

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA ELECTRÓNICA**

**Trabajo de titulación previo a la obtención del título de:
INGENIERA ELECTRÓNICA E INGENIERO ELECTRÓNICO**

**TEMA:
DESARROLLO DE UN SISTEMA PARA ADMINISTRAR MEDICAMENTOS
A ADULTOS MAYORES A TRAVÉS DEL RECONOCIMIENTO DE
CÓDIGO DE BARRAS Y SU IMPLEMENTACIÓN EN EL ROBOT NAO**

**AUTORES:
GABRIEL ERNESTO NAVAS REASCOS
CAROLINA ESTEFANÍA RIVERA BONIFAZ**

**TUTOR:
VÍCTOR VINICIO TAPIA CALVOPIÑA**

Quito, agosto del 2017

CESIÓN DE DERECHOS DE AUTOR

Nosotros Gabriel Ernesto Navas Reascos y Carolina Estefanía Rivera Bonifaz, con documentos de identificación N° 172048748-5 y N° 172506250-7 respectivamente, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: DESARROLLO DE UN SISTEMA PARA ADMINISTRAR MEDICAMENTOS A ADULTOS MAYORES A TRAVÉS DEL RECONOCIMIENTO DE CÓDIGO DE BARRAS Y SU IMPLEMENTACIÓN EN EL ROBOT NAO; mismo que ha sido desarrollado para optar por el título de: Ingeniera e Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Quito, agosto 2017



Carolina Estefanía Rivera Bonifaz

Cédula: 172506250-7



Gabriel Ernesto Navas Reascos

Cédula: 172048748-5

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación DESARROLLO DE UN SISTEMA PARA ADMINISTRAR MEDICAMENTOS A ADULTOS MAYORES A TRAVÉS DEL RECONOCIMIENTO DE CÓDIGO DE BARRAS Y SU IMPLEMENTACIÓN EN EL ROBOT NAO realizado por Carolina Estefanía Rivera Bonifaz y Gabriel Ernesto Navas Reascos, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto 2017



Víctor Vinicio Tapia Calvopiña

CI: 1708547219

DEDICATORIA

Al culminar mis estudios universitarios dedico este logro a mis padres Marco e Irene de quienes recibí un apoyo incondicional a lo largo de mi formación profesional, a mis hermanas Gabriela y Nathalia que siempre han creído en mí y me han dado el ánimo para seguir adelante y en especial a mi abuelito que desde el cielo siempre me ha guiado, protegido y que en su memoria nunca decaí hasta obtener mi título.

Carolina Estefanía Rivera Bonifaz

Dedico este proyecto de titulación primero a mis padres Gustavo y Elena que gracias a su amor, guía, dedicación y cariño día a día me han permitido cumplir con una meta más en mi vida, por siempre motivarme y darme fuerzas para seguir adelante; y a mi hermano que siempre ha sido un apoyo y pilar en mi vida.

Gabriel Ernesto Navas Reascos

AGRADECIMIENTO

Agradecemos a la Universidad Politécnica Salesiana por la formación académica y humana recibida, así como a cada uno de los docentes que nos brindaron sus conocimientos y guiaron a lo largo de nuestra formación profesional.

Al Ingeniero Vinicio Tapia por su calidad humana y por el tiempo dedicado para el desarrollo del proyecto.

Además un agradecimiento especial al Ingeniero Gustavo Navas que nos enseñó temas necesarios para la culminación de este proyecto.

ÍNDICE DE CONTENIDO

CAPITULO 1	1
ANTECEDENTES	1
1.1 Planteamiento del problema.....	1
1.2 Justificación del proyecto.....	1
1.3 Objetivos.....	2
1.3.1 Objetivo general.....	2
1.3.2 Objetivos específicos.....	2
CAPÍTULO 2	3
MARCO CONCEPTUAL	3
2.1 Trastorno mental.....	3
2.2 Robot NAO.....	3
2.2.1 Arquitectura y especificaciones técnicas.....	4
2.2.2 Análisis del movimiento del sistema.....	5
2.2.3 Matriz de rotación.....	7
2.2.4 Análisis matemático de la cinemática inversa.....	8
2.3 Arduino.....	17
2.4 Comparación entre placas Arduino.....	18
2.5 Arduino USB Host Shield.....	18
2.6 Cámara Robot NAO.....	19
2.7 Código de barras.....	20
2.7.1 Dispositivos de Entrada.....	20
2.7.2 Decodificadores.....	21
2.8 Lector de código de barras.....	22
2.9 Pistola Marca One.....	23
2.10 Reconocimiento de rostros.....	24
2.11 NAOmarks.....	25
CAPÍTULO 3	27
DISEÑO E IMPLEMENTACIÓN	27
3.1 Diagrama de interconexión.....	27
3.2 Transmisión del código de barras Arduino-Robot NAO.....	27

3.3 Base de datos.....	32
3.4 Activación controlada por el robot NAO a la pistola lectora de código de barras.....	34
3.5 Entorno creado para el robot NAO.....	38
3.6 Movilización del robot usando NAOMarks.....	38
CAPÍTULO 4.....	42
PRUEBAS Y ANÁLISIS DE RESULTADOS.....	42
4.1 Movilidad en el entorno.....	44
4.2 Funcionamiento.....	44
CONCLUSIONES	52
RECOMENDACIONES.....	54
LISTA DE REFERENCIAS.....	55

ÍNDICE DE FIGURAS

Figura 2.1. Hardware del robot NAO.....	4
Figura 2.2. Alcance cámaras vista lateral y superior.....	4
Figura 2.3. Micrófonos NAO Frontales y posteriores.....	5
Figura 2.4. Rango de desplazamiento de la cabeza.....	6
Figura 2.5. Grados de libertad del brazo derecho.....	6
Figura 2.6. Rango desplazamiento para pierna derecha.....	6
Figura 2.7. Sistema de tres ejes.....	7
Figura 2.8. Grados de libertad de la cabeza.....	8
Figura 2.9. Grados de libertad del brazo.....	9
Figura 2.10. Grados de libertad de la pierna.....	12
Figura 2.11. Código de barras.....	22
Figura 2.12. Funcionamiento.....	23
Figura 2.13. Lector de código de barras.....	24
Figura 2.14. Sistema de escaneo de código de barras.....	24
Figura 2.15. Reconocimiento de rostros.....	25
Figura 2.16. Distintos patrones de NAOmarks.....	26
Figura 3.1. Diagrama de la interfaz de conexión.....	27
Figura 3.2. Diagrama de Flujo.....	27
Figura 3.3. Diagrama de Flujo.....	28
Figura 3.4. Conexión Arduino USB Host Shield - Lector de código de barras.....	29
Figura 3.5. Puerto USB Robot NAO.....	29
Figura 3.6. Código de envío del código de barras hacia el robot NAO.....	30
Figura 3.7. Interconexión Robot NAO Arduino en Choregraphe.....	30
Figura 3.8. Interconexión Robot NAO Arduino en Phytton.....	31
Figura 3.9. Integración de datos en un solo vector.....	32
Figura 3.10. Archivo con la información de los pacientes.....	32
Figura 3.11. Acceso a la información del archivo de texto.....	33
Figura 3.12. Comparación entre códigos de barras.....	33
Figura 3.13. Programación para buscar tres veces la medicación.....	34
Figura 3.14. Pistola lectora código de barras desarmada.....	34
Figura 3.15. Pistola lectora código de barras desarmada.....	35
Figura 3.16. Pistola lectora código de barras desarmada.....	35

Figura 3.17. Cable soldado al pulsador.....	36
Figura 3.18. Señal activación desde Arduino a lectora de código de barras.....	36
Figura 3.19. Programación en Python para transmitir al Arduino.....	37
Figura 3.20. Programación en el Arduino.....	37
Figura 3.21. Programación para repetir la búsqueda del código de barras.....	38
Figura 3.22. Entorno creado para el robot NAO.....	38
Figura 3.23. Movilidad en el entorno.....	39
Figura 3.24. Reconocimiento del paciente.....	40
Figura 3.25. Entrega de la medicina al paciente.....	40
Figura 3.26. Proceso realizado por el robot.....	41
Figura 4.1. Entorno real diseñado para el robot.....	43
Figura 4.2. Identificación de la NAOmark 68.....	44
Figura 4.3. Giro a la izquierda.....	44
Figura 4.4. Identificación de la NAOmark 130.....	45
Figura 4.5. Giro a la izquierda y posición de espera.....	45
Figura 4.6. Búsqueda del medicamento.....	46
Figura 4.7. Lectura del código de barras medicamento incorrecto.....	46
Figura 4.8. Lectura del código de barras medicamento correcto.....	47
Figura 4.9. Toma del medicamento (prueba del medicamento cambiado).....	47
Figura 4.10. Lectura sin medicamento.....	48
Figura 4.11. Lectura con medicamento.....	48
Figura 4.12. Toma del medicamento.....	49
Figura 4.13. Robot dirigiéndose a buscar al paciente.....	49
Figura 4.14. NAO buscando al paciente “José”	50
Figura 4.15. NAO encuentra al paciente “José”.....	50
Figura 4.16. NAO camina hacia el paciente “José”.....	51
Figura 4.17. Entrega de medicamento.....	51

ÍNDICE DE TABLAS

Tabla 2.1 Especificación del sensor.....	5
Tabla 2.2. Características de placas Arduino.....	18
Tabla 2.3. Características USB Host Shield.....	19
Tabla 2.4. Características de las cámaras del robot NAO.....	19
Tabla 4.1. Comparación entre pruebas de campo.....	43

RESUMEN

El proyecto trata sobre la intercomunicación entre el Robot NAO de la Universidad Politécnica Salesiana y un adulto mayor; con la finalidad de administrar medicamentos a través del reconocimiento de código de barras. Debido a que con el paso del tiempo el ser humano se enfrenta a factores sociales, psíquicos y biológicos que afectan su salud mental produciendo que en muchas ocasiones no tomen sus medicamentos a la hora prescrita e incluso los confundan siendo necesaria la presencia de otra persona que les brinde asistencia.

Por esta razón se planteó un proyecto con el robot NAO que ayude a reducir los errores en la toma de medicamentos de adultos mayores y por ende las consecuencias que conllevan aquellos errores.

En el presente documento se detalla qué es y cómo funciona un lector de código de barras, como realizar la comunicación entre el lector de código de barras y el robot NAO mediante la utilización de un Arduino y su complemento USB HOST SHIELD; adicionalmente se detalla cómo crear una base de datos dentro del robot NAO la cual contiene el código de barras del medicamento, la hora a la que debe ser administrada y el nombre del paciente.

Además se puntualiza el método utilizado en el proyecto para la movilización y navegación del robot NAO en un entorno creado, los algoritmos utilizados para el reconocimiento del rostro del paciente, las medidas y peso de los medicamentos recomendados para un óptimo funcionamiento y las pruebas realizadas así como los resultados obtenidos.

ABSTRACT

The project deals with the intercommunication between the Robot of the Salesian Polytechnic University and an older adult; in order to administer medicines through barcode recognition. Because over time human beings face social, psychic and biological factors that affect their mental health causing them to not take their medications at the prescribed time and even confuse them being necessary the presence of another person to assist them.

For this reason, was proposed a project with the NAO robot to help reducing mistakes in the consumption of medication in older adults and, the consequences of those mistakes.

The document details what is and how a bar code reader works, such as how to make the performing communication between the reader and the robot NAO, using an Arduino and its USB HOST SHIELD complement; Additionally, it is detailed how to create a database inside the robot NAO which contains the bar code of the medicine, the time to be administered and the name of the patient.

In addition, It is pointed out the method used in the project for the mobilization and navigation of the NAO robot in a created environment, the algorithms used for the recognition of the patient's face, the measurements and weight of the recommended medications for an optimal operation and the tests carried out as well As the results obtained.

INTRODUCCIÓN

La Organización Mundial de la Salud señala que más del 70% de los adultos mayores toman medicación, entre 2 a 6 medicamentos prescritos y de 1 a 3 de venta libre.

La incorrecta ingesta de medicamentos puede producir severos problemas a la salud e incluso llevar hasta la muerte; ya que muchos medicamentos prescritos para afecciones cardíacas o hipertensión arterial deben administrarse de forma constante. Se estima que existen alrededor de 1.5 millones de errores en la toma de medicamentos y miles de estos terminan produciendo la muerte de la persona.

El proyecto pretende ayudar a reducir los errores en la toma de medicamentos y por ende las consecuencias que conllevan aquellos errores. Se mejorará la calidad de vida del adulto mayor y minimizará los gastos por parte de las entidades sociales debido a los errores en la toma de medicamentos.

En el capítulo uno se definirá el planteamiento del problema, la justificación y los objetivos del proyecto. En el capítulo dos se realizará un estudio el cual contendrá cálculos matemáticos de la cinemática inversa del robot NAO y se investigará los materiales utilizados para la realización del proyecto.

En el capítulo tres, se desarrollarán los algoritmos utilizados, así como los diseños tanto de hardware como de software. Además se especificará como se realizó la implementación del proyecto.

En el capítulo cuatro se realizarán las pruebas respectivas con el robot en el entorno creado, donde se comprobará la efectividad del reconocimiento del paciente y la entrega de la medicación.

CAPÍTULO 1

ANTECEDENTES

1.1 Planteamiento del problema

Con el paso del tiempo el ser humano se enfrenta a factores sociales, mentales y biológicos que afectan su salud mental, esto se ve reflejado en su mayoría en adultos mayores quienes además padecen dolores crónicos, fragilidad y dificultades de movilidad. Por tales motivos no toman sus medicamentos a la hora prescrita e incluso los confunden; siendo necesaria la presencia de otra persona para que les brinde asistencia, es por esto que se decidió realizar el presente proyecto que ayudará a las personas que tienen discapacidad física o mental.

