



UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA

# INTERACCIÓN Y MOVIMIENTO DEL ROBOT BAXTER MEDIANTE LENGUAJE NATURAL

**Por: Camilo Alonso Ruiz Bucarey**

Tesis presentada a la Facultad de Ingeniería de la Universidad de Concepción  
para optar al título profesional de Ingeniero Civil Informático

Abril 2024

Concepción, Chile

**Profesor Guía: Julio Erasmo Godoy del Campo**

© 2024, Camilo Ruiz B.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

Al futuro rey de los piratas, Monkey D. Luffy

## AGRADECIMIENTOS

Aprovecho este espacio para valorar y recordar a quienes me han apoyado a lo largo de esta etapa universitaria. Ha sido un viaje largo y lleno de tropiezos, bastante más de lo que me podría haber imaginado, pero de los cuales confío haber podido extraer diversos aprendizajes.

En primer lugar, quiero agradecer a mi profesor guía de este proyecto, Julio Godoy, quien siempre me apoyó y estuvo disponible para responder mis dudas, destacando por su paciencia en los periodos donde mi comunicación no fue la ideal.

A los profesores y profesoras del DIICC. Cada uno de ellos me ayudó a llegar a donde estoy hoy y estoy agradecido de haber podido trabajar en conjunto en diversas instancias, tanto proyectos e iniciativas impulsadas en mi etapa como centro de estudiantes, hasta seminarios realizados dentro de la universidad.

Por supuesto, a mis amigos con quienes compartí mis días y noches durante esta etapa, quienes me apoyaron e hicieron de mi vida universitaria algo más que solo trabajos y estudio. No pienso nombrarlos uno a uno pero bien saben quienes son.

Finalmente, a mis padres. Gracias por todo lo que me han entregado, de más esta decir que no estaría aquí sin ustedes. No soy capaz de verbalizar y menos de resumir en un par de oraciones mi agradecimiento. Gracias por haber estado siempre.



## Resumen

Esta Memoria de Título consiste en lograr que Baxter, un robot humanoide, realice diversas acciones definidas previamente según una orden de texto escrito o voz hablada en lenguaje natural. Se describe la implementación de un sistema que realiza dicha labor, interpreta el lenguaje natural, procesa la entrada recibida y detecta la acción indicada para finalmente ejecutarla por medio de Baxter. Se define cada una de las acciones posibles de realizar por Baxter, así como su jerarquía para la construcción de acciones más complejas. El sistema implementado es capaz de interpretar correctamente el lenguaje natural para realizar la acción deseada por el usuario, considerando oraciones claras respecto a la acción deseada. Se describen situaciones problemáticas que se presentaron durante el desarrollo y las acciones tomadas para completar el objetivo.

**Keywords** – Baxter, Lenguaje Natural (LN), Procesamiento de Lenguaje Natural (NLP)

# Índice general

|  |           |
|--|-----------|
| <b>AGRADECIMIENTOS</b>                                 | <b>I</b>  |
| <b>Resumen</b>   | <b>II</b> |
| <b>1. Introducción</b>                                 | <b>1</b>  |
| <b>2. Marco Teórico</b>                                | <b>4</b>  |
| 2.1. Procesamiento de lenguaje natural . . . . .       | 4         |
| 2.1.1. Reconocimiento automático del habla . . . . .   | 6         |
| 2.1.2. Procesamiento de texto . . . . .                | 8         |
| 2.2. Robótica . . . . .                                | 9         |
| 2.2.1. Baxter . . . . .                                | 11        |
| 2.2.1.1. Articulaciones de giro . . . . .              | 12        |
| 2.2.1.2. Articulaciones de inclinación . . . . .       | 14        |
| <b>3. Sistema de interacción</b>                       | <b>16</b> |
| 3.1. Definición e implementación de acciones . . . . . | 16        |
| 3.1.1. Acciones simples . . . . .                      | 18        |
| 3.1.2. Acciones compuestas . . . . .                   | 21        |
| 3.1.3. Acciones paralelas . . . . .                    | 22        |
| 3.1.4. Acciones complejas . . . . .                    | 24        |
| 3.2. Integración de PLN . . . . .                      | 26        |
| <b>4. Resultados</b>                                   | <b>30</b> |
| <b>5. Conclusiones</b>                                 | <b>40</b> |
| <b>Referencias</b>                                     | <b>42</b> |

# Índice de tablas

|   |    |
|---|----|
| 2.1. Límite y rango de giro . . . . .   | 13 |
| 2.2. Límite y rango de inclinación . . . . .                                    | 15 |
| 3.1. Algunas acciones involucradas en el saludo con ambas manos . . . . .       | 17 |
| 3.2. Lista de acciones simples realizables a solicitud del usuario . . . . .    | 20 |
| 3.3. Lista de acciones compuestas realizables a solicitud del usuario . . . . . | 21 |
| 3.4. Lista de acciones paralelas realizables a solicitud del usuario . . . . .  | 23 |
| 3.5. Lista de acciones complejas definidas realizables por Baxter . . . . .     | 25 |

# Índice de figuras

|   |    |
|---|----|
| 1.1. Robot Baxter . . . . .   | 3  |
| 2.1. Clasificación general de procesamiento de lenguaje natural . . . . . | 6  |
| 2.2. Esquema de la arquitectura de un reconocedor de voz . . . . .        | 7  |
| 2.3. Articulaciones del brazo del robot Baxter . . . . .                  | 12 |
| 2.4. Articulaciones de giro del robot Baxter . . . . .                    | 13 |
| 2.5. Articulaciones de inclinación del robot Baxter . . . . .             | 14 |
| 3.1. Robot Baxter en posición neutra . . . . .                            | 23 |
| 3.2. Construcción de una acción compleja . . . . .                        | 25 |
| 3.3. Elementos almacenados en el diccionario . . . . .                    | 26 |
| 3.4. Arquitectura del sistema . . . . .                                   | 29 |
| 4.1. Acción de abrazo del robot Baxter . . . . .                          | 32 |
| 4.2. Acción de abrir brazos del robot Baxter . . . . .                    | 33 |
| 4.3. Acción de levantar los brazos del robot Baxter . . . . .             | 34 |
| 4.4. Acción de celebración del robot Baxter . . . . .                     | 35 |
| 4.5. Acción de desconocimiento del robot Baxter . . . . .                 | 36 |
| 4.6. Acción de saludar con la mano izquierda del robot Baxter . . . . .   | 37 |
| 4.7. Acción de saludar con ambas manos del robot Baxter . . . . .         | 38 |

# Capítulo 1

## Introducción

En la actualidad, los robots se han vuelto cada vez más comunes en la sociedad. Su uso se extiende desde la industria hasta el hogar, y su versatilidad los hace ideales para una gran cantidad de aplicaciones. Sin embargo, el uso de los robots sigue siendo limitado por la necesidad de programar sus acciones mediante lenguajes de programación específicos. Esto supone una barrera de entrada para muchas personas que no tienen habilidades de programación y limita la utilización de los robots por parte de la sociedad en general.

