconocimiento de voz, para esta investigación se planteó el uso del idioma español representado por su acrónimo “*spe*”, quedando la estructura del inicio del gestor del dialogo como se platea a continuación:

```
topic: ~entretenimiento ()
language: spe

u:(Hola) Hola, es un gusto saludarte
```

Cuando se quieren definir palabras que se encuentren dentro de un mismo contexto se los puede declarar bajo conceptos que simplifica la estructura en los sistemas de diálogo y se los puede usar repetidamente, sin tener que repetir frases que cambien solo por palabras similares.

```

topic: ~entretenimiento ()
language: spe

concept:(saludo) [hola "buenos días" "que tal"]
concept:(actividad) [bailar cantar]

u:(~saludo) ~saludo
u:(te gusta ~actividad) si me encanta ~actividad

```

Para hacer de un dialogo una secuencia estructurada se pueden establecer subreglas que dependen de un regla anterior, generando así conexiones que dan sentido a la conversación.

```

topic: ~entretenimiento ()
language: spe

u:(hola) Hola como estas?
  u1:(bien) me alegra mucho escucharlo
  u1:(mal) Tranquilo, te gustaría cantar conmigo?
    u2:(si) cantemos algo divertido
      u3:(si) $cancion
      u3:(no) hoy solo pondré algo divertido
    u2:(no) entonces probemos bailando
      u3:(si) mírame bailar $baile
      u3:(no) levanta esos ánimos

```

Como se observa, si se inicia un diálogo, en la cual dependen de respuestas previas se puede orientar el mecanismo de interacción, es decir, si en u1 se tiene alguna de las dos respuestas, el robot contesta con lo establecido, desactivando u1 y activando u2 para continuar con la estructura diseñada para la conversación. En el caso de tener una entrada de usuario que coincida con otra regla de usuario, inmediatamente se desactivan las subreglas anteriores y se activan las nuevas.

Existen funciones que permiten hacer más dinámico un sistema de diálogo, para que se asemeje lo mayor posible a una interacción natural y no planificada, como generar respuestas similares de una manera aleatoria, reconocer palabras de una lista y brindar la respuesta adecuada, enfocarse en otro tema de conversación cuando se desee cambiar.

Se tiene también caracteres que ofrecen un significado específico dentro del sistema de diálogo como:

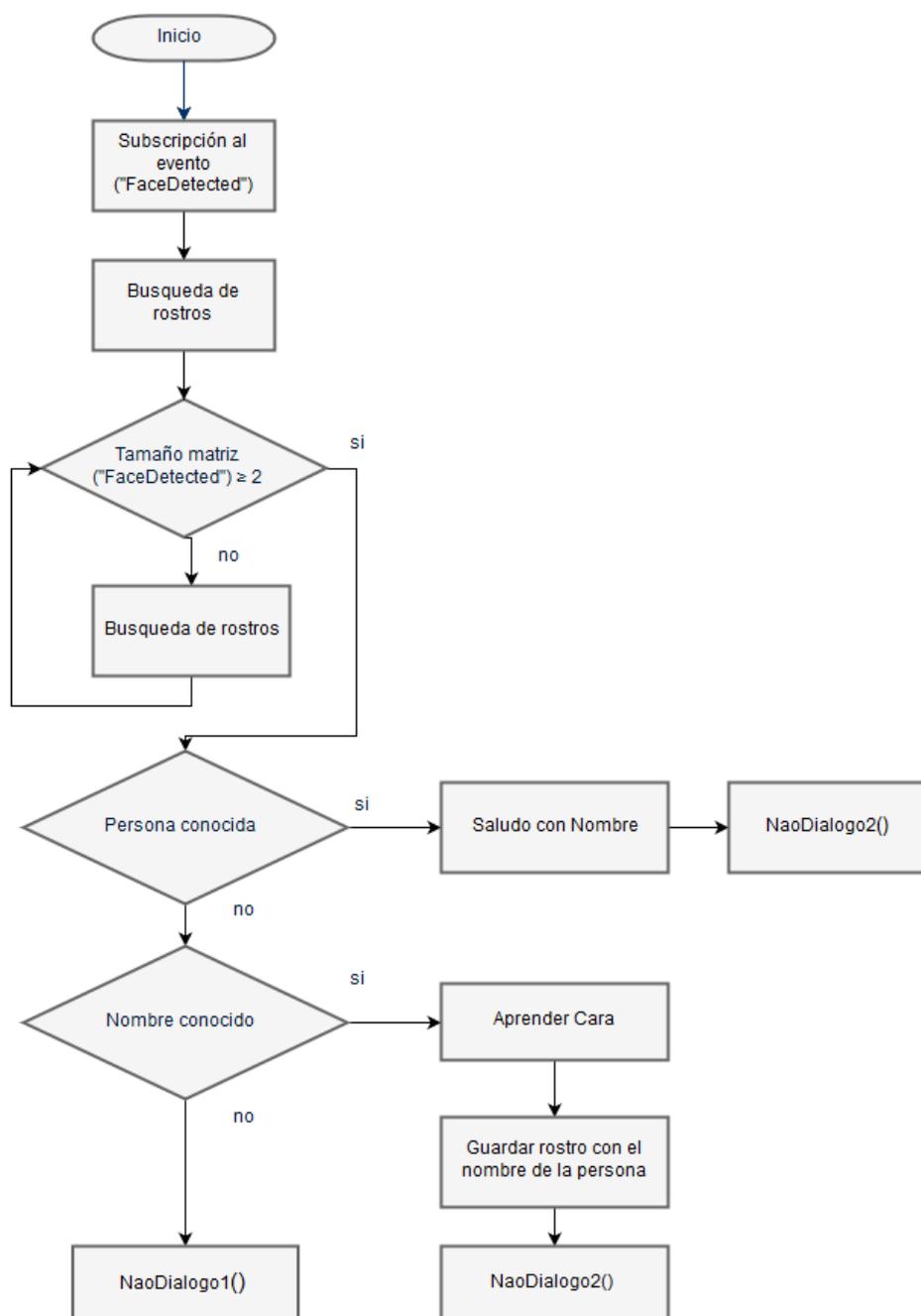
- Palabras prohibidas ( ! ). Este carácter permite excluir palabras que no se requieran cuando se de una intervención humana.
- Tag ( % ). Permite identificar propuestas o reglas dentro de la estructura del dialogo, activándolas o desactivándolas según sea necesario.
- Almacenar una entrada ( \_ ). Permite guardar de manera momentánea una palabra, con el fin de volverlo a utilizar ya sea para dar una respuesta por parte del robot o relacionarlo con algún tag.
- Variable ( \$ ). Este carácter sirve para asignar una variable de manera global, no necesita ser declarada para su uso y se almacenan en la memoria del robot, con el fin de interrelacionar el programa principal con el gestor de diálogos.
- Llamada de un concepto ( ~ ). Un vez declarado el concepto y las palabras que lo conforman, para llamarlo en cualquier momento únicamente se antepone el carácter indicado.
- Condiciones ( ==, >, < >, < ). Estas condiciones permiten realizar comparaciones a una entrada de usuario o salida del robot de manera que se puedan dar juicios de valor y permitir la selección de posibles respuestas.