Para la realización del proyecto se utilizará el robot NAO de la Universidad Politécnica Salesiana del campus sur Quito, para que realice acciones como reconocer, recoger y entregar los medicamentos a los adultos mayores a las horas prescritas avanzando a través de un pasillo hasta dejar el medicamento en el lugar establecido. Adicionalmente comunicará verbalmente al adulto mayor que debe tomar el medicamento.

Además se creará una base de datos con medicamentos preestablecidos, por lo tanto, el sistema no hará reconocimiento de medicamentos que no estén previamente cargados en su base de datos.

1.2 Justificación del proyecto

Mediante la implementación de este proyecto se reducirá los errores en la toma de medicamentos y por ende las consecuencias que conllevan aquellos errores.

Además se contribuirá a mejorar la calidad de vida del adulto mayor y minimizará los gastos por parte de las entidades sociales dedicadas al cuidado de la salud de adultos mayores debido a descuidos o negligencia en el suministro de medicamentos. Igualmente se disminuirá la cantidad de personal capacitado que necesita el adulto mayor para que le asista.

Asimismo permitirá conocer más a fondo las capacidades del robot NAO y con estos conocimientos los estudiantes de la Universidad Politécnica Salesiana podrán realizar otros proyectos a futuro.

1.3 Objetivos

1.3.1 Objetivo General

Desarrollar un sistema para administrar medicamentos a adultos mayores a través del reconocimiento de código de barras y su implementación en el robot NAO.

1.3.2 Objetivos específicos

- Estudiar y seleccionar el lenguaje de programación necesario para la creación de los algoritmos que permitirán el reconocimiento del código de barras.
- Implementar una base de datos con los medicamentos, hora y paciente a ser administrados por el robot para mejorar la gestión de entrega de los medicamentos.
- Integrar el software de reconocimiento de código de barras, base de datos, sistema de navegación y movilidad en el entorno, en el robot NAO para lograr una buena asistencia médica.
- Realizar pruebas de campo para comprobar el funcionamiento del proyecto en un entorno creado para el robot.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1 Trastorno mental

Se conoce como trastorno mental al síndrome o a un patrón de carácter psicológico sujeto a interpretación clínica que, por lo general, se asocia a un malestar o a una discapacidad. En este marco, resulta interesante destacar que una enfermedad de tipo mental es aquella que se produce a raíz de una alteración que repercute sobre los procedimientos afectivos y cognitivos del desarrollo, la cual se traduce en dificultades para razonar, alteraciones del comportamiento, impedimentos para comprender la realidad y para adaptarse a diversas situaciones. Muchos son los signos o síntomas que dan muestra de que alguien está sufriendo un trastorno mental. Así, por ejemplo, entre los mismos se encuentran trastornos del sueño, tristeza, ansiedad, alteraciones en lo que es la memoria, dificultad palpable para pensar con absoluta claridad, conducta de tipo agresivo o incluso cambios y problemas en lo que es tanto la percepción visual como la auditiva. (Perez, J & Merino, M, 2012)

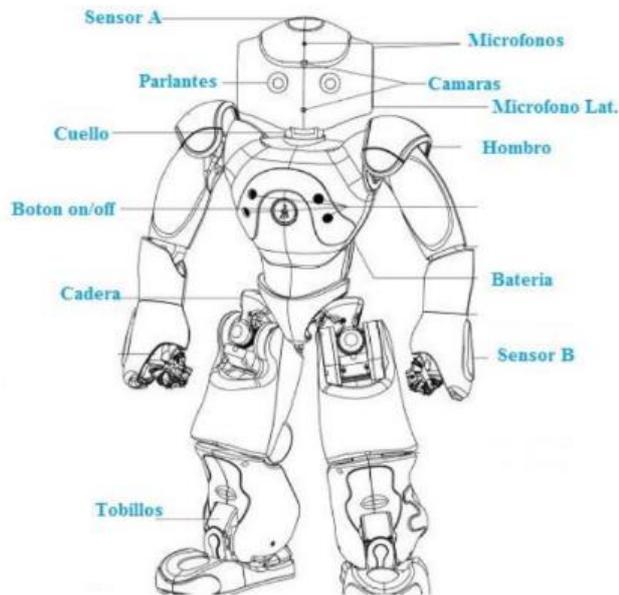
2.2 Robot NAO

El robot humanoide NAO tiene una altura de 58 centímetros, puede ser programado en su totalidad y posee una plataforma de programación amigable con el usuario. Fue desarrollado en el 2008 por la empresa Softbank Robotics.

NAO posee diversos sensores que le permiten conocer el entorno que le rodea consta de ocho sensores de presión, nueve sensores táctiles ubicados en sus manos y cabeza, dos sensores de ultrasonidos situados en su pecho, dos cámaras frontales, un acelerómetro, cuatro micrófonos en su cabeza, y un giróscopo.

Adicionalmente, posee 53 LEDs RGB distribuidos a lo largo de su cuerpo, dos altavoces y un sintetizador de voz. El robot se lo puede programar en un software propio de NAO llamado Choregraphe, el cual es compatible tanto con Windows como con Linux y Mac. Además, tiene compatibilidad con distintos lenguajes de programación como NET, C++, JAVA, Python y MATLAB. (Aliverobots, 2010)

Figura 2.1. Hardware del robot NAO



Arquitectura detallada del robot humanoide NAO.

Fuente: (Aldebaran, s.f.)

2.2.1 Arquitectura y especificaciones técnicas

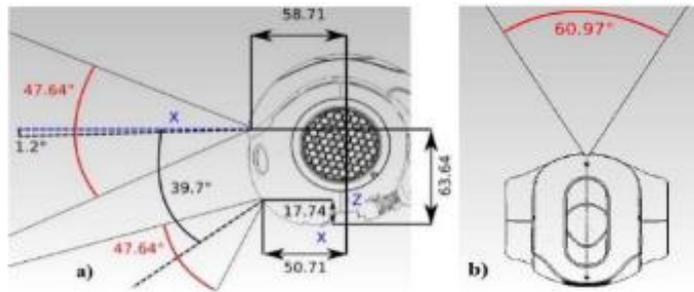
Cámaras Frontales

Posee dos cámaras frontales de una resolución de 1.22Mp, la primera ubicada en su frente la que permite tener una visibilidad frontal; la segunda se encuentra en su boca la misma que graba en dirección al piso. Las cámaras pueden cumplir funciones de identificación de objetos y reconocimiento de imágenes.

En la figura 2.2.a, se visualiza el ángulo de alcance en el eje z que puede visualizar el robot, el cual es de 47.64° grados.

En la figura 2.2.b, se representa el ángulo de alcance en el eje x que puede visualizar el robot, el cual es de 60.97° grados. (Aldebaran, s.f.)

Figura 2.2. Alcance cámaras vista lateral y superior

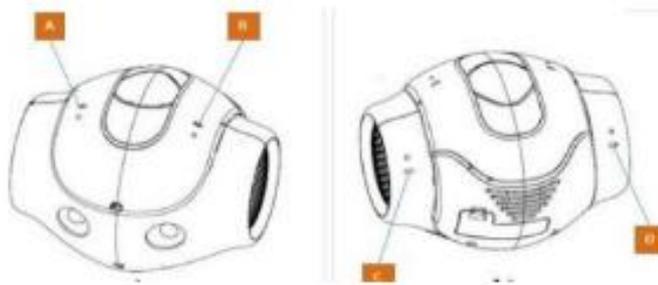


Vista lateral y superior del alcance en grados de las cámaras
Fuente: (Aldebaran, s.f.)

Micrófonos

La ubicación de los micrófonos del robot humanoide NAO se detallan en la figura 2.3.a, los micrófonos frontales o principales y en la figura 2.3.b, los micrófonos traseros o secundarios. (Aldebaran, s.f.)

Figura 2.3. Micrófonos NAO Frontales y posteriores



Ubicación de los micrófonos del androide
Fuente: (Aldebaran, s.f.)

Sensor A

En la figura 2,3 se puede observar la ubicación del sensor A y a continuación de describen sus características en la tabla 2.1.

Tabla 2.1. Especificación del sensor

Características destacadas
Señal digital
Se activan al presionarlo
Dato tipo flotante: (0.00)
Se lo programa fácilmente
(1) lógico activado
Error de precisión: 0.09%
Material: polietileno
Estabilidad: (0.02 ± 0.07)

Características del sensor ubicado en la cabeza del robot.

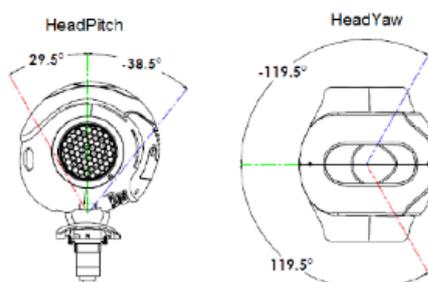
Fuente: (Aldebaran, s.f.)

2.2.2 Análisis del movimiento del sistema

Para comprender el movimiento del robot se debe analizar cada articulación, para poder definir el alcance máximo y mínimo que pueden desplazarse, los cuales están representados con un ángulo θ .

En las figuras 2.4, 2.5 y 2.6 se visualizan los diferentes desplazamientos del robot. (Robotics, 2014)

Figura 2.4. Rango de desplazamiento de la cabeza

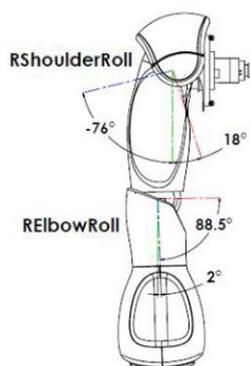


Rango de movimiento de la cabeza del robot

Fuente: (Robotics, 2014)

Los grados de libertad del brazo del robot se muestran en la figura 2.5

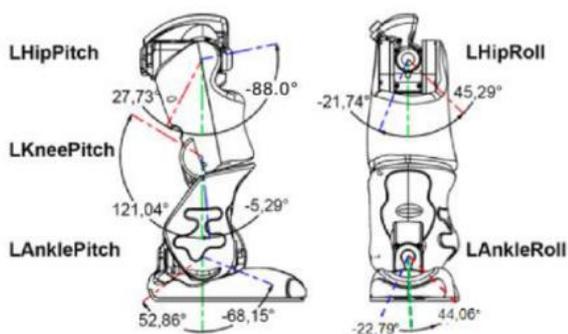
Figura 2.5. Grados de libertad del brazo derecho



Rango de movimiento para brazo derecho.

Fuente: (Robotics, 2014)

Figura 2.6. Rango desplazamiento para pierna derecha



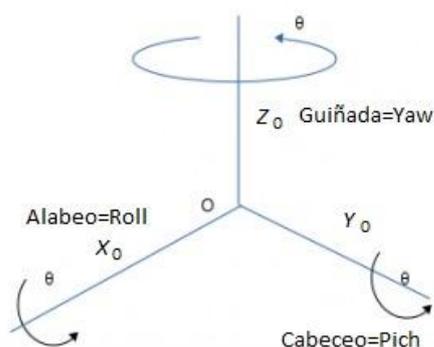
Rango de movimiento para pierna derecha.

Fuente: (Robotics, 2014)

2.2.3 Matriz de rotación

Los movimientos del robot mostrados en la figura 2.7 se pueden describir a través de una matriz de rotación en la cual es un modelo matemático resultante del producto de las rotaciones periódicas en los ejes x,y,z (Spong, M., Hutchinson, S., & Vidyasagar, M. , 2005)

Figura 2.7. Movimientos en los ejes X, Y, Z



Movimientos tridimensionales.

La matriz resultante se forma a partir de las ecuaciones 2.1 y 2.2, donde: S=Sin y C=Cos.

$$R_{xyz} = R_{z,\phi} R_{y,\theta} R_{z,\psi} \quad \text{Ec. (2.1)}$$

$$R_{xyz} = \begin{bmatrix} C_\phi & -S_\phi & 0 & 0 \\ S_\phi & C_\phi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_\theta & 0 & S_\theta & 0 \\ 0 & 1 & 0 & 0 \\ -S_\theta & 0 & C_\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C_\psi & -S_\psi & 0 \\ 0 & S_\psi & C_\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. (2.2)}$$

$$R_{xyz} = \begin{bmatrix} C_\phi C_\theta & -S_\phi C_\psi + C_\phi S_\theta S_\psi & S_\theta S_\psi + C_\phi S_\theta C_\psi & 0 \\ S_\phi C_\theta & C_\phi C_\psi + S_\phi S_\theta S_\psi & -C_\theta S_\psi + S_\phi S_\theta C_\psi & 0 \\ -S_\theta & C_\theta S_\psi & C_\theta C_\psi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. (2.3)}$$

La ecuación 2.3, se obtiene al multiplicar las tres matrices en la ecuación 2.2. (Spong, M., Hutchinson, S., & Vidyasagar, M. , 2005)

2.2.4 Análisis matemático de la cinemática inversa

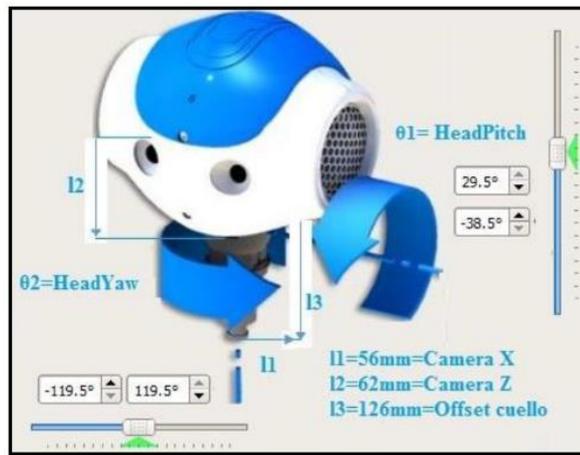
La cinemática inversa es un método que permite conocer los movimientos que pueden realizar cualquier tipo de robot. En la ecuación 2.4, describe la cinemática inversa de un robot humanoide. (Kofinas, 2012, julio)

$$D = \begin{bmatrix} C_\phi C_\theta & -S_\phi C_\psi + C_\phi S_\theta S_\psi & S_\theta S_\psi + C_\phi S_\theta S_\psi & px \\ S_\phi C_\theta & C_\phi C_\psi + S_\phi S_\theta S_\psi & -C_\theta S_\psi + S_\phi S_\theta C_\psi & py \\ -S_\theta & C_\theta S_\psi & C_\theta C_\psi & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. (2.4)}$$

A continuación, se escriben las ecuaciones en las diferentes articulaciones.