La programación de robots, mediante lenguajes de programación específicos, es una tarea compleja que requiere habilidades técnicas especializadas. Este proceso puede ser costoso y lleva mucho tiempo. Además, en la gran mayoría de los casos, puede ser necesario programar de manera específica cada una de las tareas que deben realizar.

En este contexto, la interacción mediante lenguaje natural con los robots, la cual se refiere a la capacidad de los robots para interpretar y responder a los comandos en lenguaje natural, ya sea textual u oral, surge como una posible solución a esta barrera de entrada. En lugar de programar los robots mediante lenguajes de programación

específicos, los usuarios podrían interactuar con ellos utilizando lenguaje natural, lo que permitiría a una mayor cantidad de personas utilizar los robots, mejorando su usabilidad y accesibilidad debido a una interacción más intuitiva y natural.

Baxter (ver Figura 1.1) es un robot humanoide, desarrollado y distribuido por la compañía Rethink Robotics [1]. Posee 2 brazos con 7 articulaciones en cada uno para asemejar el movimiento del brazo humano (ver Figura 2.3), tres de estas simulan el movimiento de hombro, codo y muñeca, respectivamente, además de 4 articulaciones rotatorias. Cada articulación de los brazos posee su propio rango de movimiento en sentido horario y antihorario. Un par de pinzas o “grippers” ubicados como manos en cada brazo, las cuales permiten la manipulación de pequeños objetos. Una pantalla o “cara” con una cámara frontal y 2 articulaciones encargadas del movimiento vertical y horizontal de la “cabeza”. En dicha “cara” es posible mostrar información relevante o imágenes que apoyen el proyecto.

Si bien la producción de Baxter se encuentra actualmente descontinuada desde el año 2018 [2], aún es interesante su uso académico para la realización de diversas tareas y proyectos tal como puede ser el armado de cubos Rubik automático [3] o su uso para resolver el problema de *robotic grasping* [4].



**Figura 1.1:** Robot Baxter

El objetivo general de este trabajo es proponer un sistema que, mediante un modelo de procesamiento de lenguaje natural, permita comunicarse con el robot Baxter para que realice las acciones indicadas por voz hablada o texto escrito.

Como objetivos específicos se tiene:

1. Definir e implementar tareas realizables por el robot Baxter mediante el uso de ROS.
2. Implementar un modelo de procesamiento de lenguaje natural que se comunique con Baxter.

Para el desarrollo de este proyecto, se utilizó el robot Baxter presente en la Universidad de Concepción, ubicado en el Laboratorio FabLab del edificio Tecnológico Mecánico de la Facultad de Ingeniería.

# Capítulo 2

## Marco Teórico

### 2.1. Procesamiento de lenguaje natural

El procesamiento de lenguaje natural (PLN) es uno de los subcampos de la inteligencia artificial más importantes, el cual se encarga de que máquinas o sistemas sean capaces de manipular el lenguaje humano. Trabaja sobre ideas de la filosofía lingüística, así como sobre técnicas de representación y razonamiento del conocimiento lógico y probabilístico. Esta área requiere de un estudio empírico del comportamiento humano y uno de sus principales desafíos es la desambiguación de un mensaje con múltiples interpretaciones a partir del contexto [5].

Los inicios del PLN se dan a mediados del siglo XX como una intersección entre la inteligencia artificial y la lingüística, originalmente haciendo una separación con lo que eran las técnicas de indexación y búsqueda para extracción de información de un texto, para con el tiempo ir convergiendo entre sí [6]. En la actualidad, el PLN se encuentra en el día a día de muchas personas a lo largo del mundo a través de diversas herramientas y aplicaciones, algunas de estas son los asistentes virtuales como Siri o

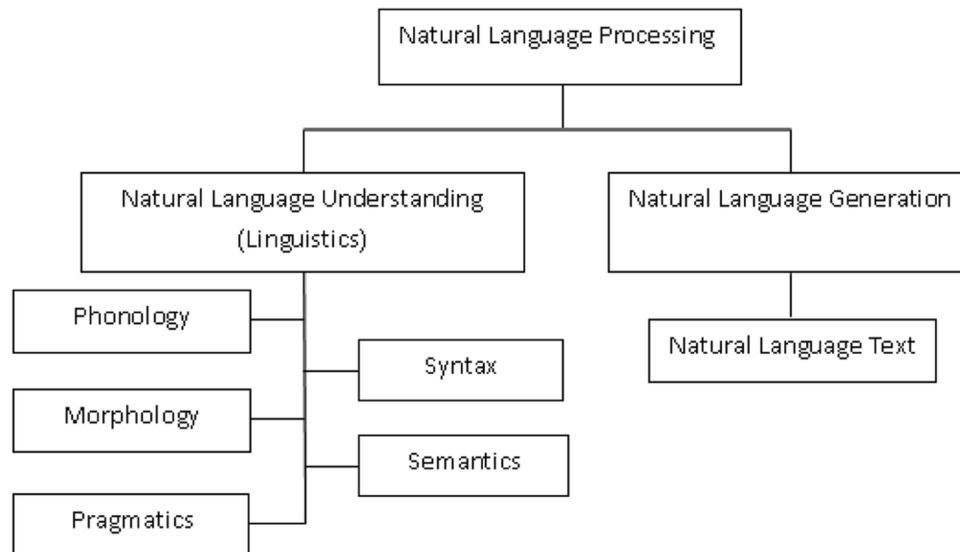


---

Alexa de Apple y Amazon respectivamente, y las herramientas de transcripción de texto automática que se pueden encontrar en Microsoft Teams o Skype.