#### 4.4 Interacción con el robot NAO

Para el desarrollo del presente proyecto se deben combinar las técnicas de persuasión que se mencionaron en el primer capítulo junto con todos los módulos descritos anteriormente con el fin de obtener un robot capaz de interactuar de la manera más natural posible con los usuarios, motivándolos o guiándolos para que el robot pueda realizar actividades enfocadas al entretenimiento.

Uno de estos métodos de persuasión es conocer a la persona con la cual se mantiene la interacción, interviniendo para esto el módulo de reconocimiento facial, donde el programa obtiene los datos de la memoria, que permite saber si es una persona conocida o es necesario aprender su cara para recordar en un futuro o en una próxima interacción, logrando con esto un nivel de conocimiento personal hacia con el usuario.

Como primer paso se ejecuta la función que contiene la programación para realizar todo el reconocimiento facial de una persona, llamada `ReconocimientoFacial()`, el diagrama de flujo que se muestra describe como son los pasos para llevarlo a cabo, se empieza suscribiendo al módulo “FaceDetected” que como ya se mencionó contiene la información del arreglo de las caras que se detecten en ese instante, a continuación se mantiene un ciclo infinito que busca rostros y mientras una de sus cámaras no encuentre una, seguirá buscando, una vez que detecte la presencia de un rostro se procede a obtener el arreglo de los rasgos característicos de este, con el fin de saber si la persona que se detectó se encuentra en la base de datos propia del módulo de reconocimiento facial. Si no se encuentra con una persona conocida se carga la función de diálogo número uno, lo contrario, si se encuentra una persona conocida, se cargan los datos del usuario y un saludo con su nombre, para proceder a lanzar la función de diálogo dos.

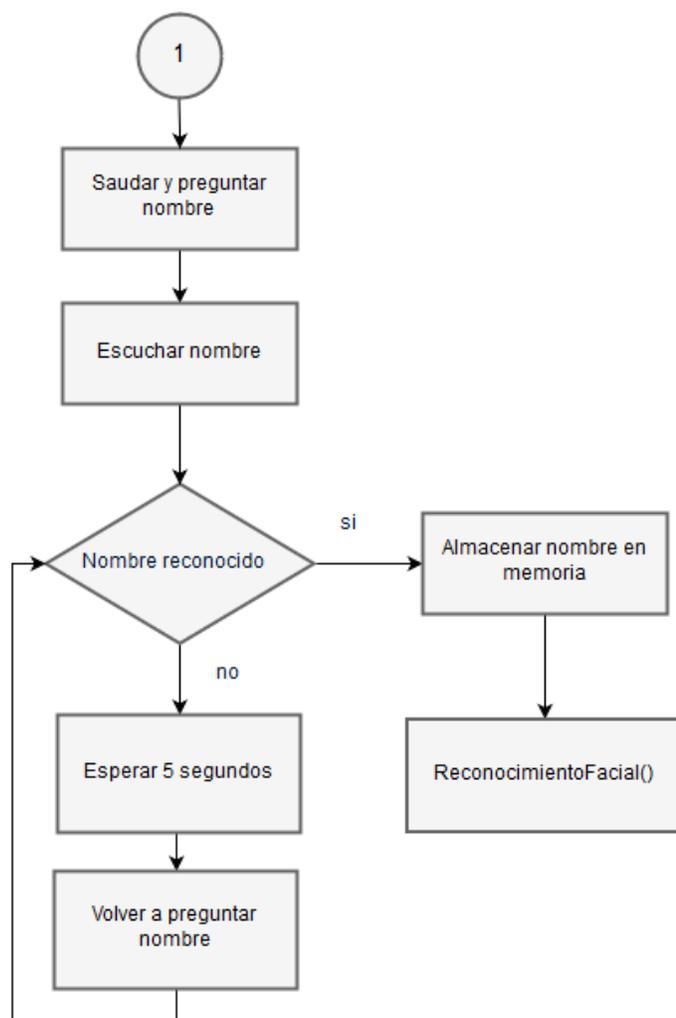


**Figura 34.** Diagrama de flujo de la función de reconocimiento facial.

Una vez realizado el proceso del reconocimiento facial, se obtiene de la memoria los datos de la persona detectada, si no fue reconocida con ningún nombre se procede a ingresar a la función de NaoDialogo1(), la cual contiene la base de datos de nombres de personas, espera a escuchar un saludo y el robot contestará de la misma manera, añadiendo la pregunta sobre cuál es el nombre de la persona que está en ese momento interactuando con el robot, almacena en una variable y vuelve a la función ReconocimientoFacial(), con el fin de aprender el rostro y almacenarlo con el nombre ingresado. En el diagrama de flujo que se muestra a continuación, se observa como es el procedimiento de la función NaoDailogo1().