Cabeza: En la figura 2.8 se pueden identificar los ángulos de la cabeza L1, L2, L3 son los ejes X, Y, Z respectivamente. (Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (, 2014)

Figura 2.8. Grados de libertad de la cabeza



Rango de movimiento de la cabeza del robot

Fuente: (Robotics, 2014)

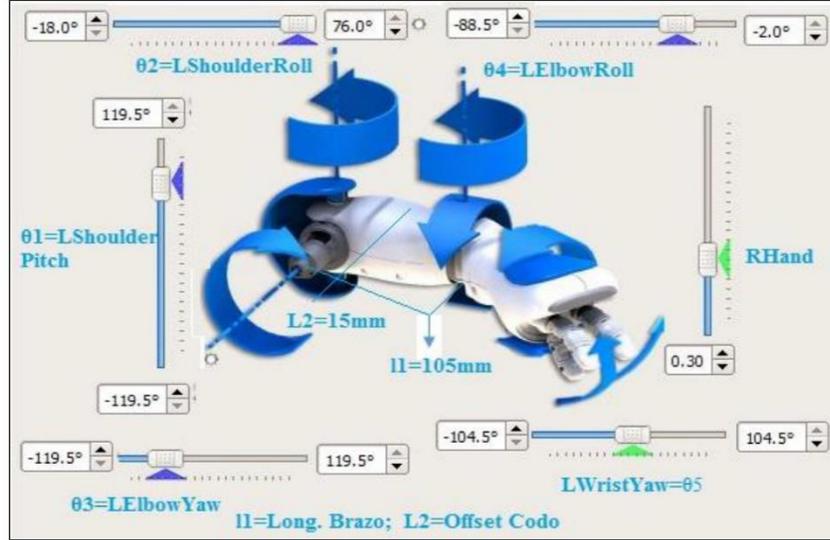
La ecuación 2.5 representa el movimiento de la cabeza en el eje z, mientras que la ecuación 2.6 representa el movimiento en el eje x.

$$\theta_1 = \left[\pm \operatorname{acos} \left(\frac{px}{L2 \cos\left(\theta_2 - \frac{\pi}{2}\right) - L1 \sin\left(\theta_2 - \frac{\pi}{2}\right)} \right) \right] \quad \text{Ec. (2.5)}$$

$$\theta_2 = \left(\pi - \operatorname{asin} \left(\frac{-pz + L3}{\sqrt{L1^2 + L2^2}} \right) - \operatorname{atan} \left(\frac{L1}{L2} \right) + \frac{\pi}{2} \right) \quad \text{Ec. (2.6)}$$

Brazo Izquierdo: En la figura 2.9, se puede ver las posiciones de ángulos. Dónde: L1= longitud brazo y L2=offset codo.

Figura 2.9. Grados de libertad del brazo



Rango de movimiento para brazo derecho.

Fuente: (Robotics, 2014)

$$D = A_{Base}^0 D_0^1 D_1^2 D_2^3 D_3^4 R_z \left(\frac{\pi}{2} \right) A_4^{End} \quad \text{Ec. (2.7)}$$

$$D' = (A_{Base}^0)^{-1} D (A_4^{End})^{-1} \left(R_z \left(\frac{\pi}{2} \right) \right)^{-1} \quad \text{Ec. (2.8)}$$

$$D'' = (D')^{-1} \quad \text{Ec. (2.9)}$$

$$D'' = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A continuación se describen las ecuaciones de la matriz:

$$r_{11} = \cos\theta_2 \sin\theta_1 \sin\theta_3 + \cos\theta_1 (\cos\hat{\theta}_2 \cos\theta_3 \cos\theta_4 - \sin\hat{\theta}_2 \sin\theta_4)$$

$$r_{12} = \cos\theta_3 \cos\theta_4 \sin\hat{\theta}_2 + \cos\hat{\theta}_2 \sin\theta_4$$

$$r_{13} = -\cos\hat{\theta}_2 \cos\theta_3 \cos\theta_4 \sin\theta_1 + \cos\theta_1 \cos\theta_4 \sin\theta_3 + \sin\theta_1 \sin\hat{\theta}_2 \sin\theta_4$$

$$r_{14} = -L1 \cos\theta_3 \cos\theta_4 + L2 \sin\theta_4$$

$$\begin{aligned}
r_{21} &= -\sin\theta_1\sin\theta_3\sin\theta_4 - \cos\theta_1(\cos\theta_4\sin\hat{\theta}_2 + \cos\hat{\theta}_2\cos\theta_3\sin\theta_4) \\
r_{22} &= \cos\hat{\theta}_2\cos\theta_4 - \cos\theta_3\sin\hat{\theta}_2\sin\theta_4 \\
r_{23} &= \cos\theta_4\sin\theta_1\sin\hat{\theta}_2 + (\cos\hat{\theta}_2\cos\theta_3\sin\theta_1 - \cos\theta_1\sin\theta_3)\sin\theta_4 \\
r_{24} &= L2\cos\theta_4 + L1\cos\theta_3\sin\theta_4 \\
r_{31} &= \cos\theta_3\sin\theta_1 - \cos\theta_1\cos\hat{\theta}_2\sin\theta_3 \\
r_{32} &= -\sin\hat{\theta}_2\sin\theta_3 \\
r_{33} &= \cos\theta_1\cos\theta_3 + \cos\hat{\theta}_2\sin\theta_1\sin\theta_3 \\
r_{34} &= L1\sin\theta_3
\end{aligned}$$

Donde $\hat{\theta}_2$ es el parámetro θ Denavit-Hartenberg para la segunda junta. El movimiento del hombro en cabeceo está representado en la ecuación 2.7, (Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (, 2014)

$$\theta_3 = \begin{cases} \arcsin\left(\frac{D''_{3,4}}{L1}\right) \\ \pi - \arcsin\left(\frac{D''_{3,4}}{L1}\right) \end{cases} \quad \text{Ec. (2.10)}$$

La ecuación 2.8 hombro en alabeo. (Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (, 2014)

$$\begin{aligned}
D''_{14} &= -L1\cos\theta_3\cos\theta_4 + L2\sin\theta_4 \\
\sin\theta_4 &= \frac{D''_{14} + L1\cos\theta_3\cos\theta_4}{L2} \\
D''_{24} &= L2\cos\theta_4 + L1\cos\theta_3\sin\theta_4 \\
D''_{24} &= L2\cos\theta_4 + L1\cos\theta_3\frac{D''_{14} + L1\cos\theta_3\cos\theta_4}{L2} \\
\cos\theta_4(L2^2 + L1^2\cos^2\theta_3) &= L2D''_{24} - L1D''_{14}\cos\theta_3 \\
\theta_4 &= \pm \arccos\left(\frac{L2D''_{24} - L1D''_{14}\cos\theta_3}{L2^2 + L1^2\cos^2\theta_3}\right) \quad \text{Ec. (2.11)}
\end{aligned}$$

Para el desplazamiento de la muñeca, se asigna la ecuación 2.9 (Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (, 2014)

$$D''' = D'(D_2^3)^{-1}(D_3^4)^{-1}$$

$$\hat{\theta}_2 = \arctan\left(\frac{D'''_{2,1}}{D'''_{2,2}}\right)$$

$$\theta_2 = \hat{\theta}_2 - \frac{\pi}{2}$$

$$\theta_2 = \arctan\left(\frac{D'''_{2,1}}{D'''_{2,2}}\right) - \frac{\pi}{2} \quad \text{Ec. (2.12)}$$

Los movimientos que puede realizar el codo son en guiñada y alabeo, descritos en las ecuaciones 2.10 y 2.11 respectivamente. (Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (, 2014)

$$\theta_1 = \arctan\left(\frac{D'''_{1,3}}{D'''_{3,3}}\right) \quad \text{Ec. (2.13)}$$

$$\theta_1 = \arctan\left(\frac{D'''_{3,1}}{D'''_{3,3}}\right) \quad \text{Ec. (2.14)}$$

Brazo Derecho: El brazo derecho e izquierdo son simétricos, por lo tanto las ecuaciones para el brazo derecho solo cambian en las distancias a lo largo del eje y, y en las juntas “roll” (RShoulderRoll, RElbowRoll).

La diferencia es que ElbowOffsetY ahora es negativo, entonces L1=-ElbowOffsetY.

Las ecuaciones para la cinemática e inversa del brazo derecho son:

$$D' = (A_{Base}^0)^{-1}D(A_4^{End})^{-1}\left(R_z\left(\frac{\pi}{2}\right)\right)^{-1} \quad \text{Ec. (2.15)}$$

$$D'' = (D')^{-1} \quad \text{Ec. (2.16)}$$

$$\theta_3 = \begin{cases} \arcsin\left(\frac{D''_{3,4}}{L1}\right) \\ \pi - \arcsin\left(\frac{D''_{3,4}}{L1}\right) \end{cases} \quad \text{Ec. (2.17)}$$

$$\theta_4 = \pm \arccos\left(\frac{L2D''_{2,4} - L1D''_{1,4} \cos\theta_3}{L2^2 + L1^2 \cos^2\theta_3}\right) \quad \text{Ec. (2.18)}$$

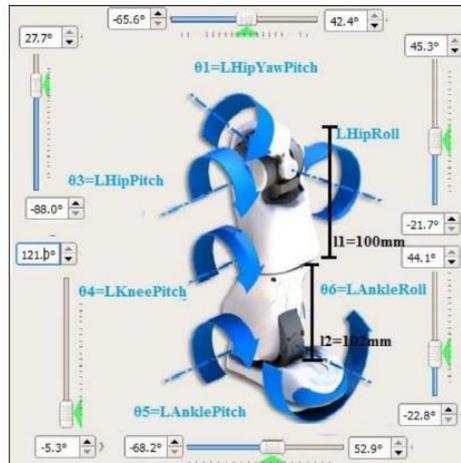
$$D''' = D'(D_2^3)^{-1}(D_3^4)^{-1} \quad \text{Ec. (2.19)}$$

$$\theta_2 = \arctan\left(\frac{D'''_{2,1}}{D'''_{2,2}}\right) - \frac{\pi}{2} \quad \text{Ec. (2.20)}$$

$$\theta_1 = \arctan\left(\frac{D'''_{1,3}}{D'''_{3,3}}\right) \quad \text{Ec. (2.21)}$$

Pierna Izquierda: En la figura 2.10, se observan los siguientes movimientos: θ_1 : Yaw/ Pitch de la cadera, θ_2 : Roll de la cadera, θ_3 : Pitch de la cadera, θ_4 : Pitch de la rodilla, θ_5 : Pitch del tobillo y θ_6 : Roll del tobillo. (Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (, 2014)

Figura 2.10. Grados de libertad de la pierna



Rango de movimiento para pierna derecha.