A nivel general, dentro del PLN, es posible observar dos categorías principales; la comprensión del lenguaje natural y la generación de lenguaje natural [7]. Por el lado de la comprensión, se tiene todo lo relacionado con el entendimiento y análisis lingüístico en función del significado y contexto de los conceptos, emociones y palabras clave que se reflejan en el lenguaje, esto implica estudios a diversos niveles tal como fonética, morfología y sintaxis del lenguaje, entre otros. Por otro lado, se encuentra la generación de lenguaje natural, la cual como su nombre indica, se centra en el proceso de producir oraciones y/o párrafos que tengan un sentido lógico e incluso que logre simular una conversación entre personas. En la Figura 2.1 es posible apreciar la división y desglose de dichas categorías, considerando que no son las únicas existentes dentro del PLN, por ejemplo, actualmente existen importantes avances en la generación de voz hablada mediante IA, incluso siendo capaz de sintetizar voces cantantes con alta fidelidad [8].

En este sentido, la comprensión del lenguaje permite que una máquina reconozca las características fonéticas y sintácticas de una grabación de audio para identificar palabras de un idioma determinado y así realizar una transcripción de voz hablada a texto. Esto se conoce como reconocimiento automático del habla (RAH) y es la base para el desarrollo de una gran variedad de herramientas relacionadas con el PLN.



**Figura 2.1:** Clasificación general de procesamiento de lenguaje natural

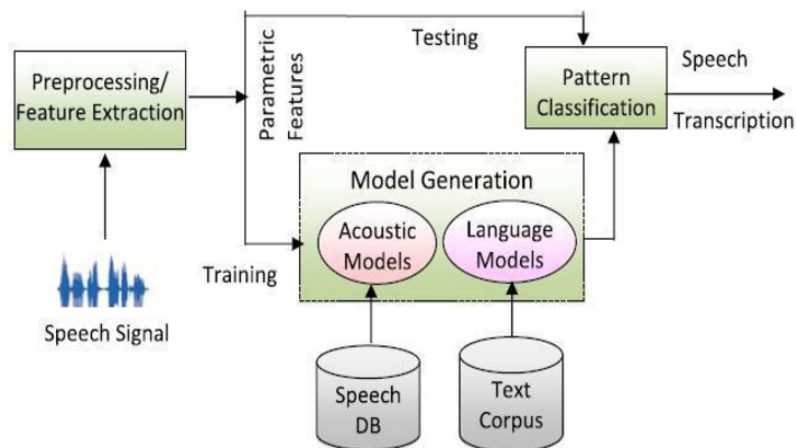
**Fuente:** Khurana et al. [7]

### 2.1.1. Reconocimiento automático del habla

IBM [9] define al reconocimiento del habla, también denominado conversión de voz a texto, como una función que utiliza el PLN para procesar el habla humana y transformarlo a formato escrito, así mismo hace la diferencia de éste con respecto al reconocimiento de voz, ya que se enfoca en un único usuario individual para la identificación de la voz mientras que el RAH posee un enfoque más amplio, independiente del individuo presentado. De esta manera, podemos definir el RAH como la capacidad de que una máquina o programa computacional identifique palabras u oraciones en formato de lenguaje hablado para así convertirlas en un formato comprensible por dicha máquina y posteriormente transcribir el mensaje a un formato de texto en un idioma determinado.

En la Figura 2.2 se puede apreciar el modelado básico para la generación de un sistema de reconocimiento de voz, considerando su fase de entrenamiento y su prueba

funcional. Es importante notar que, si bien en el proyecto actual no se realizó un entrenamiento de un modelo particular de reconocimiento de voz, entender sus bases es relevante para comprender de mejor manera el funcionamiento del mismo.



**Figura 2.2:** Esquema de la arquitectura de un reconocedor de voz  
**Fuente:** Saksamudre et al. [10]

Durante la etapa de *Pattern Classification* es posible encontrar una gran variedad de técnicas para el reconocimiento y clasificación de patrones, los cuales pueden ser desde técnicas basadas en comparación de plantillas de referencia que representen diccionarios de palabras, hasta técnicas basadas en algoritmos como *Dynamic Time Warping* el cual pondera la similitud entre diferentes secuencias basado en su variación de tiempo y velocidad [10]. En particular, y para efectos de esta memoria de título, se trabajó sobre la API de reconocimiento de voz provista por Google Cloud la cual está construida sobre un modelo de *Deep learning* de redes neuronales recurrentes pudiendo soportar procesos de reconocimiento complejos con un alto nivel de precisión. Las principales razones para la elección de plataforma son que provee una fácil integración a nivel de código, permitiendo el trabajo gratuito sobre versiones antiguas de la

librería *SpeechRecognition* de Python, así como buen rendimiento en la transcripción de voz hablada a texto escrito [11].

### 2.1.2. Procesamiento de texto

Parte fundamental del procesamiento de lenguaje natural, es la capacidad de procesar, limpiar y ordenar los caracteres y oraciones presentes en un texto, esto debido a la necesidad de mejorar la eficiencia de los algoritmos permitiendo obtener menores tiempos de procesamiento, así como mejores resultados al eliminar ambigüedades o porciones de texto que no necesariamente aporten información relevante para conseguir el objetivo deseado.

Para la gran mayoría de textos a procesar, es fundamental realizar eliminación de signos de puntuación, eliminación de mayúsculas, así como *stop words*, las cuales son aquellas palabras que no aportan un gran valor semántico al texto, conectores o preposiciones. De la misma forma, para el lenguaje español es posible filtrar las letras tildadas para así reducir la cantidad de caracteres posibles a encontrar.

Uno de los primeros pasos para el procesamiento de un texto es la tokenización, la cual es la técnica encargada de separar el texto en pequeñas unidades denominadas *tokens*. Esto implica separar un texto y sus oraciones en cada una de las palabras que se encuentran contenidas en este.

Otra de las técnicas utilizadas para procesar textos es la normalización, la cual tiene como objetivo estandarizar las palabras del texto, reduciéndolas a un forma base en común para así mejorar la eficiencia al reducir la información a procesar. Dentro de esta, es posible distinguir dos técnicas principales, el *stemming* y la lematización.

Por un lado, el *stemming* se encarga de reducir una palabra a su raíz, incluso si dicha raíz no pertenece al diccionario. El objetivo de esto es reducir diversas palabras similares a una misma raíz o un mismo prefijo en común, por ejemplo las palabras calcular, calculando y calculé, se verían reducidas a *calcul*.

Por otro lado, tenemos la lematización, la cual si bien es una forma más compleja de normalización de textos, puede ser sumamente beneficiosa en diversas áreas de aplicación [12]. El objetivo de esta técnica es reducir las palabras a su raíz en función del contexto, pero a diferencia del *stemming* se preocupa de que la palabra reducida no pierda significado.