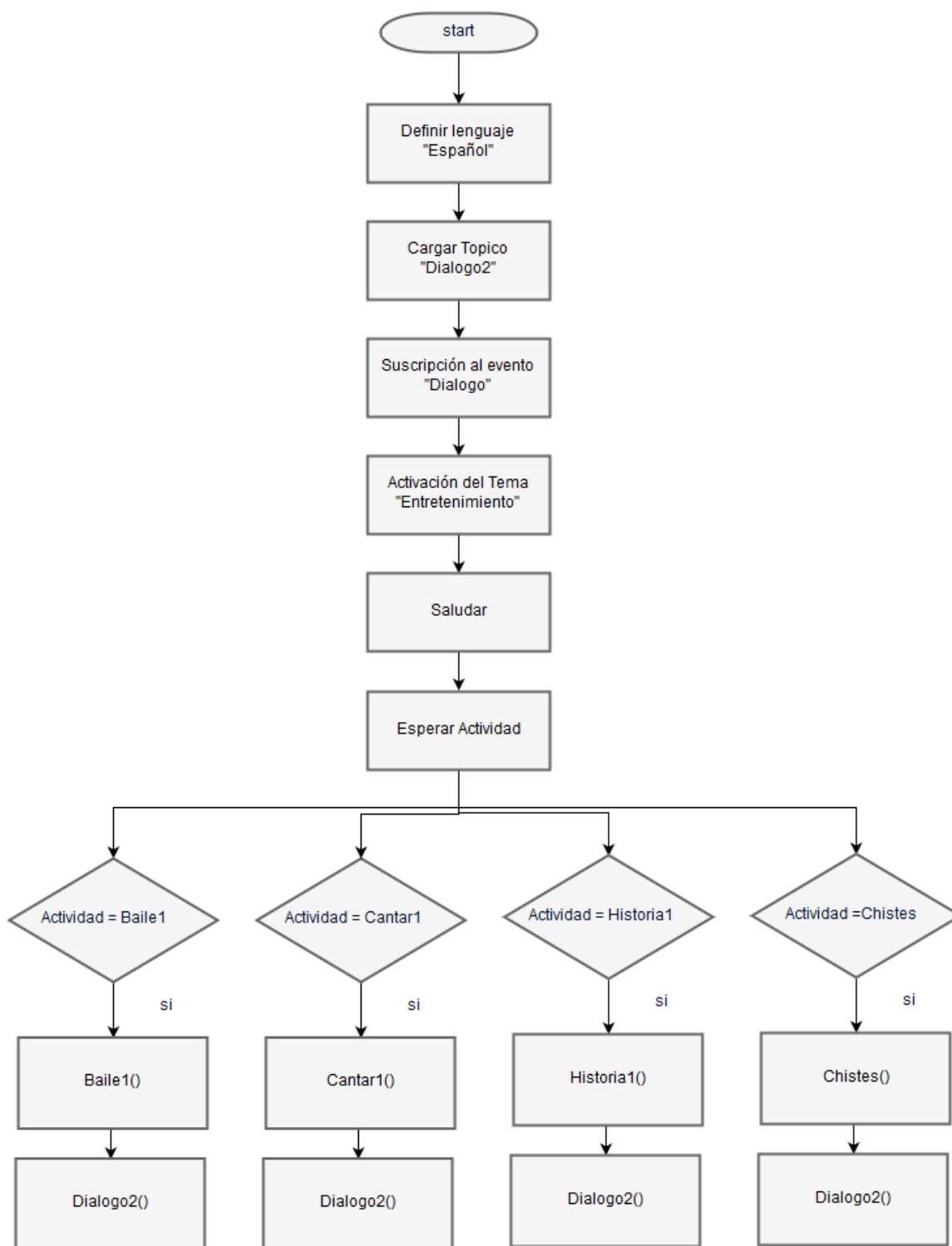


CONTINÚA 



**Figura 35.** Diagrama de flujo de la función de Diálogo 1.

En el caso de que la persona fue reconocida se procede a realizar un saludo con el nombre con el que fue almacenado su rostro en la memoria del robot y se ejecuta la función `NaoDialogo2()`, la cual contiene el diálogo de persuasión diseñado para que se cumplan con las actividades de entretenimiento que se las programó para el robot, estas incluyen bailes, canciones, historias y chistes. La función que contiene el segundo diálogo.



**Figura 36.** Diagrama de flujo de la función de Diálogo 2.

Mediante el gestor de diálogo se diseñó una conversación que lleva al usuario a cumplir con las actividades que propone el robot, usando el método de persuasión por habla y gestos. La conversación está diseñada con el fin de mantener al usuario centrado en la temática de entretenimiento, enfocándose en obtener respuestas ante las propuestas de actividades planteadas por el robot, dichas respuestas pueden ser afirmativas, negativas o acorde a la pregunta del robot.

Cuando las posibles respuestas que se den por parte del usuario sean afirmativas, el robot expresará emoción mediante gestos, estos pueden ser señas de victoria, risa, felicidad, etc. Caso contrario si se tienen respuestas negativas ante las propuestas de actividades del robot este mostrará descontento mediante gestos como llanto, enojo, malestar, etc. Con el fin de usar los movimientos como una forma de persuasión que influye en la toma de decisiones de los usuarios.

Si a pesar de los gestos usados aún se siguen teniendo respuestas negativas, el diálogo está diseñado de tal modo que al final se realiza una actividad propuesta por el robot, debido a que pedirá la opinión del usuario al realizarla y más no su consentimiento de hacerlo, manejando de esta forma técnicas persuasivas mediante diálogos y comunicación verbal.

### **Consideraciones.**

Los diagramas de flujo que se muestran en este capítulo, representan la forma en la cual se diseñaron las funciones tanto, de reconocimiento facial como para el gestor de diálogo, de la manera más sencilla posible con el fin de entender su funcionamiento; No se especifican los diagramas de flujo para las actividades programadas debido a la complejidad que representa mostrar los movimientos que se realizan.

El código mostrado representa un ejemplo de la forma en la que se usan los algoritmos de las funciones utilizadas para el entretenimiento persuasivo, así como los gestores de diálogo

implementados, para mayor información de cómo se llevó a cabo la programación de las funciones, actividades, movimientos y diálogos véase en Anexos.

## CAPÍTULO V

### RESULTADOS

#### 5.1 Evaluación

Las pruebas se desarrollan con el objetivo de conocer el funcionamiento real de las funciones básicas de interacción implementadas sobre el robot humanoide NAO y de esta manera determinar si es posible obtener un robot que sea capaz de interactuar y persuadir a los usuarios para realizar actividades enfocadas al entretenimiento. En primera instancia se realizan varias demostraciones grupales para obtener información sobre la aceptación de las actividades que realiza el robot previo a las interacciones individuales que se pretenden realizar como experimentación principal.