Fuente: (Robotics, 2014)

A partir de estos movimientos se puede construir la ecuación para un sistema no lineal:

$$D = A_{Base}^0 D_0^1 D_1^2 D_2^3 D_3^4 D_4^5 D_5^6 R_z(\pi) R_y\left(-\frac{\pi}{2}\right) A_6^{End} \quad \text{Ec. (2.22)}$$

$$\hat{D} = (A_{Base}^0)^{-1} D (A_6^{Final})^{-1} \quad \text{Ec. (2.23)}$$

$$\tilde{D} = R_x\left(\frac{\pi}{4}\right) \hat{D} \quad \text{Ec. (2.24)}$$

$$D' = (\tilde{D})^{-1} \quad \text{Ec. (2.25)}$$

Resolviendo se obtiene las ecuaciones de traslación del sistema:

$$\begin{aligned} D'_{(1,4)} &= L2 \sin \theta_5 - L1 \sin(\theta_4 + \theta_5) \\ D'_{(2,4)} &= (L2 \cos \theta_5 + L1 \cos(\theta_4 + \theta_5)) \sin \theta_6 \\ D'_{(3,4)} &= (L2 \cos \theta_5 + L1 \cos(\theta_4 + \theta_5)) \cos \theta_6 \end{aligned}$$

En las ecuaciones anteriores podemos observar las variables L1 y L2 que representan la longitud del muslo y la tibia respectivamente. Para calcular la distancia del triángulo formado desde la base de la pierna hasta el final de la misma utilizaremos la siguiente ecuación.

$$d = \sqrt{(s_x - p'_x)^2 + (s_y - p'_y)^2 + (s_z - p'_z)^2} \quad \text{Ec. (2.26)}$$

Donde $(s_x, s_y, s_z) = (0,0,0)$ es el nuevo origen y $(p'_x, p'_y, p'_z) = (D'_{(1,4)}, D'_{(2,4)}, D'_{(3,4)})$ es la posición del nuevo punto final. Usando la ley de cosenos encontramos el ángulo interno entre el muslo y la tibia:

$$\theta'_4 = \arccos\left(\frac{L1^2 + L2^2 - d^2}{2L1L2}\right)$$

$$\theta''_4 = \pi - \theta'_4$$

$$\theta_4 = \pm \theta''_4$$

$$\theta_4 = \pm \left(\pi - \cos^{-1} \left(\frac{L1^2 + L2^2 - \|\bar{0} - \bar{p}\|^2}{2L1L2} \right) \right) \quad \text{Ec. (2.27)}$$

Para encontrar el ángulo θ_6 usamos las ecuaciones de traslación $D'_{(2,4)}$ y $D'_{(3,4)}$:

$$\begin{aligned} \frac{D'_{(2,4)}}{D'_{(2,4)}} &= \frac{p'_y}{p'_z} \\ \frac{(L2\cos\theta_5 + L1\cos(\theta_4 + \theta_5))\sin\theta_6}{(L2\cos\theta_5 + L1\cos(\theta_4 + \theta_6))\cos\theta_6} &= \frac{p'_y}{p'_z} \\ \theta_6 &= \arctan\left(\frac{p'_y}{p'_z}\right) \quad \text{si } ((L2\cos\theta_5 + L1\cos(\theta_4 + \theta_5))) \neq 0 \quad \text{Ec. (2.28)} \end{aligned}$$

Ahora procedemos a calcular el ángulo del tobillo θ_5 :

$$\begin{aligned} \widetilde{D}' &= \widetilde{D} \left(D_5^z R_z(\pi) R_y\left(-\frac{\pi}{2}\right) \right)^{-1} \\ D'' &= (\widetilde{D}')^{-1} \end{aligned}$$

Usando la nueva posición final $(p''_x, p''_y, p''_z) = (D''_{(1,4)}, D''_{(2,4)}, D''_{(3,4)})$, las nuevas ecuaciones de traslación son:

$$\begin{aligned} D''_{(1,4)} &= L2 \cos \theta_5 + L1(\cos \theta_5 \cos \theta_4 - \sin \theta_5 \sin \theta_4) \\ D''_{(2,4)} &= -L2 \sin \theta_5 - L1(\sin \theta_5 \cos \theta_4 + \cos \theta_5 \sin \theta_4) \\ D''_{(3,4)} &= 0 \end{aligned}$$

Teniendo en cuenta que:

$$\begin{aligned} D''_{(1,4)} &= p''_x \\ (L2 + L1\cos\theta_4)\cos\theta_5 &= p''_x + L1\sin\theta_5\sin\theta_4 \\ \cos\theta_5 &= \frac{p''_x + L1\sin\theta_5\sin\theta_4}{L2 + L1\cos\theta_4} \quad \text{si } L2 + L1\cos\theta_4 \neq 0 \end{aligned}$$

$$\sin\theta_5(-L2 - L1\cos\theta_4) - L1\cos\theta_5\sin\theta_4 = p''_y$$

$$\begin{aligned}
& \sin\theta_5(-L2 - L1\cos\theta_4) - L1 \frac{p''_x + L1\sin\theta_5\sin\theta_4}{L2 + L1\cos\theta_4} \sin\theta_4 = p''_y \\
& -\sin\theta_5(L2 + L1\cos\theta_4) - \frac{L1p''_x\sin\theta_4}{L2 + L1\cos\theta_4} - \frac{L1^2\sin\theta_5\sin^2\theta_4}{L2 + L1\cos\theta_4} = p''_y \\
& -\sin\theta_5(L2 + L1\cos\theta_4)^2 - L1^2\sin\theta_5\sin^2\theta_4 = p''_y(L2 + L1\cos\theta_4) + L1p''_x\sin\theta_4 \\
& \theta_5 = \arcsin\left(-\frac{p''_y(L2 + L1\cos\theta_4) + L1p''_x\sin\theta_4}{L1^2\sin^2\theta_4 + (L2 + L1\cos\theta_4)^2}\right) \quad \text{Ec. (2.29)}
\end{aligned}$$

La división es siempre factible, porque $L1^2\sin^2\theta_4 + (L2 + L1\cos\theta_4)^2$ es mayor que cero para cualquier valor de θ_4 .

Finalmente procedemos a deducir las ecuaciones para los ángulos θ_3 , θ_2 y θ_1 a partir de las ecuaciones de traslación obtenidas de D''' .

$$D''' = \widetilde{D}'(D_3^4 D_4^5)^{-1} \quad \text{Ec. (2.30)}$$

A continuación se describen las ecuaciones de traslación obtenidas de D''' .

$$\begin{aligned}
r_{11} &= \cos\hat{\theta}_1\cos\hat{\theta}_2\cos\theta_4 - \sin\hat{\theta}_1\sin\theta_3 \\
r_{12} &= -\cos\theta_3\sin\hat{\theta}_1 - \cos\hat{\theta}_1\cos\hat{\theta}_2\sin\theta_3 \\
r_{13} &= \cos\hat{\theta}_1\sin\hat{\theta}_2 \\
r_{21} &= -\cos\theta_3\sin\hat{\theta}_2 \\
r_{22} &= \sin\hat{\theta}_2\sin\theta_3 \\
r_{23} &= \cos\hat{\theta}_2 \\
r_{31} &= -\cos\hat{\theta}_2\cos\theta_3\sin\hat{\theta}_1 - \cos\hat{\theta}_1\sin\theta_3 \\
r_{32} &= -\cos\hat{\theta}_1\cos\theta_3 + \cos\hat{\theta}_2\sin\hat{\theta}_1\sin\theta_3 \\
r_{33} &= -\sin\hat{\theta}_1\sin\hat{\theta}_2
\end{aligned}$$

$$\hat{\theta}_2 = \arccos D'''_{(2,3)}$$

$$\begin{aligned}
\theta_2 &= \hat{\theta}_2 - \frac{\pi}{4} \\
\theta_2 &= \arccos D'''_{(2,3)} - \frac{\pi}{4} \quad \text{Ec. (2.31)}
\end{aligned}$$

$$\theta_3 = \arcsin\left(\frac{D'''_{(2,2)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)}\right) \quad \text{Ec. (2.32)}$$

$$\hat{\theta}_1 = \arccos\left(\frac{D'''_{(1,3)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)}\right)$$

$$\theta_1 = \hat{\theta}_1 + \frac{\pi}{2}$$

$$\theta_1 = \arccos\left(\frac{D'''_{(1,3)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)}\right) + \frac{\pi}{2} \quad \text{Ec. (2.33)}$$

Pierna Derecha: Teniendo en cuenta que los movimientos de la pierna izquierda son iguales en la pierna derecha, tenemos que:

$$D_{base}^{piernaIzq}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = MD_{base}^{piernaDer}(\theta_1, -\theta_2, \theta_3, \theta_4, \theta_5, -\theta_6)M$$

Finalmente se muestran las ecuaciones para los movimientos de la pierna derecha.

$$\theta_1 = \cos^{-1}\left(\frac{D'''_{(1,3)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)}\right) + \frac{\pi}{2} \quad \text{Ec. (2.34)}$$

$$\theta_2 = \arccos D'''_{(2,3)} - \frac{\pi}{4} \quad \text{Ec. (2.35)}$$

$$\theta_3 = \arcsin\left(\frac{D'''_{(2,2)}}{\sin\left(\theta_2 + \frac{\pi}{4}\right)}\right) \quad \text{Ec. (2.36)}$$

$$\theta_4 = \pm \left(\pi - \cos^{-1}\left(\frac{L1^2 + L2^2 - \|\bar{0} - \bar{p}\|^2}{2L1L2}\right) \right) \quad \text{Ec. (2.37)}$$

$$\theta_5 = \arcsin\left(-\frac{p''_y(L2 + L1\cos\theta_4) + L1p''_x\sin\theta_4}{L1^2\sin^2\theta_4 + (L2 + L1\cos\theta_4)^2}\right) \quad \text{Ec. (2.38)}$$

$$\theta_6 = \begin{cases} \arctan\left(\frac{D'_{(2,4)}}{D'_{(3,4)}}\right) & \text{si } ((L2\cos\theta_5 + L1\cos(\theta_4 + \theta_5))) \neq 0 \end{cases} \quad \text{Ec. (2.39)}$$

2.3 Arduino

Arduino es una plataforma de código abierto que consta de entradas y salidas tanto analógicas como digitales. (Arduino, 2015)

Esta placa trabaja con microcontroladores brindando ventajas en relación a otros sistemas como:

- **Accesibilidad:** en relación a otros sistemas que utilizan microcontroladores, Arduino es más económico, haciéndolo más accesible al público.
- **Compatibilidad:** en comparación a la mayoría de los sistemas con microcontroladores que sólo se pueden ejecutar en el sistema operativo Windows, Arduino puede ejecutarse también en Macintosh OSX y GNU/Linux.
- **Flexibilidad:** el entorno de programación está diseñado para que los usuarios principiantes puedan utilizarlo sin ningún problema y los usuarios más experimentados puedan implementar sus conocimientos de programación libremente.
- **Software expandible:** la plataforma Arduino al ser diseñada con un software de código abierto, puede ser expandida a través de librerías C++ por programadores más experimentados.
- **Constante evolución:** los esquemas de la placa se encuentran publicados bajo una licencia común; es decir que cualquier diseñador experimentado puede crear su propia versión de Arduino y mejorarlo. Asimismo diseñadores con un nivel de conocimiento menor pueden construir una versión de la placa más simple y económica. Adicional a esto Arduino funciona con microcontroladores ATMEGA8 y ATMEGA168 los cuales son de fácil acceso a los usuarios. (CC, 2015)

2.4 Comparación entre placas Arduino

Tabla 2.2. Características de placas Arduino.

Arduino	Microcontrolador	Voltaje	Memoria	Pines
Pro Mini	ATmega168 o Atmega328, versiones de 3.3V y 8Mhz	5v	32KB y 16KB de flash	14 pines digitales y 6 pines analógicos
Arduino Uno	ATmega320 de 8 bits a 16Mhz	5v	32KB flash 2KB de SRAM y 1KB de EEPROM.	14 pines digitales y 6 pines analógicos
Arduino Mega	ATmega2560 de 8 bits a 16Mhz	5v	SRAM de 8KB, 4KB de EEPROM y 256KB de flash	54 pines digitales y 16 pines analógicos

Tabla con las características básicas del Arduino Pro Mini, Uno y Mega.

2.5 Arduino USB Host Shield

El Arduino USB Host Shield permite conectar dispositivos USB a su tarjeta Arduino. USB Host Shield está basado en un MAX3421E, que es un controlador USB que contiene la lógica digital y los circuitos analógicos necesarios para implementar un periférico USB.

Las siguientes clases de dispositivos son compatibles con USB Host Shield:

- Dispositivos de interfaz humana tales como joysticks, tableta gráfica, teclados, mouse, etc.
- Controladores de Xbox360. Nintendo Wii y Sony PS3.
- FTDI, PL-2303, ACM, receptores GPS y algunos celulares.
- Dispositivos con sistema operativo Android.

- Dispositivos digitales: cámaras Canon, Nikon DSLR y P & S, Powershot y PTP genérico.
- Dispositivos de almacenamiento masivo: USB, lectores de tarjetas de memoria, discos duros externos, etc. (arduino., 2016)

Tabla 2.3. Características USB Host Shield.

Tensión de funcionamiento	5V
Controlador USB	MAX3421E
Corriente máxima	400mA

Tabla con las características básicas del USB Host Shield.

Fuente: (Arduino, 2016)

2.6 Cámara Robot NAO

El robot NAO posee 2 cámaras de 1,22 megapíxeles cuyas características técnicas se encuentran detalladas en la tabla 3.1. Esta cámara tiene una resolución muy pequeña para poder detectar un código de barras; adicionalmente la librería ALBarcodeReader propia del robot NAO solo puede leer códigos QR, por este motivo se planteó adicionarle una cámara externa, pero para que las imágenes sean procesadas se requiere una computadora y al utilizarla el robot deja de ser autónomo; por estas razones se tomó la decisión de realizar la lectura del código de barras a través de una pistola lectora de códigos. (Aldebaran)

Tabla 2.4. Características de las cámaras del robot NAO.

Cámara	Modelo	MT9M114
	Tipo	SOC Sensor de imagen
Imagen	Resolución	1.22 Mp
	Formato óptico	1/6 inch
	Píxeles activos	1288x968
Salida	Salida de la cámara	1280*960@30fps
	Formato de salida	(YUV422 color space)

Tabla con las características básicas de las cámaras del Robot NAO.

Fuente: (Aldebaran)

2.7 Código de barras

El código de barras es una técnica que permite clasificar diferentes productos a través de líneas negras intercaladas entre espacios en blanco, estas líneas son de diferentes espesores, los cuales se miden en mils (El mil es la mínima unidad de medida del sistema inglés, es la milésima parte de una pulgada. Equivale a 0,0254 milímetros), además, estos códigos son muy eficientes y tienen una gran precisión.

El lector de código de barras, funciona irradiando un haz de luz hacia el código, dentro del lector se encuentra un sensor que percibe la luz irradiada y la transforma en una señal eléctrica, esta es descifrada como un número entero. (Codigos, 2013)

El proceso de leer un código de barras se divide en tres pasos.

1. Dispositivo de Entrada
2. Decodificador
3. Lector de código de barras

Primero, un dispositivo de entrada debe convertir la luz en energía eléctrica. Existe una diversidad de dispositivos de entrada.