Otra técnica propia del procesamiento de lenguaje natural es la coincidencia de cadenas o *string matching* la cual se puede utilizar para encontrar pasajes relevantes dentro de un motor de búsqueda, extraer información útil de grandes colecciones de textos o incluso corrección ortográfica automática y detección de plagio. Esta técnica puede tener aplicarse para comparar de manera literal la coincidencia de palabras entre textos mediante el uso de *algoritmos ingenuos* hasta la posibilidad de utilizar métodos más complejos como el algoritmo de Boyer-Moore o cálculo de distancia Levenshtein para así obtener mejores resultados sobre análisis de sentimientos u otras áreas que presenten diversos desafíos.

## 2.2. Robótica

La robótica, como área interdisciplinaria, se ha ido desarrollando en base a décadas de avances en inteligencia artificial y sistemas autónomos, logrando la integración de algoritmos inteligentes con ingeniería mecánica, permitiendo así la creación de máquinas capaces de realizar tareas complejas de manera autónoma.

Si bien el concepto general de un robot se encuentra presente en la mitología desde hace siglos atrás con las *Doncellas Doradas* supuestamente construidas por el dios griego Hefesto [13], no es hasta el desarrollo de la electricidad y la mecánica compleja durante la revolución industrial que se empieza a desarrollar adecuadamente el concepto moderno de robot.

En la actualidad, un robot se puede definir como un agente físico que realiza tareas manipulando el mundo físico por medio de efectores tales como piernas, ruedas, pinzas, entre otros [5]. Hoy en día podemos notar una convergencia entre la robótica y la inteligencia artificial, en donde la capacidad de los robots para comprender, aprender y colaborar junto a los humanos avanza de manera acelerada. De esta forma no solo se encuentran presentes en entornos de fabricación y automatización de procesos, sino que también se han desarrollado robots autónomos para su operación en diversas áreas como puede ser la exploración para el aprendizaje sobre entornos desconocidos.

En este contexto, durante el año 2007 surge *switchyard*, o como se conoce actualmente, Robot Operating System (ROS) el cual es un *framework* de código abierto para el desarrollo de software de robots que cuenta también con los servicios básicos propios de un sistema operativo, siendo uno de sus principales beneficios, la abstracción del hardware y la capacidad de permitir controlar un robot sin que el usuario necesariamente conozca todos los detalles de este en profundidad [14].

Existe una amplia variedad de robots que funcionan con ROS como sistema operativo, uno de ellos es el antes mencionado Baxter de la empresa Rethink Robotics [15].

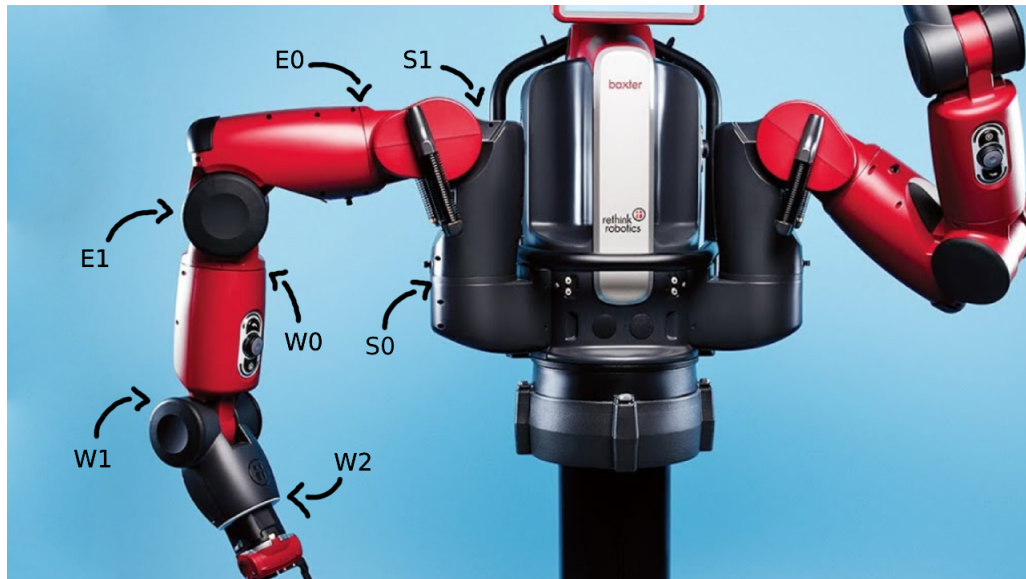
### 2.2.1. Baxter

Como se mencionó previamente, Baxter es un robot humanoide el cual posee siete grados de libertad y cuyas articulaciones están compuestas de actuadores elásticos en serie con resortes entre el motor/engranaje y la salida del actuador cuya elasticidad permite detectar fuerzas externas así como contactos con un humano.

La versión de Baxter de uso académico se maneja y configura usando un kit de desarrollo de software que se ejecuta sobre un computador remoto cuyo sistema cuenta con un sistema operativo Ubuntu 16.04, Python 2.7 y ROS en su versión *kinetic*. Este equipo es el encargado de entregar las instrucciones al robot por medio de una conexión local cableada.

Cada una de las articulaciones de los brazos del Baxter cuenta con capacidad de rotación independiente y se encargan de un movimiento en particular. A todas estas articulaciones se les asigna una letra y número como nombre en función de su movimiento y se pueden visualizar en la Figura 2.3. Así, cada articulación se nombra de la siguiente manera:

- **S0**: Giro del hombro
- **S1**: Inclinación del hombro
- **E0**: Giro del codo
- **E1**: Inclinación del codo
- **W0**: Giro de la muñeca
- **W1**: Inclinación de la muñeca
- **W2**: Giro de la muñeca (mano)