*Figura 37.* Primera demostración grupal del funcionamiento del robot



**Figura 38.** Segunda demostración grupal del funcionamiento del robot NAO

De acuerdo a la aceptación que se dio por parte de los usuarios grupales de las actividades que son posibles que realice el robot, se procedió a desarrollar evaluaciones individuales con el fin de analizar el impacto de las funciones de entretenimiento persuasivas.

Para evaluar el desempeño de la propuesta del desarrollo de un sistema básico de robótica de entretenimiento persuasivo la cual se desarrolla en este trabajo, se ha planteado las pruebas experimentales para diferentes grupos de edades en los usuarios.

Debido al nivel de interacción que se da en diferentes rangos de edades se ha tomado cinco grupos principales de edades de los usuarios con los que se llevó a cabo las pruebas y las cuales varían dentro de los siguientes límites:

- Usuarios de 7 a 12 años
- Usuarios de 13 a 19 años

- Usuarios de 20 a 29 años
- Usuarios de 30 a 59 años
- Usuarios de 59 a 80 años

Estos rangos fueron definidos debido a la capacidad de interacción que se tiene en cada grupo, catalogados en relación al análisis que se da para niños, adolescentes, jóvenes, adultos y adultos mayores.

Se toman veinte personas para el experimento, de las cuales cuatro pertenecen a cada grupo antes descrito en los rangos de edades establecidos.

Las pruebas de funcionamiento se ejecutan para cada usuario y consiste en realizar una o varias interacciones con el robot, explicando previamente como es su funcionamiento y la temática que trata.

Para medir el grado de funcionamiento de los algoritmos implementados, además de las técnicas de persuasión y la satisfacción del usuario al interactuar con el robot, se evalúan y cuantifican cuatro parámetros fundamentales:

- Entretenimiento.
- Actividades realizadas.
- Nivel de persuasión.
- Experiencia que se tuvo ante la interacción con un robot.

Los parámetros de prueba se cuantifican de acuerdo al nivel de agrado que se tenga por parte del usuario mediante una encuesta, la cual contiene preguntas, donde su calificación es numérica, aunque no contiene unidad alguna, permite valorar el nivel de satisfacción para cada uno de los parámetros mencionados.

La encuesta consta de diez preguntas, de las cuales siete deben ser valorados en una escala del uno al cinco mientras que las tres restantes serán de sí o no. El cuestionario se compone de las siguientes preguntas:

- ¿Considera usted que el robot es entretenido?
- ¿Fueron de su agrado los chistes interpretados por el robot?
- ¿Fue de su agrado los bailes que realizó el robot?
- ¿Le parecieron interesantes las historias contadas?
- ¿Le gusto cuando el robot cantó?
- ¿Los movimientos y gestos usados motivaban a realizar las actividades que proponía el robot?
- ¿El robot le convencía de realizar las actividades de entretenimiento propuestas cuando hablaba?
- ¿Le gusto a usted que el robot le pueda reconocer y saludar cuando lo vea?
- ¿Le gusto experiencia de interactuar con un robot?
- ¿Volvería a realizar otra interacción con el robot?

Estas preguntas fueron planteadas con el objetivo de analizar los parámetros que se describen a continuación:

- **Evaluación de Entretenimiento.**

El parámetro de entretenimiento tiene como objetivo validar todo el sistema y conocer si para los usuarios la interacción con un robot humanoide se tornó en un ámbito entretenido en general, después de haber realizado la interacción y observado las actividades que realiza el robot NAO.

- **Evaluación de las actividades.**

Las actividades que realiza el robot se dividen en chistes, bailes, historias y canciones, las cuales están inmersas cada una en una pregunta de la encuesta realizada y tienen su propia valoración, por

ello para el análisis total se obtiene una media de todos los valores obtenidos y se representa en un solo valor numérico.

- **Evaluación de persuasión**

Los parámetros de persuasión se enfocan en conocer si los movimientos y gestos usados por el robot, influyen o no en la aceptación de los usuarios al momento que el robot NAO propone realizar las actividades que tiene programadas, además de eso también se plantea si el diálogo que mantiene el robot guía de una misma manera al usuario y persuadirlo de aceptar las actividades propuestas.

- **Evaluación de experiencia**

Se califica si la experiencia que tiene el usuario al interactuar con un robot de una manera natural, manteniendo un diálogo y observando como realiza las actividades para el cual fue programado, además de conocer si en un futuro volvería a interactuar de la misma manera con un robot.

## **5.2 Análisis de resultados**

Los resultados obtenidos se realizan a partir de las respuestas dadas por los usuarios que interactuaron con el robot NAO y son analizadas de acuerdo a cada grupo de edad ya planteado.

Las pruebas experimentales se llevan a cabo en el laboratorio CIM-2000 del departamento de Eléctrica y Electrónica de la Universidad de las Fuerzas Armadas-ESPE.



**Figura 39.** Interacción con usuarios entre 7 a 12 años.



**Figura 40.** Interacción con usuarios entre 13 a 19 años.



**Figura 41.** Interacción con usuarios entre 20 a 29 años



**Figura 42.** Interacción con usuarios entre 30 a 59 años.



**Figura 43.** Interacción con usuarios entre 60 a 80 años.

### 5.2.1 Resultados de desempeño para cada rango de edades.

Los resultados obtenidos para tres de los cuatro parámetros analizados se tabulan y agrupan de acuerdo a los valores dados por cada usuario en las pruebas experimentales y se representan en las tablas que se muestran a continuación:

**Tabla 11.**

*Datos obtenidos de los usuarios de 7 a 12 años.*

	<b>Entretenimiento</b>	<b>Actividades</b>	<b>Persuasión</b>
<b>Usuario 1</b>	4,0	4,75	2,0
<b>Usuario 2</b>	3,0	4,25	2,5
<b>Usuario 3</b>	3,0	3,0	2,0
<b>Usuario 4</b>	4,0	4,0	4,5
<b>TOTAL</b>	4,25	4,0	3,0

**Tabla 12.**

*Datos obtenidos de los usuarios de 13 a 19 años.*

	<b>Entretenimiento</b>	<b>Actividades</b>	<b>Persuasión</b>
<b>Usuario 1</b>	5,0	4,25	5,0
<b>Usuario 2</b>	4,0	3,0	2,0
<b>Usuario 3</b>	5,0	4,0	4,0
<b>Usuario 4</b>	5,0	4,0	4,5
<b>TOTAL</b>	4,75	3,81	3,88