2.7.1 Dispositivos de Entrada

A continuación, se detalla los tipos de dispositivos de Entrada que son: Lápiz, Charge Couple Device y el lector.

El Lápiz, es el dispositivo más económico y fácil de usar, pero presenta algunos inconvenientes como son que se debe colocarlo encima del código de barras, produciendo el deterioro del mismo adicionalmente presenta inconvenientes en la movilidad y en el ángulo ya que deben ser precisos al momento de utilizarlo. (Codigos, 2013)

El Charge Couple Device es otro sistema de entrada que permite leer eficientemente los códigos de barras, pero presenta algunas restricciones como son el pequeño espacio en el cual se puede detectar el código ya que solo detecta el código cuando existe una

separación de entre 1 a 3 pulgadas, adicionalmente no puede leer códigos de barras con grosores muy amplios ya que presenta un haz de luz pequeño.

El lector es el dispositivo con mayor grado de confiabilidad y el más utilizado ya que puede realizar la detección del código de barras a una amplia distancia como es de entre 6 a 30 pulgadas a una distancia de 30 pies. Por estas características se decidió utilizar este tipo de lector de código de barras en el proyecto técnico. (Codigos, 2013)

2.7.2 Decodificadores

Luego de pasar por el dispositivo de entrada se procede a decodificarlo, en el cual se transforman los pulsos eléctricos en un número entero este dispositivo realiza dos acciones las cuales son:

- Determina entre los diferentes códigos de barras.
- Transformar los pulsos eléctricos en números enteros.
- Convertir los números en vectores.

Los decodificadores suelen estar colocados internamente o externamente del dispositivo de entrada, pero cuando se lo tiene colocado internamente su costo es inferior, pero tiene la desventaja de ser menos eficaz, y en el caso contrario cuando se lo coloca en la parte exterior su costo es superior, pero tiene una mayor eficacia.

Hay códigos de barras de 8, 12, 13 y 14 dígitos, esto depende del tipo de producto, por ejemplo, los productos de peso fijo, variable y confecciones y calzado tienen 13 dígitos.

Productos de tamaño muy reducido solicitarán un código más pequeño al tradicional y después de realizar un estudio se asignará el código de 8 dígitos. (Codigos, 2013)

El código de barras de 13 dígitos tiene las siguientes características:

Figura 2.11. Código de barras



Significado de cada número del código de barras.

Fuente: (Codigos, 2013)

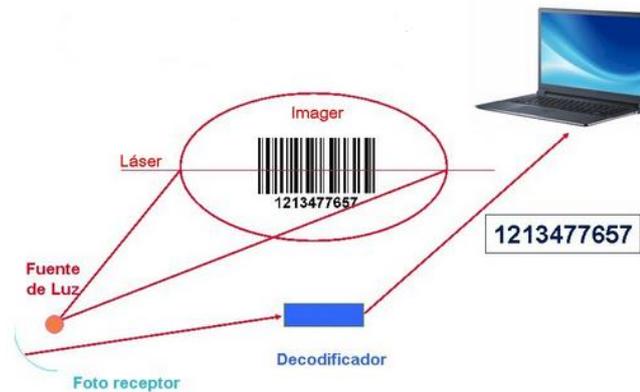
- Los tres primeros dígitos, indican el país de procedencia.
- Los siguientes 4 dígitos el código de la empresa fabricante.
- Los siguientes 5 dígitos el código del producto.
- El último dígito es de verificación, control. Que se usa para asegurarse que el lector de barras ha leído el código bien. (Codigos, 2013)

2.8 Lector de código de barras

Los códigos de barras se leen con lectores especializados que proyectan una fuente de luz encima del código impreso. La fuente luminosa puede ser de tres tipos:

- Tecnología CCD: La fuente de iluminación es generada por leds luminosos de bajo consumo. Esta tecnología es utilizada para leer códigos de una Dimensión 1D.
- Tecnología de rayo láser visible: La fuente de iluminación es un solo rayo visible (necesidad de orientar el código) o varios rayos láser visible (sólo se necesita presentar el código al lector) que cruzan todas las barras del código de barras. Esta tecnología permite la lectura de códigos de barras de una Dimensión 1D.
- Tecnología Imager: La fuente de iluminación es un haz de luz brillante que cubre completamente el código a ser leído por lo que no es necesario orientar el código. Esta tecnología permite la lectura de códigos de barras de una y dos dimensiones (1D y 2D).

Figura 2.12. Funcionamiento



Funcionamiento del lector de código de barras.

Fuente: (Codigodebarras.com, 2016)

Los lectores irradian un haz de luz, este haz no es reflejada por las secciones negras y si es reflejada por las secciones blancas del código de barras. Este haz de luz es detectado por una foto receptor y transformarlos en pulso eléctricos. Estos se decodifican y se transforman en un numero entero. Los lectores funcionan en base a rayos láseres. Para que se realice la lectura de un código el haz de luz debe atravesar todo el código de barras.(Codigodebarras.com, 2016)

2.9 Pistola Marca One

Es un scanner lector de código de barras que posee las siguientes características:

- Fuente de luz 650 (laser visible)
- Modelo escaneado en una sola línea bidireccional
- Velocidad de escaneo 200 lineas/seg
- Resolución de 3.3 mil
- Ancho de escaneado 30mm
- Profundidad de exploración 0-250mm
- Esquina 30
- Elevación 65
- Capacidad de decodificación Upc/ean, code 128, ucc/ean128,code39, code 30 full ascii, codbar, interleaved 2 de 5, code 93, msi, code11, isbn, issn, código postal de china, código de barra unidimensional. (Labs, 2016)

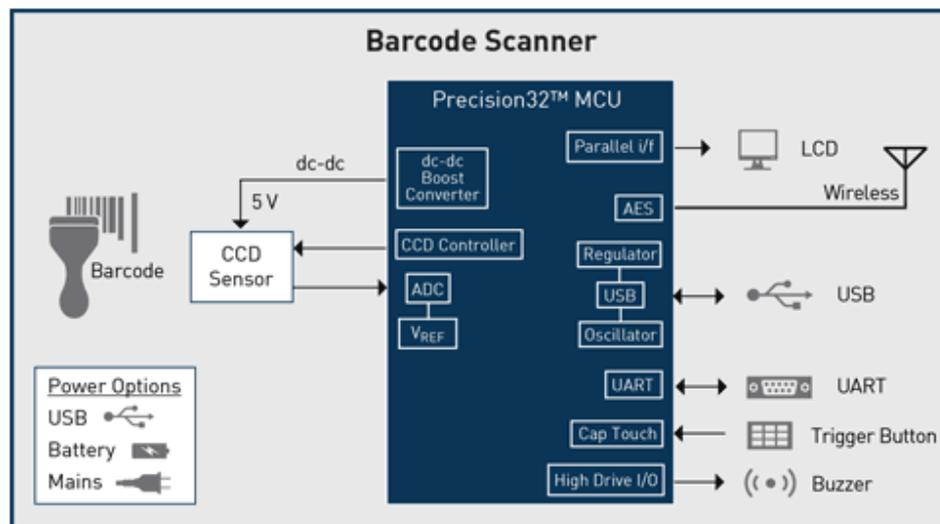
Figura 2.13. Lector de código de barras



Pistola marca One de 32B.

Fuente: (Labs, 2016)

Figura 2.14. Sistema de escaneo de código de barras



Sistema de escaneo de código de barras basado en un microcontrolador altamente integrado de 32 bits.

Fuente: (Labs, 2016)

2.10 Reconocimiento de rostros

ALFaceDetection es un módulo de visión integrado a Choregraphe que sirve para que el robot NAO detecte y reconozca rostros humanos, adicionalmente puede buscar al rostro caminando hasta colocarse en frente de la persona. Primero debe aprender los rostros y esta etapa de aprendizaje se puede realizar a través del método LearnFace de la API o a través del bloque en Choregraphe LearnFace.

Una vez que haya hecho clic en el bloque e ingresado el nombre de la persona, esta persona tiene 5 segundos para colocar su cara o mostrar una foto delante del robot NAO.

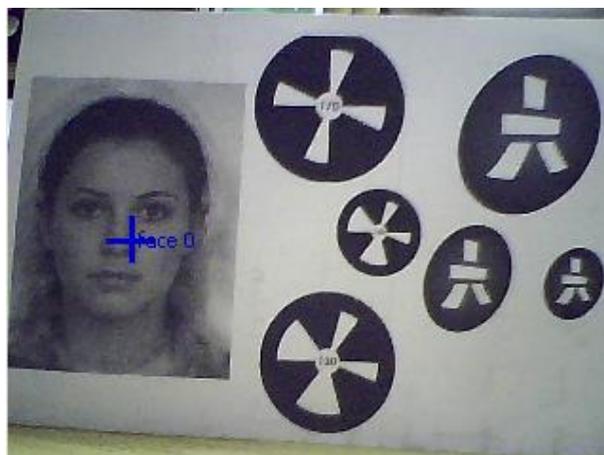
Entonces se pone en marcha el proceso de aprendizaje mientras los ojos de NAO están azules. Sus ojos se tornan verde en menos de un segundo si la cara es vista por NAO en condiciones adecuadas (por ejemplo, ninguna sombra parcial en la cara, sin luz de fondo, la persona no está demasiado lejos).

Si los ojos siguen azules después de algunos segundos, la persona debe moverse con el fin de cambiar las condiciones de aprendizaje.

Una vez que el robot aprende el rostro de la persona ya la tiene almacenada en su memoria y se puede utilizar la información cuando se desee.

Para detectar las caras se utiliza la casilla Choregraphe Walk Tracker o WB Tracker y cambiar el valor predeterminado de Red Ball a Face. El robot NAO se moverá hacia la persona manteniendo siempre la cara en medio de su campo de visión. (Aldebaran, s.f.)

Figura 2.15. Reconocimiento de rostros



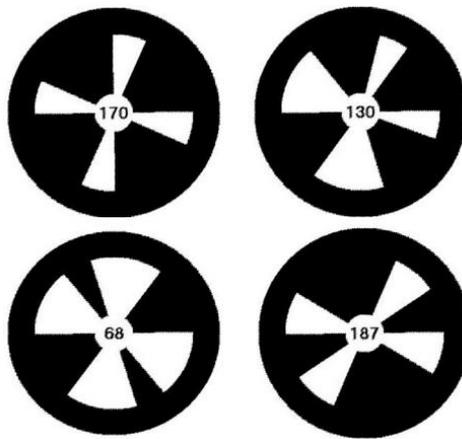
Reconocimiento de rostro con ALFaceDetection.

Fuente: (Aldebaran, s.f.)

2.11 NAOmarks

NAOmarks es un conjunto de figuras circulares de color negro con zonas blancas en su interior, cada una de ellas son reconocidas por el robot NAO como un número entero el cual puede ser asociado a su vez con acciones del robot, como por ejemplo girar a la derecha, girar a la izquierda, detenerse, girar la cabeza, mover sus manos, etc. En nuestro caso la NAOmarks 68 es utilizada para girar a la izquierda y 130 para indicarle que ha llegado al lugar donde va a esperar la hora a la que debe buscar y entregar el medicamento. En la figura 2.16 se pueden visualizar los NAOmarks utilizados en el proyecto.

Figura 2.16. Distintos patrones de NAOmarks



Distintos NAOmarks.

Fuente: (Aldebaran, s.f.)

CAPÍTULO 3 DISEÑO E IMPLEMENTACIÓN

3.1 Diagrama de interconexión

La Interfaz de conexión de todo el proceso se muestra en la figura 3.1

Figura 3.1. Diagrama de la interfaz de conexión



Diagrama de la interfaz de conexión entre la pistola lectora de código de barras y el robot NAO.

3.2 Transmisión del código de barras desde Arduino al Robot NAO

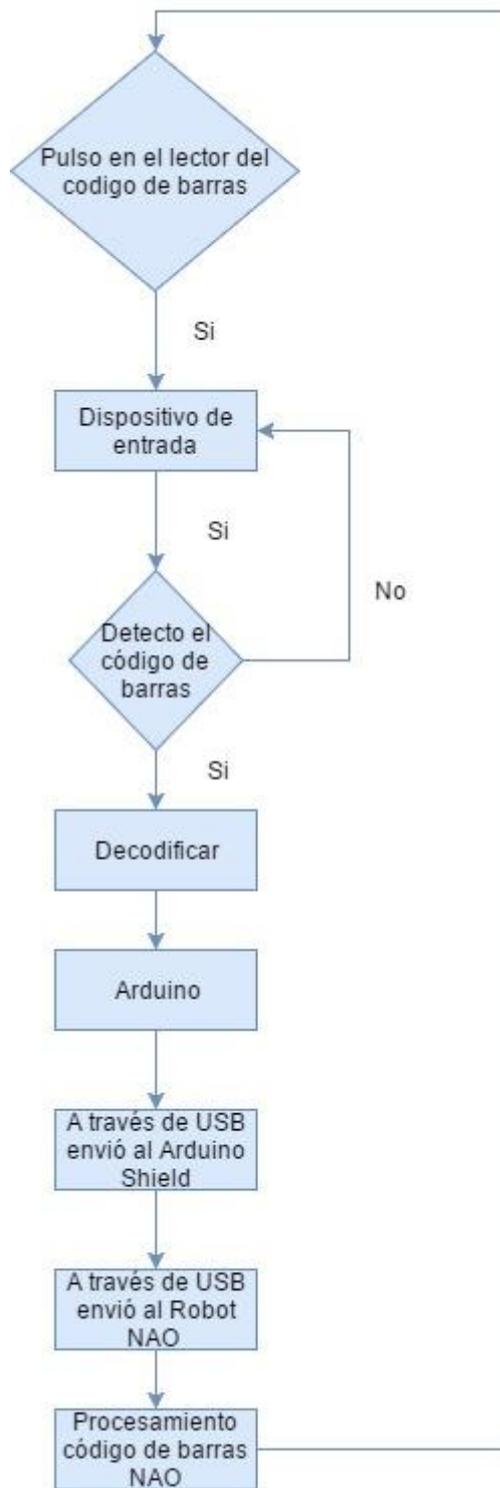
En los siguientes diagramas de flujo se muestra cómo se realiza la transmisión del código de barras desde la pistola hasta el robot NAO.