**Figura 2.3:** Articulaciones del brazo del robot Baxter

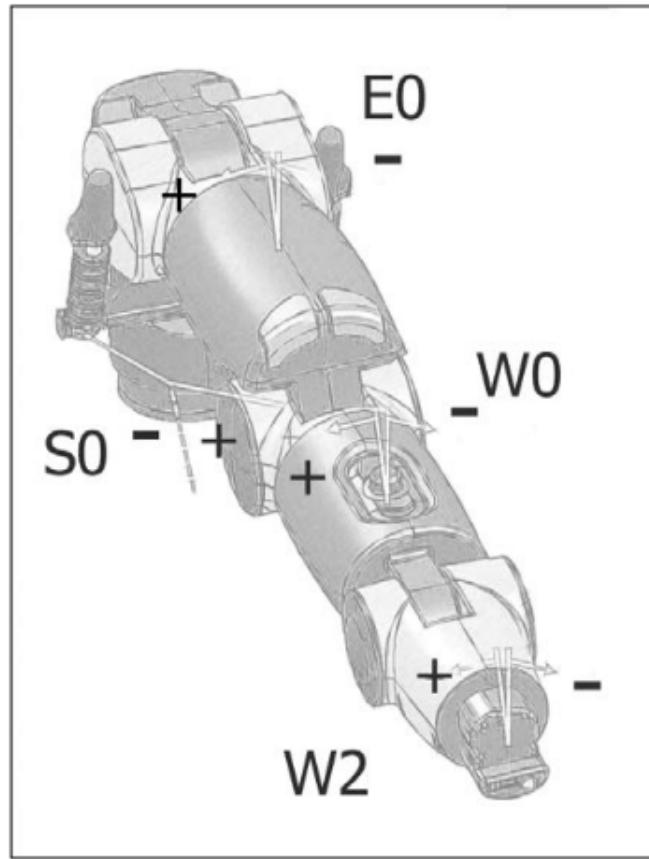
Por medio de la asignación de los nombres se referencia a cada articulación y es posible de manejar a través de código, cada una de estas posee un diferente rango de movilidad el cual es medido en radianes, para su mejor comprensión, las 7 articulaciones se separan en 2 categorías.

#### 2.2.1.1. Articulaciones de giro

Dentro de esta categoría se encuentran 4 de las articulaciones del Baxter, S0, E0, W0 y W2. Estas se encargan de orientar la sección correspondiente rotando su centro en torno a al eje central de la articulación.

Considerando un marco de referencia en donde observamos el Baxter y sus brazos frente a frente, un giro horario de la articulación representa un valor negativo del giro, por el contrario, un giro antihorario representa un giro positivo. Esto se puede apreciar mejor en la Figura 2.4.





**Figura 2.4:** Articulaciones de giro del robot Baxter

Fuente: Fairchild and Harman [14]

Al mismo tiempo, cada una de estas articulaciones posee un límite y rango de movimiento distinto definido en la Tabla 2.1.

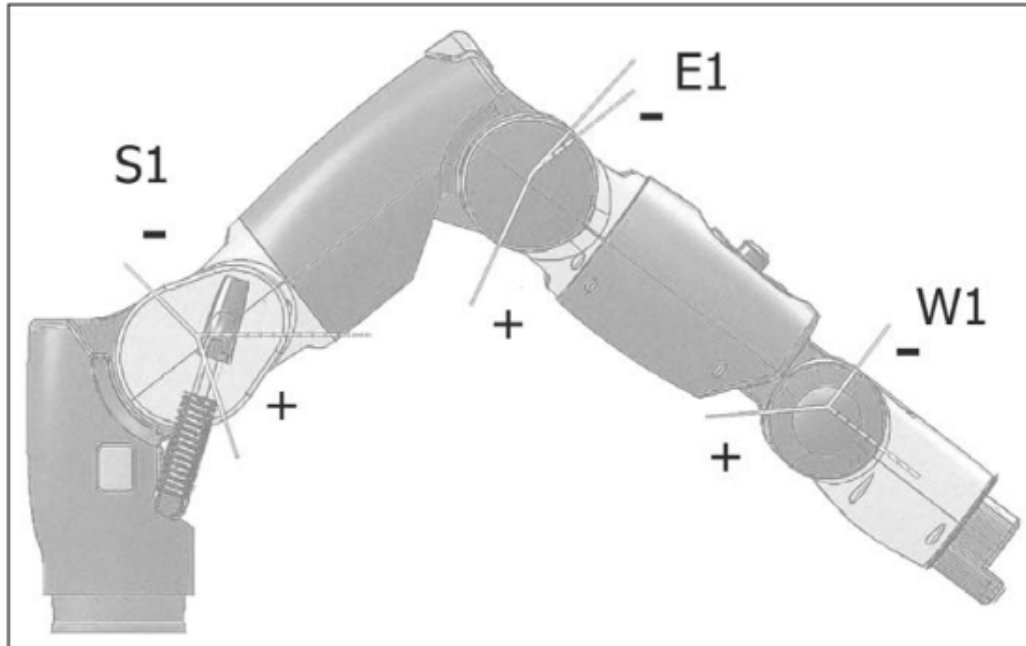
| Articulación | Mínimo      | Máximo  | Rango | Mínimo    | Máximo | Rango |
|--------------|-------------|---------|-------|-----------|--------|-------|
|              | en radianes |         |       | en grados |        |       |
| <b>S0</b>    | -141.00     | +51.00  | 192.0 | -2.461    | +0.890 | 3.351 |
| <b>E0</b>    | -173.50     | +173.50 | 347.0 | -3.028    | +3.028 | 6.056 |
| <b>W0</b>    | -175.25     | +175.25 | 350.5 | -3.059    | +3.059 | 6.118 |
| <b>W2</b>    | -175.25     | +175.25 | 350.5 | -3.059    | +3.059 | 6.118 |

**Tabla 2.1:** Límite y rango de giro

### 2.2.1.2. Articulaciones de inclinación

Las restantes 3 articulaciones del Baxter pertenecen a aquellas que se enfocan en movimientos inclinativos. Si bien, este tipo de articulaciones también realizan un movimiento en base a una rotación de la articulación, estas se inclinan hacia arriba o abajo rotando su eje de forma perpendicular a la articulación.

Los valores de inclinación son positivos cuando la dirección del movimiento tiende a “cerrar” la articulación, en el caso contrario, el valor es negativo cuando el movimiento busque extender la articulación. Esto se puede apreciar de mejor manera en la Figura 2.5, aparte de los rangos de movimiento de cada articulación, presentes en la Tabla 2.2.



**Figura 2.5:** Articulaciones de inclinación del robot Baxter  
**Fuente:** Fairchild and Harman [14]

| Articulación | Mínimo      | Máximo | Rango | Mínimo    | Máximo | Rango |
|--------------|-------------|--------|-------|-----------|--------|-------|
|              | en radianes |        |       | en grados |        |       |
| <b>S1</b>    | -123        | +80    | 163   | -2.147    | +1.047 | 3.194 |
| <b>E1</b>    | -3          | +150   | 153   | -0.052    | +2.618 | 2.670 |
| <b>W1</b>    | -90         | +120   | 210   | -1.571    | +2.094 | 3.665 |

**Tabla 2.2:** Límite y rango de inclinación

Las distintas características de movimiento, potenciadas por su gran número de articulaciones y amplio rango de movimiento de cada una de estas; su similitud a la fisionomía humana, considerando sus brazos, manos y cabeza; y las capacidades de desarrollo, potenciado por un *framework* como ROS, el cual, permite una fluida interacción gracias a su sistema operativo y kit de desarrollo de software; hacen de Baxter un excelente robot para el trabajo investigativo y demostrativo en diversas áreas que busquen una integración con la robótica, y en particular, de esta memoria de título.