**Tabla 13.***Datos obtenidos de los usuarios de 20 a 29 años de edad.*

	<b>Entretención</b>	<b>Actividades</b>	<b>Persuasión</b>
<b>Usuario 1</b>	5,0	3,25	4,5
<b>Usuario 2</b>	5,0	4,25	4,0
<b>Usuario 3</b>	5,0	3,25	4,0
<b>Usuario 4</b>	5,0	4,0	4,0
<b>TOTAL</b>	5,0	3,69	4,13

**Tabla 14.***Datos obtenidos de los usuarios de 30 a 59 años de edad.*

	<b>Entretención</b>	<b>Actividades</b>	<b>Persuasión</b>
<b>Usuario 1</b>	5,0	4,75	4,0
<b>Usuario 2</b>	5,0	4,25	4,5
<b>Usuario 3</b>	5,0	4,0	4,0
<b>Usuario 4</b>	4,0	4,5	4,5
<b>TOTAL</b>	4,75	4,38	4,25

**Tabla 15.***Datos obtenidos de los usuarios de 60 a 80 años de edad.*

	<b>Entretención</b>	<b>Actividades</b>	<b>Persuasión</b>
<b>Usuario 1</b>	5,0	4,0	4,0
<b>Usuario 2</b>	5,0	4,5	5,0
<b>Usuario 3</b>	5,0	4,5	4,5

CONTINÚA 

<b>Usuario 4</b>	5,0	5,0	5,0
<b>TOTAL</b>	5,0	4,5	4,63

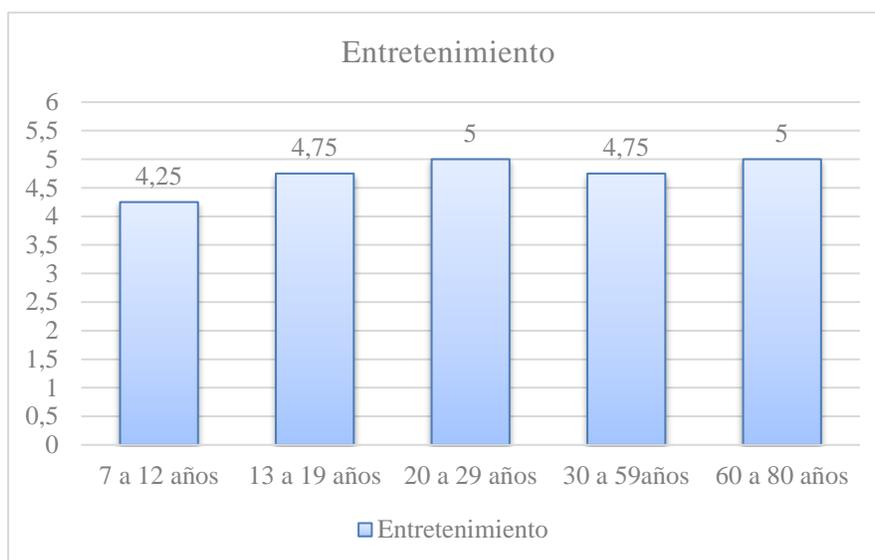
De los valores totales obtenidos para cada rango de edad, se los agrupa y se los representa en la Tabla 16, haciendo factible el análisis comparativo de los parámetros planteados, comparándolos con cada grupo de usuarios, con el fin de observar la variación de los datos con respecto al nivel de entretenimiento que se observa, cuan agradable son las actividades que realiza el robot y el nivel de persuasión que este tuvo sobre los participantes.

**Tabla 16.**

*Resultados mostrados por cada grupo de edad.*

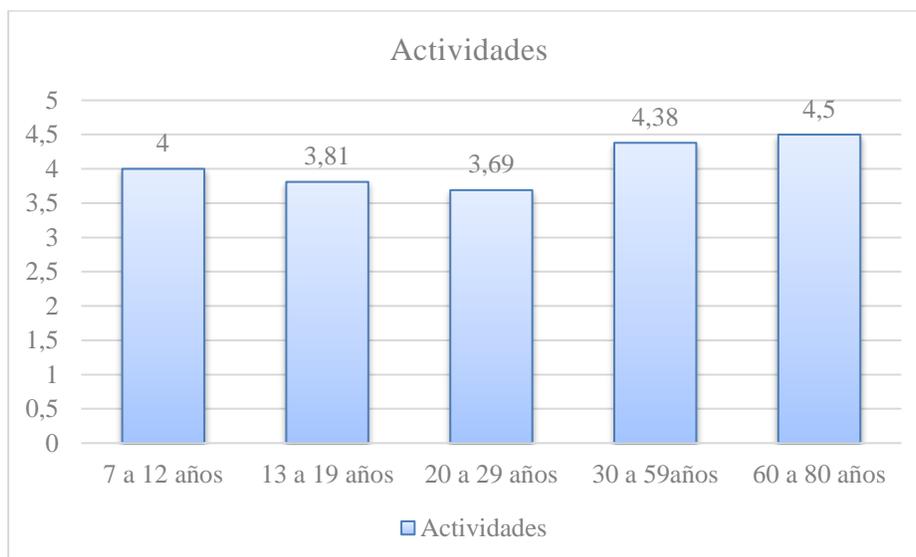
<b>Rango</b>	<b>Entretenimiento</b>	<b>Actividades</b>	<b>Persuasión</b>
<b>7 a 12 años</b>	4,25	4,0	3,0
<b>13 a 19 años</b>	4,75	3,81	3,88
<b>20 a 29 años</b>	5,0	3,69	4,13
<b>30 a 59 años</b>	4,75	4,38	4,25
<b>60 a 80 años</b>	5,0	4,5	4,63

En la Figura 44 se observa los valores que cada grupo de usuarios asigna al parámetro de entretenimiento, donde se puede apreciar mínimas variaciones cerca del límite de mayor ponderación, lo cual muestra que el robot maneja la temática de entretenimiento de manera óptima para todos los rangos de edades analizados.