Figura 3.2. Diagrama de Flujo



Envío de información desde el robot NAO al Arduino.

Figura 3.3. Diagrama de Flujo



Recepción del código de barras al Robot NAO.

Utilizando la librería propia del USB Host Shield, la cual permite conectar un dispositivo USB externo en el Arduino, se envía los datos desde el lector hacia el

Arduino mediante el código que se puede observar en la Figura 3.4. El código base se obtuvo de la página web <http://www.electroingenio.com/arduino/bar-code-scanner-arduino-usb-shield/> el cual se modificó y cambió para poder enviar y recibir datos.

Figura 3.4. Conexión Arduino USB Host Shield - Lector de código de barras

```
HIDBoot<USB_HID_PROTOCOL_KEYBOARD> Keyboard(&Usb);
//HIDBoot<HID_PROTOCOL_KEYBOARD> Keyboard(&usb);

class KbdRptParser : public KeyboardReportParser
{
    void PrintKey(uint8_t mod, uint8_t key);
protected:
    virtual void OnKeyDown (uint8_t mod, uint8_t key);
    virtual void OnKeyPressed(uint8_t key);
};

void KbdRptParser::OnKeyDown(uint8_t mod, uint8_t key)
{
    uint8_t c = OemToAscii(mod, key);

    if (c)
        OnKeyPressed(c);
}

void KbdRptParser::OnKeyPressed(uint8_t key)
{

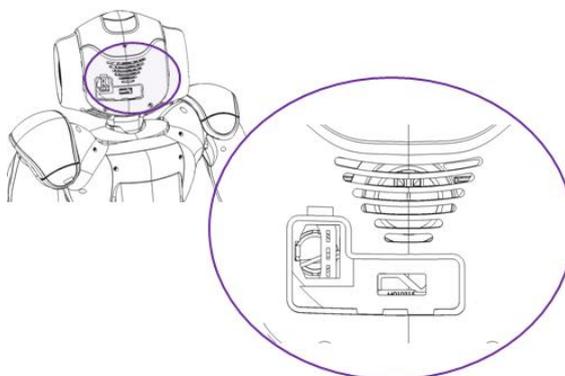
```

Código para conectar un dispositivo USB externo en el Arduino.

Fuente: Electroingenio.

Se envía el valor del código de barras hacia el robot NAO mediante el puerto USB, propio del Arduino, hacia el puerto USB del robot NAO se puede observar donde está ubicado físicamente en la figura 3.5.

Figura 3.5. Puerto USB Robot NAO



Ubicación del USB del robot NAO,

Fuente: (Aldebaran)

A continuación, se procedió a implementar el código que permita enviar el código de barras por el puerto USB, y debido a que se requiere que este proceso sea rápido se escogió una velocidad de transmisión de 115200 kbps como se muestra en la figura 3.6.

Figura 3.6. Código de barras enviado desde el Arduino hacia el robot NAO

```

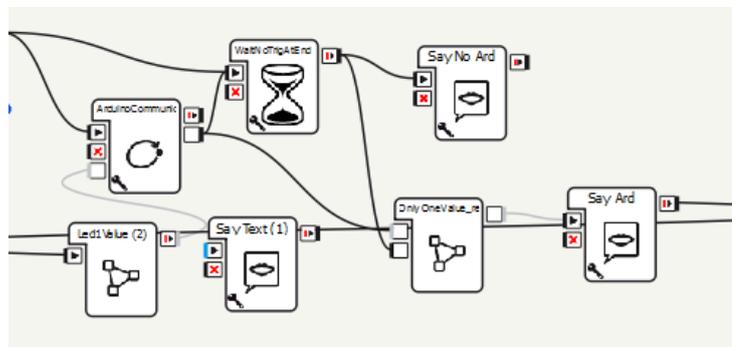
void KbdRptParser::OnKeyPressed(uint8_t key)
{
    codigo[cont]= (char) key ;
    if (cont1<=12){
        Serial.print( "\\\"Button1\\\", " );
        Serial.print(codigo);
        Serial.println( "]" );
        delay(200);
        cont1++;
    }
    else {
        cont1=0;
    }
}

```

Código para transmitir el código de barras desde el Arduino hacia el robot NAO.

El código de barras es enviado al robot NAO mediante la programación en Choregraphe la cual se muestra en la figura 3.7 donde Arduinocommunication es el bloque que primero comprueba si tiene conectado un Arduino, si lo está, el robot NAO dice “conectado” caso contrario dice “no conectado”, Arduinocommunication adicionalmente recibe el código de barras desde el Arduino y lo guarda en la memoria del robot.

Figura 3.7. Interconexión Robot NAO Arduino en Choregraphe



Bloques principales para la interconexión del Robot NAO con Arduino.

El bloque principal que interconecta el Arduino con el Robot NAO Arduinocommunication fue realizado en Python y se muestra en la figura 3.8; lo que se hace es obtener los datos a través del comando `serial.getData()` e ir guardando el código de barras en la memoria el robot mediante el comando `ALMemory.raiseMicroEvent`.

El código base se obtuvo de la página web http://www.mangedisque.com/Alma/NAOStuff/arduino_usb_NAO_tutorial.zip el cual se modificó para enviar un vector de datos al robot y recibir un dato desde el robot.

Figura 3.8. Interconexión Robot NAO Arduino en Python

```
try:
    self.serial.open();
except BaseException, err:
    self.log( "ERR: got error: %s" % str( err ) );

rPeriod = self.getParameter( 'rPeriod' );
while( not self.bMustStop ):
    s = self.serial.getData();
    if( s != False):
        strVarName = "Arduino/" + s[0];
        values = s[1];
        if( s[0] == "Ping" ):
            self.timeLastPing = time.time();
            self.output_ping();
            ALMemory.raiseMicroEvent( strVarName, values );
        time.sleep( rPeriod );
self.serial.close();
self.bIsRunning = False;
self.onStopped();
self.log( self.boxName + ": start - end" );
```

Lectura del código de barras del puerto USB y guardado en la memoria del Robot NAO.

El código de barras es almacenado en la memoria del robot número a número lo cual dificulta el manejo de esta información, por este motivo se creó un bloque en Python que integre los doce números en un solo vector, el cual es enviado hacia la salida del bloque `onStopped` como se puede observar en la figura 3.9.

Figura 3.9. Integración de datos en un solo vector.

```
def onInput_onStart(self, p):
    if( self.cont <= 12 ):
        self.codigo[self.cont]=p
        self.cont=self.cont + 1
    if (self.cont > 12 ):
        self.onStopped(str(self.codigo))
        self.cont=0
    pass
```

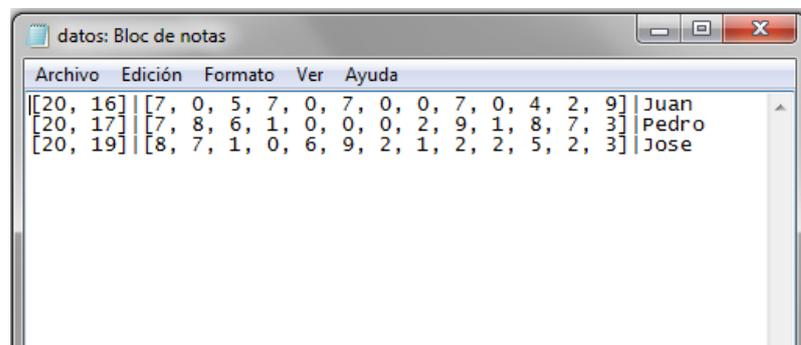
Código necesario para unir los datos del código de barras en un solo vector.

3.3 Base de datos

Se procedió a elaborar un archivo de texto en el cual se ingresó la hora, el código de barras del medicamento y el paciente como se puede observar en la Figura 3.10.

Para garantizar que el robot no dependa de una computadora para acceder a la base de datos y pueda movilizarse libremente se colocó el archivo en la carpeta behavior_1 el cual es cargado al Robot NAO al correr el programa.

Figura 3.10. Archivo con la información de los pacientes



Muestra la hora, el código de barras del medicamento y el paciente a ser administrado.

A continuación, se creó un bloque en Python en el cual se compara la hora actual, la cual esta guardada en la variable aTime con la hora a la que debe ser administrado el medicamento la cual esta guardada en la variable text[0] y de esta manera acceder al archivo anteriormente creado el cual está almacenado en la memoria del robot en y se accede a él a través del comando:

os.path.join(ALFrameManager.getBehaviorPath(self.behaviorId),"datos.txt"), donde "datos.txt" es el archivo plano que contiene la base de datos. El código necesario se muestra en la Figura 3.11.

Figura 3.11. Acceso a la información del archivo de texto

```
def onInput_onStart(self,p):
    currentTime = datetime.datetime.now()
    aTime = str([ currentTime.hour, currentTime.minute])
    self.onStopped1(aTime)
    self.a=0;
    while True:
        try:
            filepath = os.path.join(ALFrameManager.getBehaviorPath(self.behaviorId), "datos.txt")
            textfile=open(filepath)
            for text_success in textfile:
                text = text_success.split('|')
                time.sleep(2)
                if(aTime==text[0]):
                    self.onStopped(str(text[1]))
                    self.onStopped1(text[2])
```

Código para la comparación de la hora y el acceso al archivo con la información de los pacientes.

Una vez que se accede a la información del paciente se procede a comparar el código de barras del medicamento almacenado en la base de datos con el código que el robot detecta, para esto se creó un bloque en Python que se muestra en la figura 3.12

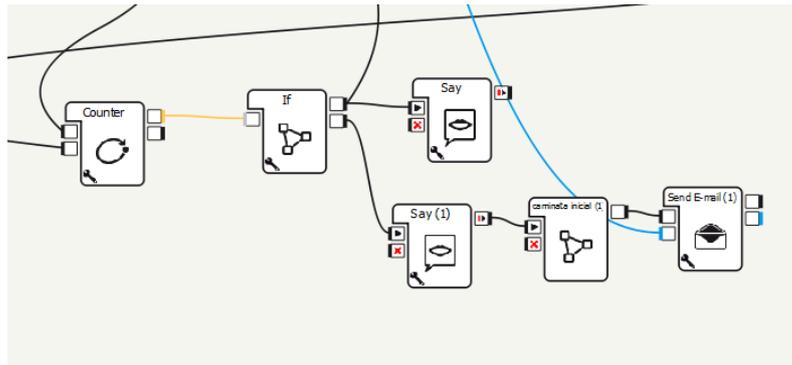
Figura 3.12. Comparación entre códigos de barras

```
def onInput_input(self,p1):
    if (p1==self.cod):
        self.onStopped()
    if (p1!=self.cod):
        self.onStopped1()
    pass
def onInput_onStart(self,p):
    self.cod=p
    pass
```

Código para la comparación del código de barras almacenado y el código detectado.

Cuando el robot no encuentra el medicamento correcto dice “No encuentro la medicación” y procede a buscarla nuevamente, esto lo realiza tres veces luego de lo cual dice “No hay la medicación” y regresa a la posición de espera en donde envía un correo al usuario informando que se terminó la medicación y con el código de barras respectivo; la programación se puede observar en la figura 3.13.

Figura 3.13. Programación para buscar 3 veces la medicación



Programación en Choregraphe para buscar tres veces la medicación que debe entregar a la hora determinada.

3.4 Activación controlada por el robot NAO a la pistola lectora de código de barras

La pistola lectora de código de barras para ser activada debe ser pulsada por una persona, para evitar este inconveniente y que sea activada por una señal proveniente del Robot NAO primero se procedió a desarmar la pistola como se puede observar en la figura 3.14 y 3.15.

Figura 3.14. Pistola lectora código de barras desarmada



Circuitería electrónica de la pistola lectora de código de barras.

Figura 3.15. Pistola lectora código de barras desarmada



Circuitería electrónica de la pistola lectora de código de barras con cinca corta.

Como se puede observar en la figura 3.15 la cinta que transmite la información entre las dos placas es muy corta para poder montarlo en el robot NAO, por lo que se decidió colocarle una más grande como se puede observar en la figura 3.16.

Figura 3.16. Pistola lectora código de barras desarmada



Circuitería electrónica de la pistola lectora de código de barras con la cinta más grande.

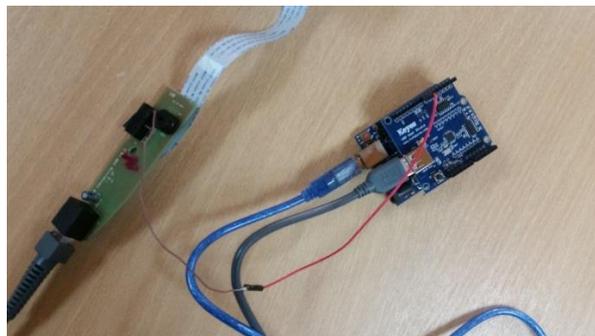
A continuación, se procedió a soldar un cable al pulsador que activa la pistola con el pin 8 del Arduino como se puede observar en la figura 3.17 y 3.18.