## Capítulo 3

# Sistema de interacción

El sistema diseñado permite una interacción fluida entre el robot Baxter y el usuario. Para lograr dicha comunicación fue necesario definir las capacidades de Baxter, teniendo en cuenta las limitaciones propias de este y sus articulaciones, e implementar el sistema por donde el usuario entrega las instrucciones a realizar, en un formato que pueda ser procesado y comprendido por el robot. De esta forma, el trabajo desarrollado se divide en 2 porciones principales:

1. Definición, implementación y combinación de acciones realizables por el robot Baxter.
2. Integración de herramientas de procesamiento de lenguaje natural.

### 3.1. Definición e implementación de acciones

Inicialmente, se exploraron las capacidades del robot Baxter en cuanto a movimiento de sus articulaciones, considerando rango y dirección del movimiento, para así definir cada una de las posibles acciones a realizar.

De esta manera, se definieron diversas acciones realizables por el robot Baxter, las cuales tienen como objetivo emular el movimiento y comportamiento de una persona, involucrando movimientos con los brazos y manos, como puede ser un saludo, hasta capacidad de respuesta al asentir o negar con movimientos de cabeza luego de una pregunta.

De la misma forma, cada acción definida se descompone en subacciones simples que representan movimientos centrados en una única articulación y se categorizan en diversas jerarquías de acción. La jerarquía está dada según la complejidad del movimiento, así como el número de articulaciones involucradas en el mismo.

Para visualizar de mejor manera la definición de acciones, en la Tabla 3.1 es posible apreciar algunas de las acciones y su combinación para generar acciones de mayor jerarquía.

| <b>Acción</b>               | <b>Tipo</b> | <b>Descripción</b>   |
|-----------------------------|-------------|--|
| Levantar el brazo izquierdo | Simple      | Levanta el brazo izquierdo en un grado determinado                     |
| Levantar la mano izquierda  | Simple      | Coloca el antebrazo del robot en un ángulo de 90° en relación al brazo |
| Sacudir la mano izquierda   | Simple      | Mueve la mano del robot de derecha a izquierda                         |
| Saludar con la izquierda    | Compuesta   | El robot saluda con su mano izquierda                                  |
| Saludar con la derecha      | Compuesta   | El robot saluda con su mano derecha                                    |
| Saludar con ambas manos     | Compleja    | El robot mueve sus dos brazos para saludar                             |

**Tabla 3.1:** Algunas acciones involucradas en el saludo con ambas manos

Así es como para realizar una acción más compleja como puede ser *Saludar con ambas manos* se utilizan dos acciones de menor jerarquía como *Saludar con la mano izquierda* y *Saludar con la mano derecha*, las cuales a su vez se forman al ejecutar secuencialmente algunas acciones simples.

Para el movimiento del robot en casi todas las acciones, es necesario obtener la posición actual de las articulaciones involucradas en el movimiento, una vez obtenida la información, se calcula la nueva posición en la cual debe quedar la articulación en función de un rango de movimiento definido calculado en radianes. Si bien es posible ejecutar acciones secuencialmente con total libertad, para poder apreciar de mejor manera una acción concreta, se recomienda siempre colocar al robot Baxter en su posición “estándar”. Dicha acción se encuentra definida dentro del total de acciones y permite mover ambos brazos del robot a una posición neutra que imita a una persona con sus brazos relajados.

En base a lo explicado anteriormente, se definieron las 4 categorías de acciones presentadas a continuación.

### 3.1.1. Acciones simples

Corresponde al conjunto de acciones más primitivas que construyen cada una de las acciones de mayor jerarquía, o que también se pueden ejecutar de manera independiente. Se compone de un total de 20 acciones y cada movimiento se refleja sobre un único brazo o articulación del robot.

Esta categoría se encuentra compuesta principalmente por movimientos cortos y sencillos que involucran una única articulación. Una excepción de esto es el módulo de conversación integrado con ChatGPT, si bien esta acción se encuentra categorizada como simple, su ejecución involucra diversas acciones y provoca que la interacción

con el robot Baxter cambie temporalmente para llevar adelante una conversación.

Cada acción se define en un archivo independiente de código el cual es llamado al momento de ejecutar la acción, este al ejecutarse mueve la articulación correspondiente en función de un ángulo definido para realizar la acción deseada.

Dentro de las acciones simples se encuentra definida la instrucción que termina la interacción con el robot Baxter como “terminar”. Todas las acciones que el robot Baxter puede realizar a pedido del usuario se encuentran listadas en la Tabla 3.2.

| <b>Acción</b>                 | <b>Descripción</b>   |
|-------------------------------|--|
| Levantar el brazo izquierdo   | Levanta el brazo izquierdo en un grado determinado                     |
| Levantar el brazo derecho     | Levanta el brazo derecho en un grado determinado                       |
| Estirar el brazo izquierdo    | Estira el brazo izquierdo horizontalmente hacia un lado                |
| Estirar el brazo derecho      | Estira el brazo derecho horizontalmente hacia un lado                  |
| Levantar la mano izquierda    | Coloca el antebrazo del robot en un ángulo de 90° en relación al brazo |
| Levantar la mano derecha      | Coloca el antebrazo del robot en un ángulo de 90° en relación al brazo |
| Sacudir la mano izquierda     | Mueve la mano del robot de derecha a izquierda                         |
| Sacudir la mano derecha       | Mueve la mano del robot de izquierda a derecha                         |
| Enderezar el codo izquierdo   | Rota el antebrazo del robot en base a la orientación del brazo         |
| Enderezar el codo derecho     | Rota el antebrazo del robot en base a la orientación del brazo         |
| Adelantar el hombro izquierdo | Mueve hacia adelante el hombro del robot                               |
| Adelantar el hombro derecho   | Mueve hacia adelante el hombro del robot                               |
| Quebrar la muñeca izquierda   | Retrae la muñeca (gripper) del robot                                   |
| Quebrar la muñeca derecha     | Retrae la muñeca (gripper) del robot                                   |
| Sonreír                       | Carga y muestra una imagen sonriente en la pantalla del robot          |
| Cara de pregunta              | Carga y muestra una imagen de interrogante en la pantalla del robot    |
| Cara de enojo                 | Carga y muestra una imagen molesta en la pantalla del robot            |
| Izquierda Arriba Abajo        | El robot sube y baja el brazo completo                                 |
| Derecha Arriba Abajo          | El robot sube y baja el brazo completo                                 |
| Terminar                      | Indicativo de terminar la ejecución de las acciones                    |
| Conversar                     | Inicia módulo de conversación con el robot                             |

**Tabla 3.2:** Lista de acciones simples realizables a solicitud del usuario



### 3.1.2. Acciones compuestas

Conjunto de acciones que, construidas en base a las acciones simples, se ejecutan de manera secuencial para generar un movimiento completo.