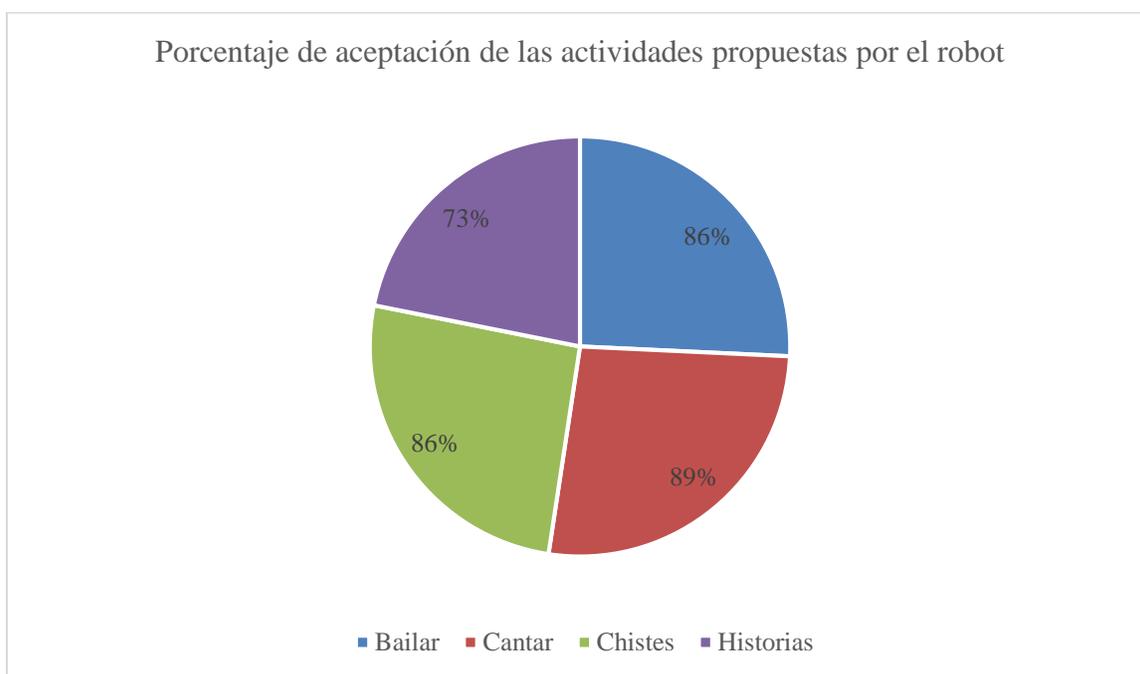
**Figura 44.** Gráfica de aceptación del parámetro de entretenimiento.

Como se muestra en la Figura 45, los datos obtenidos sobre las actividades que realiza el robot son del agrado de los usuarios, mostrando mayor empatía para aquellos cuyo rango de edad va de los 60 a 80 años.



**Figura 45.** Gráfica de aceptación del parámetro Actividades

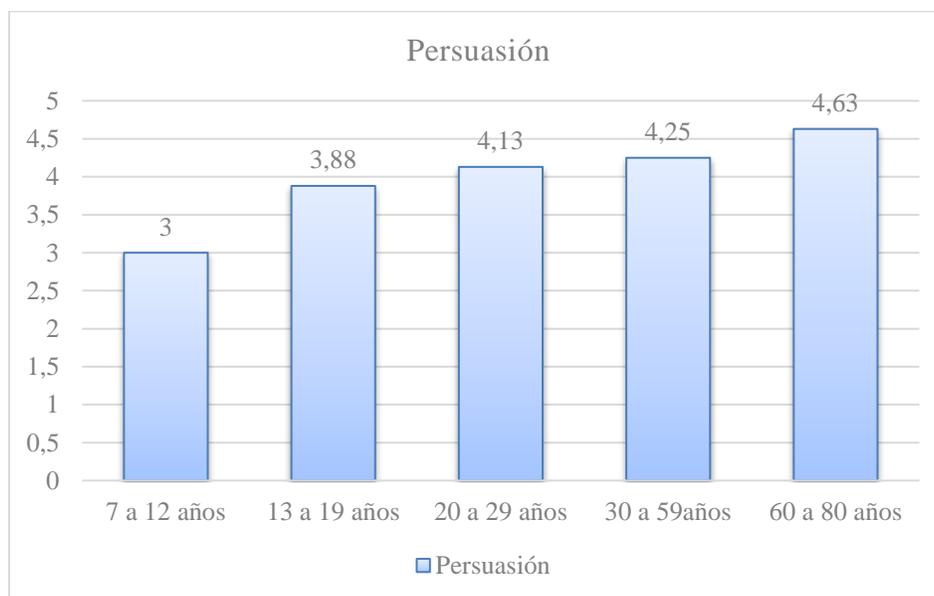
De las actividades programadas para que sean ejecutadas por el robot, se analizan los datos individuales para indicar que actividad es la que mayor aceptación tiene por parte de los usuarios y como se puede observar en la Figura 46. Observar al robot cantar es la actividad que causo mayor empatía en las personas con una aceptación del 89%, seguido de los bailes y los chistes con una valoración igual del 86% y por último con una aceptación del 73% se tiene la actividad de contar historias.



**Figura 46.** Gráfica del porcentaje de aceptación las actividades propuestas por el robot

Uno de los parámetros más importantes que se maneja en este proyecto de investigación, es el nivel de persuasión que observan los usuarios al interactuar con el robot NAO. Los datos que se muestran en la Figura 47, indican en qué grado las técnicas de persuasión implementadas sobre el robot influenciaron sobre los usuarios con el fin de que estos acepten que se realicen las actividades propuestas.

Como se observa, el nivel de influencia en los niños de menor edad se torna a un valor por debajo de los demás rangos analizados, esto se debe a que los usuarios en este rango de edad buscan mayores respuestas, que en muchos casos se apartan de la temática que maneja el robot.



**Figura 47.** Gráfica de aceptación del parámetro Persuasión

### 5.2.2 Resultados sobre la experiencia de interacción con un robot.

La experiencia que se tiene al interactuar con el robot se analiza con las tres últimas preguntas que se plantean en la encuesta realizada.