Figura 3.17. Cable soldado al pulsador



Cable soldado al pulsador de activación de la pistola lectora de código de barras.

Figura 3.18. Señal de activación desde el Arduino a la pistola lectora de código de barras



Cable soldado al pulsador de activación de la pistola para dar la señal de activación desde el Arduino a la pistola lectora de código de barras.

La señal de activación de la pistola proviene del Robot NAO como se puede observar en la figura 3.19, su programación en Python permite transmitir al Arduino mediante el comando `self.serial.sendData` el pulso de activación de la pistola.

Figura 3.19. Programación en Python para transmitir al Arduino

```
def onInput_onData(self, v):
    if ( time.time() - self.timeLastPing > 5. ): # don't pollute serial
detected
        return;
    if( len( v ) != 2 ):
        self.log( "ERR: data has bad formatting: %s" % str( v ) );
        return;
    self.serial.sendData( v[0], v[1] );
    pass

def onInput_onComments(self, strComments):
    if ( time.time() - self.timeLastPing > 5. ): # don't pollute serial
detected
        return;
    self.serial.sendComments( strComments );
```

Código en Python para transmitir el pulso de activación del Robot NAO al Arduino.

Se puede observar en la figura 3.20 la programación en el Arduino para dar el pulso de activación a la pistola lectora de código de barras la cual se activa en la salida 8 del Arduino y permanecerá activada por 10 segundos.

Figura 3.20. Programación en el Arduino

```
if( strcmp( pDataBegin, "Led1", nLenDataName ) == 0 && nNbrData >= 1 )
{
    Serial.print( "#Led1 info received:" );
    Serial.println( anReceivedValues[0] );
    analogWrite( ledPin, 0);
    delay(10000);
    analogWrite( ledPin, anReceivedValues[0] );
}
```

Programación en el Arduino para dar el pulso de activación a la pistola lectora de código de barras.

Si al momento de buscar el código de barras no detecta ninguno, procede a buscar nuevamente hasta encontrar alguno, la programación de este bloque se la puede observar en la figura 3.21.

Figura 3.21. Programación para repetir la búsqueda del código de barras

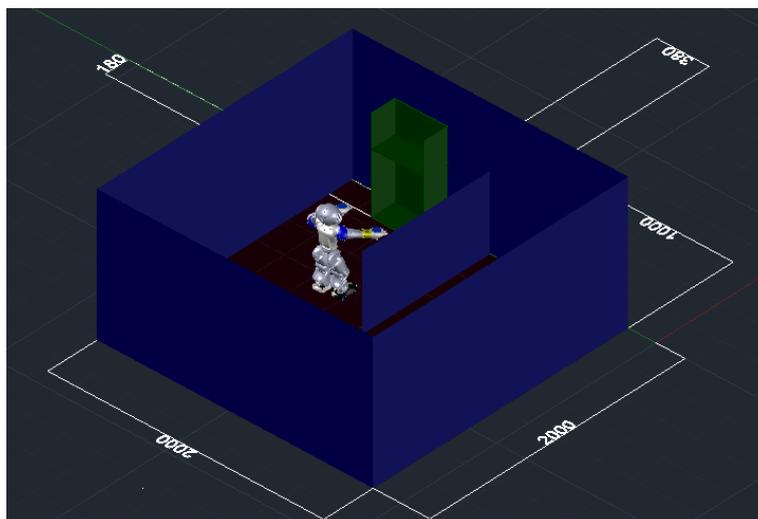
```
def onInput_input(self,p1):  
    self.cod=p1  
    pass  
def onInput_onStart(self,p):  
    if (self.cod=="0"):  
        self.onStopped()  
    if (self.cod!="0"):  
        self.onStopped1()  
    pass
```

Programación en Python para repetir la búsqueda del código de barras.

3.5 Entorno creado para el robot NAO

El entorno que se creó para el robot NAO se lo puede visualizar en la figura 3.22 en la cual se observa el plano en AutoCad, las paredes de la estructura son de espumaflex y el mueble de madera. Se escogió estas dimensiones ya que permiten una correcta movilidad y se adecuan al tamaño de robot.

Figura 3.22. Entorno creado para el robot NAO

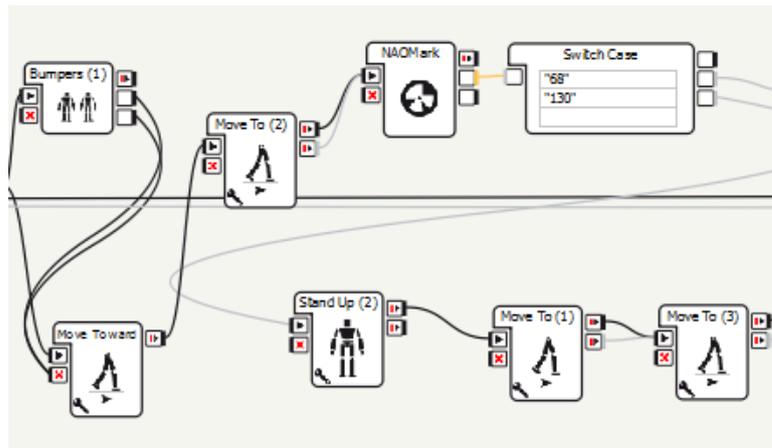


Entorno creado para la movilidad del robot NAO.

3.6 Movilización del robot usando NAOMarks.

Para la movilidad del robot en el entorno creado se procedió a colocar NAOMarks en diferentes ubicaciones y con ayuda de los sensores ubicados en los pies se realizó la programación que se muestra en la figura 3.23

Figura 3.23. Movilidad en el entorno



Programación de NAOmarks.

Se procedió a programar al robot para que camine hacia adelante hasta que los sensores en sus pies detecten el límite de la pared, en ese momento buscará el patrón de referencia; y cuando el robot detecta la NAOmark 68 girará a la izquierda y continuará caminando hasta llegar a otro tope.

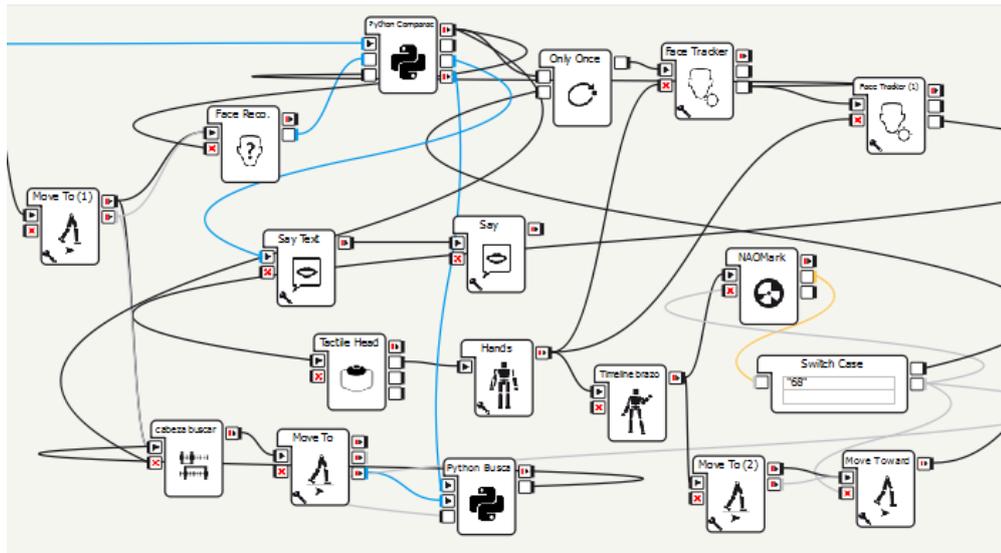
En el caso de que detecte la NAOmark 130 procederá a ubicarse en la posición de espera hasta que sea la hora de entregar la medicación.

A la hora establecida, el robot caminará hacia adelante hasta que los sensores en sus pies detecten el límite del mueble donde está el medicamento. Luego de que el robot ya recoja la medicación empezará a buscar al paciente correspondiente.

Primero NAO camina hacia adelante una distancia (1m) luego se detiene y procede a mover la cabeza de lado a lado buscando al paciente, si no lo encuentra gira un ángulo de 45° y vuelve a girar la cabeza, esto lo realiza cíclicamente barriendo los posibles sectores donde debería estar el paciente, la programación se puede observar en la figura 3.24.

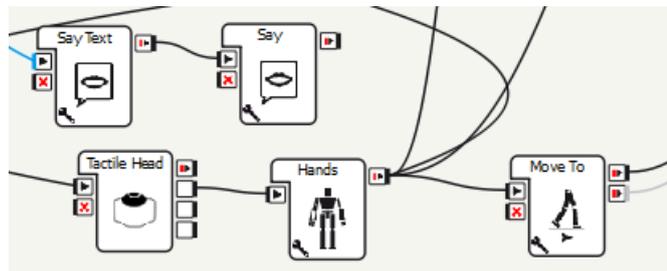
Una vez que reconoce al paciente se acerca para entregar la medicina diciendo su nombre; el robot entregará la medicación al tocarle un sensor de la cabeza. Esta programación se muestra en la figura 3.25

Figura 3.24. Reconocimiento del paciente



Programación del reconocimiento del paciente.

Figura 3.25. Entrega de la medicina al paciente



Programación para la entrega de la medicina.

A continuación en la figura 3.26 se muestra el proceso completo realizado por el robot NAO.

Figura 3.26. Proceso realizado por el robot

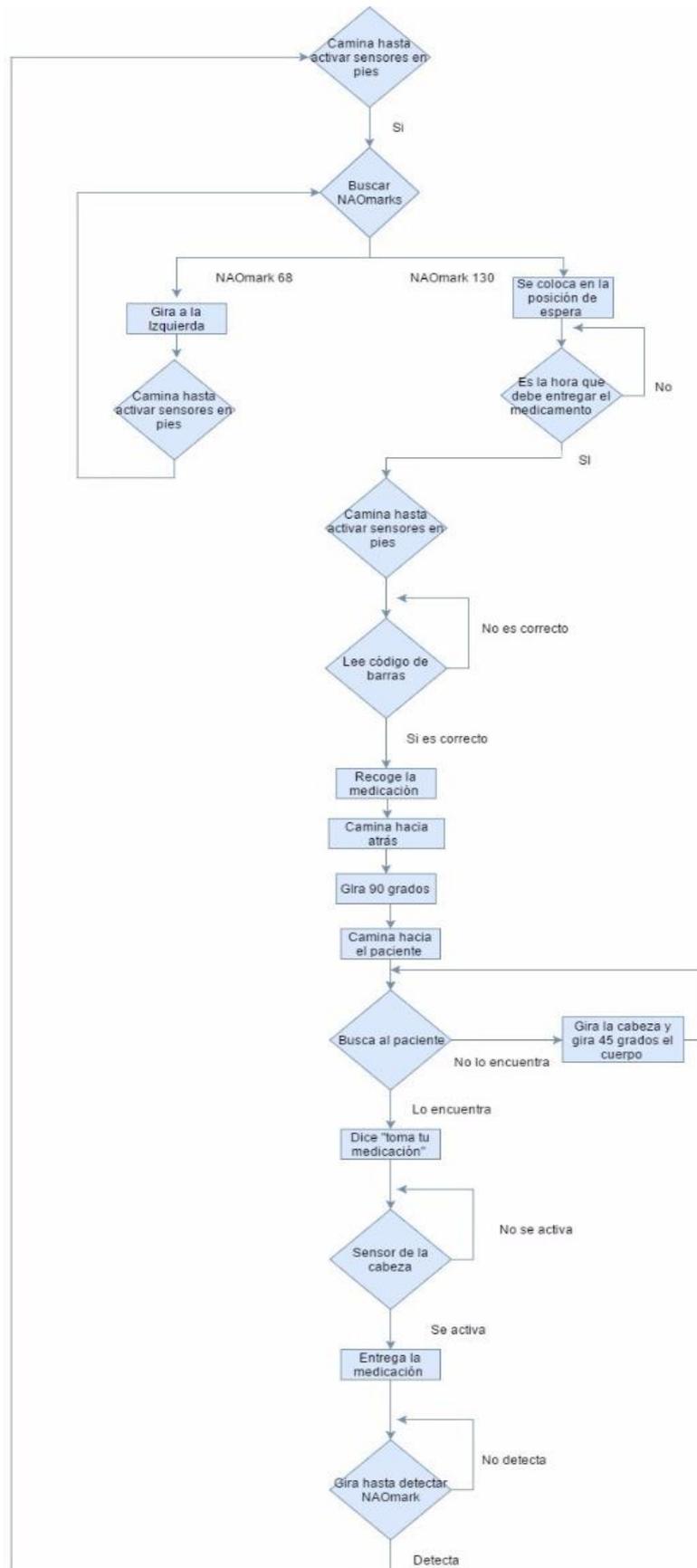


Diagrama de flujo del proceso que realiza el robot NAO.

CAPÍTULO 4

PRUEBAS Y ANALISIS DE RESULTADOS

4.1 Movilidad en el entorno

Al realizar las pruebas de campo para que el robot NAO recoja el medicamento se presentaron varias complicaciones, ya que al caminar con un medicamento muy grande éste se golpea contra las piernas del robot y bota la caja, por este motivo las cajas de los medicamentos se sugieren tengan una medida de 11cm x 4cm x 2.5cm y un peso de 60 gr. Adicionalmente se realizaron pruebas cambiando de posición la caja, pero tampoco es factible de esta manera, ya que se cambia el punto de gravedad del robot y provoca que pierda el equilibrio al momento de movilizarse.