La característica principal de este grupo de acciones es que actúan sobre un grupo de articulaciones pertenecientes a un mismo brazo o sobre la cabeza del robot, esto permite realizar acciones más concretas y esperables de una interacción, algunos ejemplos de acciones compuestas son saludar con alguna de las manos del robot o que este asiente con la cabeza mientras cambia la imagen proyectada en su pantalla.

A continuación, en la Tabla 3.3 se pueden apreciar todas las acciones compuestas que puede realizar el robot Baxter.

| Acción                    | Descripción   |
|---------------------------|---|
| Decir que sí              | El robot asiente con la cabeza mientras muestra una imagen alegre en su cara        |
| Decir que no              | El robot niega con su cabeza moviendo a los lados la cara con una imagen de rechazo |
| Saludar con la derecha    | El robot saluda con su mano derecha   |
| Saludar con la izquierda  | El robot saluda con su mano izquierda   |
| Celebrar con la derecha   | El robot levanta y mueve su brazo derecho a modo de celebración                     |
| Celebrar con la izquierda | El robot levanta y mueve su brazo izquierdo a modo de celebración                   |
| Incógnita izquierda       | El robot encoge su hombro izquierdo indicando duda o desconocimiento                |
| Incógnita derecha         | El robot encoge su hombro derecho indicando duda o desconocimiento                  |

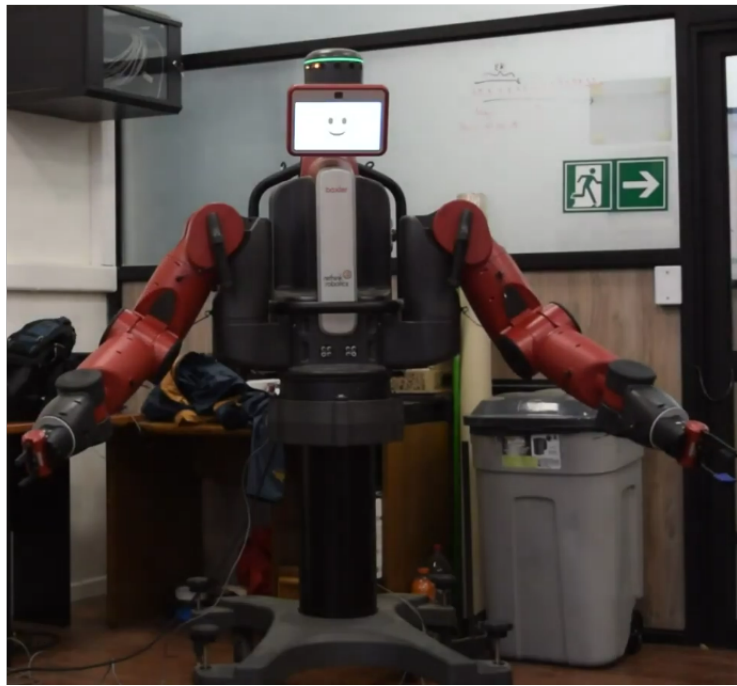
**Tabla 3.3:** Lista de acciones compuestas realizables a solicitud del usuario

### 3.1.3. Acciones paralelas

Conjunto de acciones que se ejecutan de forma paralela mediante el trabajo con hilos en el robot Baxter. Estas acciones se construyen en base a las acciones simples definidas y permiten el movimiento en simultáneo de ambos brazos del robot.

Para realizar el movimiento del robot, se toma control sobre un miembro del robot, ya sea alguno de sus brazos o su cabeza, y sobre este se ejecutan instrucciones que permiten mover cada una de las articulaciones. Dichas instrucciones se ejecutan en un proceso del computador mediante un nodo de ROS, debido a esto es que no es posible mover simultáneamente dos o más miembros del robot sin ejecutar las instrucciones sobre cada miembro a través de procesos separados. Es por esto que este conjunto de acciones trabaja de manera paralela aprovechando el trabajo sobre hilos del procesador para así poder mover diferentes miembros del robot.

Todas las acciones ubicadas en esta categoría se pueden encontrar listadas en la Tabla 3.4 y se ejecutan en base a dos acciones simples que mueven una misma articulación en ambos brazos del robot. En esta misma categoría podemos encontrar la instrucción “prepárate” (ver Figura 3.1) que coloca al robot Baxter en la posición neutra que se tiene definida, la cual permite visualizar de mejor manera cualquier instrucción posterior.



**Figura 3.1:** Robot Baxter en posición neutra

| <b>Acción</b>         | <b>Descripción</b>  |
|-----------------------|---|
| Prepararse            | El robot vuelve a su posición estándar                                  |
| Abrir los brazos      | El robot abre sus dos brazos al mismo tiempo (T-Pose)                   |
| Levantar los brazos   | El robot sube sus dos hombros en un ángulo dado                         |
| Sacudir las manos     | El robot sacude ambas manos de afuera hacia adentro                     |
| Adelantar los hombros | El robot mueve sus dos hombros hacia adelante                           |
| Quebrar las muñecas   | El robot retrae sus dos muñecas en función de la posición del antebrazo |