- El cien por ciento de los usuarios encuestados les gusto que el robot lo pueda reconocer y sea capaz de saludarlo cuando lo vea usando el nombre con el cual se presentó, esto ayudo a mejorar la experiencia de interacción humano-robot debido a que muestra un nivel de inteligencia ante los usuarios, haciéndolo más amigable y transmitiendo mayor confianza.
- La experiencia de interactuar con un robot fue muy aceptada, debido a que el cien por ciento de los usuarios mostro su agrado ante este hecho, debido a que mantener un diálogo en español

con una máquina que es capaz de entender e interpretar las respuestas del usuario, permiten crear nuevas experiencias y expectativas agradables sobre las nuevas tecnologías que se están desarrollando.

- Al plantear a cada usuario si volvería a realizar otra interacción con el mismo robot, en su totalidad todos estuvieron de acuerdo en que si lo harían, con expectativas de observar nuevas actividades u observar cómo es su comportamiento ante respuestas cambiantes del usuario.

### **5.3 Trabajos futuros**

El presente proyecto puede traer consigo investigaciones más avanzadas sobre inteligencia artificial, interacción social, persuasión mediante modulación de voz y mayor rango en reconocimiento verbal.

La visión de este proyecto a futuro se enfoca en el desarrollo de un sistema de diálogos más avanzado que contenga tanto una mayor cantidad de vocabulario, terminologías, conceptos, reglas y expresiones idiomáticas que permitan ampliar los rangos de respuestas que se den por parte de un robot e indexarlo a internet con el objetivo de obtener mayor cantidad de información y ante preguntas que no se contemplen dentro del sistema de diálogo, buscar e interpretar de manera autónoma posibles respuestas.

El sistema desarrollado es escalable, es decir se pueden agregar mayor cantidad de actividades o nuevas funciones que no se centren únicamente en la que ya fueron programadas, sino modularmente se pueden ir agregando una mayor cantidad de estas con el fin de tener múltiples opciones de acuerdo a los gustos de cada usuario y aumentar las valoraciones en relación al nivel de entretenimiento, como es la temática tratada en este proyecto, así también, se puede desarrollar

sistemas de dialogo nuevo que contengas diversos contenidos en la cual los usuarios decidan el foque del tema a tratar.

Dentro de este aspecto es factible la investigación que permita añadir funciones que tengan como objetivo el aprendizaje autónomo de un robot para ampliar sus sistemas de diálogo y rango de reconocimiento vocal.

Esta investigación abre puertas al desarrollo de sistemas de colaboración humano-robot o robot-robot con el fin de cumplir tareas específicas que lleven a un objetivo, tratando más a fondo los sistemas de interacción social y persuasión.

## CAPÍTULO VI

### CONCLUSIONES Y RECOMENDACIONES

#### 6.1 Conclusiones

En la etapa de selección de la plataforma de desarrollo para la programación del robot NAO, se realizó una comparación los sistemas ROS y QIBUILD, destacando las características que contiene cada uno de estos, en relación al tipo de comunicación, compatibilidad, soporte tanto de librerías como de distintos lenguajes de programación, destacando el uso del sistema QIBUILD por sus ventajas que brinda al ser diseñado para trabajar específicamente con este tipo de robots, optimizando recursos de software y facilitando el trabajo de comunicación con el hardware.

Se desarrollaron funciones que permiten realizar una interacción Humano-Robot por medio de una comunicación verbal.

Se implementó estrategias de persuasión mediante el uso de los algoritmos como los de reconocimiento facial, sistemas de diálogo y generando movimientos que permitan tanto, expresar emociones, realizar gestos o señas y también programarlas en una secuencia tal que vayan acorde a las actividades que deben ser ejecutadas por el robot.

Se desarrolló actividades acordes a la temática de entretenimiento planteada en este proyecto, como son: cantar, bailar, contar historias y chistes, los cuales están enfocados en divertir y entretener a los usuarios que interactúen con el robot.

Se diseñó un diálogo tal que permite enlazar las expresiones corporales y gestos diseñados para emular sentimientos y así lograr persuadir al usuario que se encuentra interactuando con el robot, para elegir cualquier actividad que este propone como forma de entretenimiento o influenciándolo para aceptar las opciones que cada actividad posee.

Se realizan pruebas experimentales con una muestra de veinte personas que se encuentran dentro de un rango de edad de entre los 7 a los 80 años, por lo que son agrupándolos entre niños, adolescentes, jóvenes, adultos y adultos mayores, con el fin de conocer el nivel de aceptación que tiene cada uno de estos grupos al interactuar con un robot.

Se realiza el análisis de cuatro parámetros fundamentales que han sido cuantificados por cada usuario, estos parámetros son: nivel de entretenimiento, nivel de aceptación de las actividades, nivel de persuasión y la experiencia de interactuar con un robot. Los datos son tabulados y agrupados por los rangos de edades establecidos, concluyendo que el entretenimiento para todos los grupos establecidos es satisfactorio, pues sus datos varían dentro del límite máximo de satisfacción, las actividades también tienden a los mismos resultados, siendo la actividad cantar la más aceptada por los usuarios y las historias la menos popular, dentro del análisis de persuasión se observan niveles bajos de aceptación de este parámetro en los niños y mejora acorde la edad se incrementa y finalmente la experiencia de interactuar con un robot fue muy aceptada por todos los usuarios, donde el 100% de estos les gusto la experiencia de interacción y lo volverían hacer en un futuro.

Al momento de realizar la experimentación se observó una clara diferencia entre trabajar con hombres y mujeres, debido a que el sistema de detección del habla, responde mejor ante tonos de voz graves debido a que se facilita su procesamiento.

Se analiza el nivel de persuasión y los valores bajos que se dan para este parámetro en los usuarios de 7 a 12 años y de acuerdo a la experimentación se observó que esto se debe a que los niños de menor edad tratan de establecer un diálogo propio, saliéndose del enfoque y la temática que mantiene el robot, dificultando dar respuestas ante temáticas no programadas.

## **6.2 Recomendaciones**

Se recomienda que el uso del robot para futuras interacciones, se lleven a cabo en ambientes con poco nivel de ruido, debido a que este fenómeno afecta en gran medida a los sistemas de detección y procesamiento de la voz.

Es recomendable que para la programación de los movimientos que se vayan a realizar con el robot, previamente se realicen pruebas en un simulador, con el fin de evitar caídas o movimientos no esperados en el robot real.