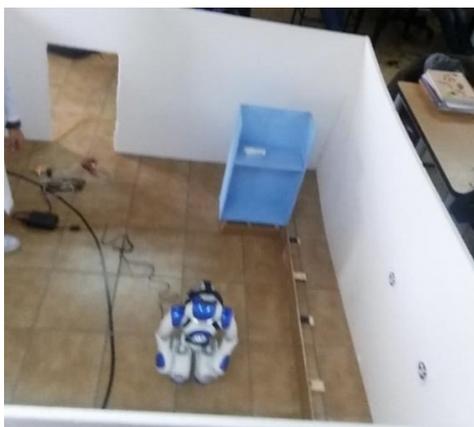
Al realizar pruebas de movilidad en el entorno creado para el robot NAO se obtuvo varios inconvenientes en el movimiento debido a que por problemas de hardware el robot no se mueve en línea recta teniendo una desviación, por lo que se decidió realizar el movimiento de diferentes formas como son NAOmarks, y mediante la utilización de sus sensores.

Al momento de las pruebas de campo se observó que el robot NAO tenía problemas al desplazarse debido a que su hardware está deteriorado por el uso, ya que este no puede caminar derecho sin que se produzca una desviación; para evitar este problema se planteó como solución darle un ángulo de giro mientras camina para compensar la desviación, pero debido a que la desviación no siempre es igual no es posible corregir este error mediante este método, luego se intentó corregir el error deteniéndolo y volviendo a poner en marcha intentando así disminuir el error, pero aun así persistió la desviación. También se probó girar el robot al final del trayecto un ángulo, pero al no ser siempre el mismo ángulo el que se desvía tampoco se puede aplicar este método.

Se procedió a utilizar la opción de movilidad como seguidor de línea, pero esta opción presentó problemas debido a la luminosidad del ambiente, ya que para que funcione de manera correcta se necesita que el ambiente tenga una luminosidad no mayor a 70 lux.

Finalmente se optó por utilizar las NAOmarks, los sensores de los pies del robot y unos topes a lo largo de la trayectoria establecida para el robot; de esta manera se logró que el mismo camine más recto y se redujo la pérdida de equilibrio. El entorno diseñado para el robot se puede observar en la figura 4.1.

Figura 4.1. Entorno real diseñado para el robot.



Entorno para el robot NAO.

A continuación, se muestran los resultados de las pruebas de campo.

Tabla 4.1. Pruebas realizadas.

	Pruebas	Toma correcta	Entrega correcta	Toma correcta (%)	Entrega correcta (%)
Pruebas sin métodos 28/11/2016-20/12/2016	12	1	0	8,33	0
Pruebas diferentes métodos 21/02/17 al 02/03/17	15	3	1	20	6,67
Prueba seguidor de línea 14/03/17 al 25/04/17	10	4	2	40	20
Prueba con topes 18/05/17	8	7	6	87,5	75
Prueba con topes 25/05/17	10	10	9	100	90
Porcentaje promedio de la última pruebas con topes				93,75	82,5

Pruebas realizadas por el robot NAO.

4.2 Funcionamiento

En la figura 4.2 se observa que el robot se detiene y detecta la NAOmark 68, y procede a girar a la izquierda como se observa en la figura 4.3

Figura 4.2. Identificación de la NAOmark 68.



Identificación de la NAOmark 68.

Figura 4.3. Giro a la izquierda.



El robot gira a la izquierda al detectar la NAOmark 68.

En la figura 4.4 se observa que el robot se detiene y detecta la NAOmark 130, procede a girar a la izquierda y ubicarse en la posición de espera como se observa en la figura 4.5

Figura 4.4. Identificación de la NAOmark 130.



Identificación de la NAOmark 130.

Figura 4.5. Giro a la izquierda y posición de espera.



El robot gira a la izquierda y se ubica en la posición de espera al detectar la NAOmark 130.

En la figura 4.6 se observa que el robot se comienza a caminar en busca del medicamento.

Figura 4.6. Búsqueda del medicamento.



El robot se dirige a buscar el medicamento.

En la figura 4.7 se observa que el robot está buscando el medicamento que debe entregar al paciente, al no ser el correcto sigue buscándolo, cuando se cambia al medicamento correcto, como se observa en la figura 4.8, procede a levantarse y tomarlo como se observa en la figura 4.9

Figura 4.7. Lectura del código de barras medicamento incorrecto



Lectura del código de barras cuando se tiene el medicamento incorrecto.

Figura 4.8. Lectura del código de barras medicamento correcto.



Lectura del código de barras cuando se tiene el medicamento correcto.

Figura 4.9. Toma del medicamento (prueba del medicamento cambiado)



Toma del medicamento correcto comparado previamente con el que está guardado en su base de datos

En la figura 4.10 se observa que no hay medicamento, en este caso el robot realiza varias búsquedas hasta encontrarlo como se puede observar en la figura 4.11, cuando ya encuentra el medicamento procede a levantarse y tomarlo como se observa en la figura 4.12.

Figura 4.10. Lectura sin medicamento



Lectura del código de barras cuando no hay un medicamento.

Figura 4.11. Lectura con medicamento



Lectura del código de barras cuando si hay un medicamento.

Figura 4.12 Toma del medicamento



Toma del medicamento correcto comparado previamente con el que está guardado en su base de datos

En la figura 4.13 se observa que el robot tomó el medicamento y ahora se dirige a buscar al paciente.

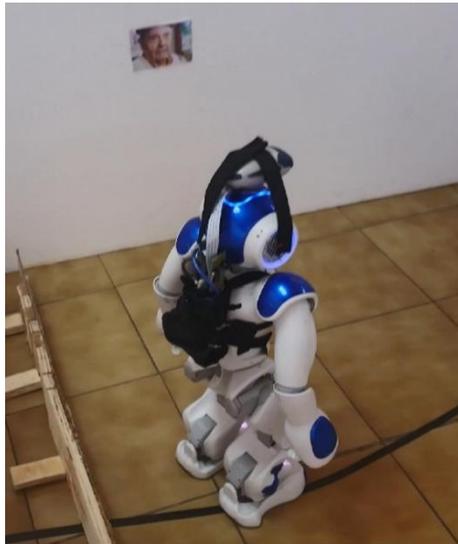
Figura 4.13. Robot dirigiéndose a buscar al paciente.



El robot tomó el medicamento y se dirige a buscar al paciente.

En la figura 4.14 se observa al robot NAO buscando al paciente “José”, girando la cabeza y su cuerpo como se explicó previamente en la sección 3.6, hasta que lo encuentra como se observa en la figura 4.15 y procede a acercarse y entregarle el medicamento como se observa en la figura 4.16.

Figura 4.14. NAO buscando al paciente “José”



NAO buscando al paciente “José” a través de girar la cabeza y su cuerpo.

Figura 4.15. NAO encuentra al paciente “José”



NAO encuentra al paciente “José” a través de girar la cabeza y su cuerpo.

Figura 4.16. NAO camina hacia el paciente “José”



NAO camina hacia el paciente “José” y procede a entregarle la medicación.

En la figura 4.17 se observa que al tocar el sensor de la cabeza del robot, éste entrega el medicamento.

Figura 4.17. Entrega de medicamento.



El robot entrega el medicamento al tocar el sensor.

CONCLUSIONES

Al implementar un peso extra en el robot NAO se determinó que pierde ligeramente su estabilidad al momento de desplazarse ya que se colocó una mochila que pesa aproximadamente 100gr para conectar el Arduino con la pistola lectora de código de barras. Esto generó que el punto de gravedad varíe y que el robot pierda estabilidad durante su movilidad; para reducir este error se colocó la maleta lo más centrado y pegado al cuerpo posible y se evidenció que se redujo la pérdida de equilibrio significativamente.

Se determinó que para realizar este proyecto los lenguajes de programación más apropiados son Python y Timeline de Choregraphe para la programación el robot NAO, ya que estos son lenguajes propios del robot y tiene una buena comunicación con el mismo, y para la lectura de código de barras se utilizó una interfaz USB/USB con un Arduino, el cual se programó en el software del mismo nombre basado en lenguaje C++.

Se comprobó que la información obtenida de la pistola lectora de códigos de barras y la hora a la que debe ser administrado el medicamento deben estar en el orden y formato igual al de la información previamente guardada en la base de datos ya que de otra manera no se realizará la comparación entre ambas correctamente y el robot no obedecerá a la programación.

Para la identificación de los medicamentos se concluyó que la cámara del robot NAO no era apta para leer códigos de barras ya que su resolución es de 1.22 megapíxeles y la resolución mínima para leer códigos de barras es de 5 megapíxeles aproximadamente; no se cambió por una cámara de mayor resolución debido a que el robot NAO necesitaría una computadora externa que procese la imagen y eso haría que el proceso no sea autónomo; por lo que se decidió realizar la lectura del código mediante una pistola lectora de códigos de barras, lo que significó el aumento en hardware y software, teniendo que aplicar ingenio y creatividad para lograrlo.

Se concluyó que el robot NAO no puede caminar en línea recta sin que se produzca una desviación en su trayectoria; y este error no se puede corregir mediante software como se comprobó en las pruebas de campo explicadas en el capítulo 4. Por ello se decidió colocar unos pequeños topes que ayuden a corregir el error en el movimiento.

Para la movilidad del robot con las NAO Marks se concluyó que el ambiente en donde se realiza el trayecto debe contar con una iluminación mayor a 100 lux, caso contrario el robot no es capaz de identificar la marca.

Se concluye una efectividad aceptable al momento de recoger el medicamento de aproximadamente el 82.5% el cual se lo realiza mediante la utilización de sus sensores y topes que ayudan a que el robot no se desvíe.

Se concluye que para la identificación del paciente era necesario que el robot pueda verlo desde cualquier posición, para lo cual se procedió a mover la cabeza de lado a lado para buscar al paciente, si no lo encuentra gira un ángulo de 45° y vuelve a girar la cabeza, esto lo realiza cíclicamente barriendo todos los posibles sectores donde debería estar el paciente, cubriendo los 365 grados hasta encontrarlo.

RECOMENDACIONES

Para realizar aplicaciones en las que el robot se desplace la superficie no debe ser totalmente lisa pero si debe permitir cierto grado de deslizamiento ya que si es una superficie con mucha fricción el robot pierde el equilibrio y puede llegar a caerse.

Para realizar aplicaciones que utilicen la hora como parámetro principal es recomendable verificar que la configuración de zona horaria del robot sea la misma que de la zona en donde se trabaja, esto se puede modificar al ingresar la dirección IP del robot NAO en un navegador.

Se recomienda que los medicamentos tengan una medida aproximada de 11cm x 4cm x 2.5cm y un peso de 60 gr, para que la movilidad del robot se realice de forma óptima.

Se recomienda la implementación de un dispensador de medicamentos para que siempre este el medicamento en el mismo lugar.

Se recomienda el uso de las NAOmarks para la movilidad ya que estas permiten que el robot pueda realizar diferentes acciones.

Se recomienda no usar el seguidor de línea para desplazarse debido a que se necesita de una iluminación específica para que funcione correctamente.

Se recomienda una luminosidad de entre 80-100 lux en el área donde se encuentran las NAOmarks para que estas se detecten con facilidad.

LISTA DE REFERENCIAS

- Aldebaran. (s.f.). Obtenido de http://doc.aldebaran.com/2-1/family/robots/connectivity_NAO.html
- Aldebaran. (s.f.). Obtenido de http://doc.aldebaran.com/2-1/family/robots/video_robot.html#robot-video
- Aliverobots. (2010). RobotNAO. Obtenido de <http://aliverobots.com/NAO/>
- Arduino. (2015). Proyectos ARDUINO. Obtenido de <https://proyectoarduino.wordpress.com/%C2%BFque-es-arduino/>
- Arduino. (2016). Obtenido de <https://www.arduino.cc/en/Main/ArduinoUSBHostShield>
- Arduino. (2016). ARDUINO . Obtenido de <http://arduino.cl/que-es-arduino/>
- CC, A. (2015). Arduino Pro Mini. Recuperado el 15 de 11 de 2015, de <https://www.arduino.cc/en/Main/ArduinoBoardProMini>
- Codigodebarras.com. (noviembre de 2016). Obtenido de http://www.codigodebarras.com/tema.php?ID=como_se_lee
- Codigos. (noviembre de 2013). Obtenido de <http://alimentosinocuos.blogspot.com/2013/11/codigo-de-barras-que-significa-cada.html>
- Kofinas, N. (2012, julio). Forward and Inverse Kinematics for the NAO Humanoid Robot. Atenas, Grecia.
- Kofinas, N., Orfanoudakis, E., & Lagoudakis, M. (. (2014). Complete Analytical Forward and Inverse Kinematics. Atenas, Grecia.
- Labs, S. (2016). Recuperado el 20 de noviembre de 2016, de Ventajas de una solución integrada basada en un microcontrolador de 32 bits: <http://www.silabs.com/products/mcu/pages/barcode-scanner-design-using-a-32-bit-microcontroller.aspx>
- Mbcestore. (2012). codigo de barras. Obtenido de <http://www.mbcestore.com.mx/codigo-de-barras/>

Perez, J & Merino, M. (2012). Definiciones. Obtenido de <http://definicion.de/trastorno-mental/>

Robotics. (2014). CHOREGRAPHE Software 2.1.4. Paris, Francia: Aldebaran Robotics.

Spong, M., Hutchinson, S., & Vidyasagar, M. . (2005). Control y modelamiento de robot. New York: JOHN WILEY & SONS, INC.

TheAmazel. (20 de Mayo de 2014). Youtube. Obtenido de <https://www.youtube.com/watch?v=UGj3H6ETHJg>