Al utilizar el módulo de reconocimiento facial, es recomendable hacerlo bajo ambientes con una buena cantidad de luz, para que los datos del rostro no se vean alterados y tener un óptimo valor de umbral de detección.

Se recomienda el uso del robot en periodos de tiempo cortos, debido a que el uso constante recalienta los motores en las articulaciones, provocando alertas en los sensores de temperatura y enviando alertas que interrumpen las actividades que se estén realizando en ese momento con el robot.

## REFERENCIAS BIBLIOGRÁFICAS

- Aldebaran Robotics. (04 de Junio de 2018). *Aldebaran documentation*. Obtenido de <http://doc.aldebaran.com/>
- Alonso, F. (2014). *Sistema de interacción humano-robot basado en diálogos multimodales y adaptables*. Madrid: Universidad Carlos III de Madrid.
- Anderson, k., & Owan, P. (2003). *Real-Time emotion recognition using biologically inspired models*. In proceeding of the 4th international conference on audio and video based biometric person authentication.
- Brenna, A., & Aude, B. (2010). A survey of Tactile Human-Robot Interactions. *Elsevier B.V.*, 18.
- Chidambaram, V., Chiang, Y., & Mutlu, B. (2012). Designing Persuasive Robots: How Robots might persuade people using vocal and nonverbal cues. *University of Wisconsin-Madison*.
- Eric Fujiwara, D. M. (2013). Development of a Glove-Based Optical Fiber Sensor for Applications in Human-Robot Interaction. *Proc. 8th ACM/IEEE HRI*.
- Gemignani, G., Capobianco, R., Bastianelli, E., Bloisi, D., Iocchi, L., & Nardi, D. (2016). Living with robots: Interactive environmental knowledge acquisition. *Robotics and Autonomous Systems*.
- González E, J., & Jiménez B, J. (2009). La robótica como herramienta para la educación en ciencias e ingeniería. *Revista Iberoamericana de Informática*, 6.
- Goodrich , M., & Schultz, A. (2007). Human-Robot Interaction: A survey. *Foundations and Trends in Human-Computer Interaction*.

- Ham, J., Cuijpers, R., & Cabibihan, J. (2015). Combining Robotic Persuasive Strategies: The Persuasive Power of a Storytelling Robot that Uses Gazing and Gestures. *Int J of Soc Robotics*.
- Honda Motor Co. (06 de junio de 2018). *ASIMO*. Obtenido de ASIMO: <https://www.honda.mx/asimo/>
- Ismail, L., Shamsuddi, S., Yussof, H., & Jaafar, A. (2011). Face Detection Technique of Humanoid Robot NAO for Application in Robotic Assistive Therapy. *IEEE*, 5.
- Kanda, T., Hirano, T., & Eaton, D. (2004). Interactive Robots as Social Partners and Peer Tutors for Children: A Field Trial. *HUMAN-COMPUTER INTERACTION*, 61-84.
- Kirby, R., Forlizzi, J., & Simmons, R. (2010). Affective social robots. *Robotics and Autonomous Systems*, vol. 58, 322-332.
- Lucas, J. M., Alcázar, R., Montero, J. M., Fernández, M., Ferreiros, J., Macías, J., & Pardo, J. (2008). Desarrollo de un Robot-Guía con Integración de un sistema de diálogo y expresión de emociones. *Journal*, 8.
- Millán Romera, J. A. (2015). *Reconocimiento gestual para interacción humano-robot basado en ROS*. Sevilla: Universidad de Sevilla.
- Monajjemi, V., Wawerla, J., Vaughan, R., & Mori, G. (2013). HRI in the Sky: Creating and Commanding Teams of UAVs with a Vision-mediated Gestural Interface. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Rehj, C., Matthias, S., Paul, S., & Xuan, W. (2010). Robust Spoken Instruction Understanding for HRI. *In Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction*.

- Richmond, V., Gorham, J., & McCroskey, J. (1987). The relationship between selected immediacy behaviors and cognitive learning. *West Virginia University, Communication yearbook*.
- Robins, B., Dautenhahn, K., Boekhorst, R., & Billard, A. (2005). Robotic Assistants in Therapy and Education of Children with Autism. *University of Hertfordshire, 20*.
- RoboCup Federation. (28 de mayo de 2018). *RoboCup*. Obtenido de <http://www.robotcup.org/>
- Robotnik. (15 de Diciembre de 2017). *Robotnik*. Obtenido de Entretenimiento: <https://www.robotnik.es/servicios-de-robotica-proyectos/aplicaciones-moviles/entretenimiento/>
- Robotronica. (12 de junio de 2018). *Aliverobots*. (Professional Robotics Solutions ) Recuperado el 16 de 12 de 2017, de <http://aliverobots.com/nao/>
- ROS (Robot Operating System). (07 de junio de 2018). *ROS*. Obtenido de <http://www.ros.org/>
- Scheutz, M. (2009). The Inherent Dangers of Unidirectional Emotional Bonds between Humans and Social Robots. *Proc. IEEE Int'l Conf. Robotics and Automation*.
- Serrano, D. (2011). *Introduction to ROS - Robot Operating System*. Obtenido de <https://www.sto.nato.int/publications/...EN.../EN-SCI-271-06.pdf>
- Shamsuddin, S., Yussof, H., Ismail, L., & Mohamed, S. (2012). *Initial Response of Autistic Children in Human-Robot Interaction Therapy with Robot NAO*. Malaysia: IEEE.
- Siegel, M., Breazeal, C., & Norton, M. (2009). Persuasive Robotics: The Influence of Robot Gender on Human Behavior. *IEEE/RSJ International Conference on. 2009.*, 7.
- Siegel, M., Breazeal, C., & Norton, M. (2009). Persuasive Robotics: The Influence of Robot Gender on Human Behavior. *IEEE/RSJ International Conference on Intelligent Robots and System*.

Sony Corporation. (12 de junio de 2018). *aibo a new story begins*. Obtenido de <https://aibo.sony.jp/en/>

Ubuntu homepage. (07 de junio de 2018). *Ubuntu*. Obtenido de <https://www.ubuntu.com>

Zhang, L., Jiang, M., Farid, D., & Hossain, M. (2013). Intelligent facial emotion recognition and semantic-based topic detection for a humanoid robot. *ELSERVER*, 1-3.