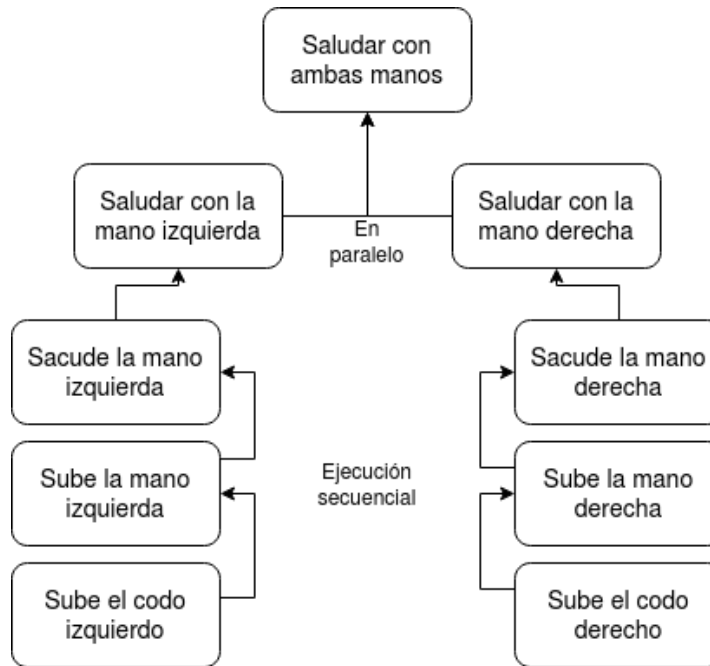
**Tabla 3.4:** Lista de acciones paralelas realizables a solicitud del usuario

### 3.1.4. Acciones complejas

Por último, las acciones complejas corresponden al conjunto de acciones de mayor jerarquía entre las cuatro categorías. Estas acciones, si bien están basadas sobre acciones simples al igual que las categorías anteriormente mencionadas, se ejecutan llamando instrucciones de mayor jerarquía como pueden ser acciones paralelas o compuestas, de manera que es posible realizar acciones de mayor complejidad. Al mismo tiempo, esta categoría tiene dos formas distintas de ejecutarse, siendo estas paralela y secuencial en función de las acciones que construyan la acción compleja.

El formato considerado que define una acción compleja a nivel de código, también permite hacer una diferencia entre distintas cantidades de elementos involucrados en la acción así como es necesario definir qué tipo de elementos conforman la acción a realizar, pudiendo variar entre instrucciones que representen acciones que ya se encuentren definidas y, por otro lado, archivos de código del proyecto que no necesariamente correspondan a una acción definida.

En la Figura 3.2 se puede apreciar la construcción de acciones de mayor jerarquía a partir de acciones simples, además de como se combinan distintas acciones para generar una acción compleja.



**Figura 3.2:** Construcción de una acción compleja

En la Tabla 3.5 presentada a continuación, se pueden apreciar las cuatro acciones complejas definidas para el robot Baxter.

| <b>Acción</b>           | <b>Descripción</b>   |
|-------------------------|--|
| Saludar con ambas manos | El robot mueve sus dos brazos para saludar   |
| Dar un abrazo           | El robot mueve sus brazos para abrazar hacia adelante  |
| Celebrar                | El robot celebra con ambos brazos  |
| Indicar desconocimiento | El robot se encoge de hombros y gesticula con las manos para indicar que no sabe lo que le piden o dicen |

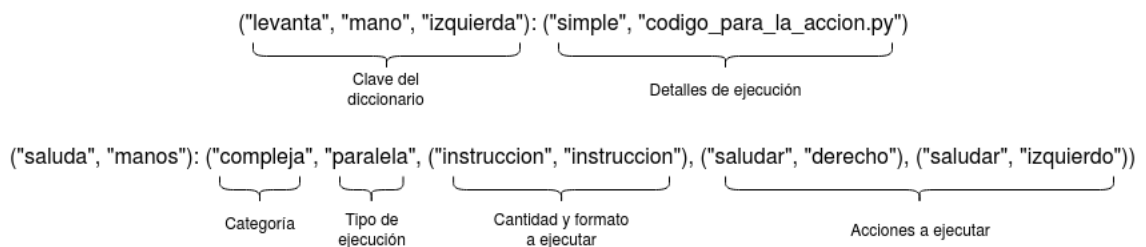
**Tabla 3.5:** Lista de acciones complejas definidas realizables por Baxter

## 3.2. Integración de PLN

Una vez definidas cada una de las acciones, fue necesario implementar el método de comunicación entre el usuario y el robot Baxter, para ello se almacenaron cada una de las acciones definidas dentro de un diccionario de tuplas que contienen palabras clave para la acción e información relevante para la ejecución de la misma, tal como el archivo que contiene el código para su ejecución y la categoría a la que pertenece dicha acción.

Para las palabras clave fue necesario considerar posibles ambigüedades debido a la similitud sintáctica entre algunas acciones. A cada acción se le asignaron una o más palabras clave que representan la intencionalidad de una solicitud de acción. Estas palabras clave se almacenan normalizadas como una tupla que representa la clave del diccionario.

El valor asignado a una clave corresponde a una tupla que siempre en su primera posición indica la categoría a la cual pertenece dicha acción, y posterior a ella, información relevante para la ejecución. En la Figura 3.3 se puede ver el par clave-valor de las acciones *Levantar la mano izquierda* y *Saludar usando ambas manos* siendo estas de categoría simple y compleja respectivamente.



**Figura 3.3:** Elementos almacenados en el diccionario

Por medio de una entrada de texto, la oración ingresada es procesada, se normaliza y *tokeniza* guardando cada una de las palabras o *tokens* resultantes en una tupla y se buscan estos elementos dentro del diccionario de acciones por medio de un proceso de comparación. Se considera y selecciona una acción siempre que se hayan encontrado todos los elementos de la tupla correspondiente dentro de la tupla resultante de la oración ingresada. En caso de no encontrar coincidencias, se realiza lematización sobre la entrada y se vuelve a comparar.

Una vez implementado el funcionamiento del programa mediante ingreso de oraciones de texto escrito, se trabaja sobre el reconocimiento de voz hablada y transcripción a texto. Para esto se trabaja con la librería *speech recognition* en su versión 3.8.1 con el módulo de reconocimiento de Google, esto permite interpretar una amplia variedad de palabras habladas en español para trabajarlas como texto plano.

Una vez funcionando por completo el sistema, con reconocimiento y transcripción de voz a texto, se integra la posibilidad de conectarse a ChatGPT por medio de su API para así poder simular una conversación. Esta conversación permite la consulta de todo tipo de preguntas a Baxter las cuales serán respondidas por medio de gesticulación indicando si la respuesta es afirmativa, negativa o una señal de desconocimiento si no es posible responder a dicha pregunta con las opciones anteriores, no conoce la respuesta o no logró entender la pregunta.

En primera instancia es necesario formular un *prompt* que permita entregar toda la información y parámetros de respuesta al robot. Esto permite limitar las respuestas de Baxter a las tres posibles respuestas descritas anteriormente y evita cualquier otro tipo de respuesta que no sea posible de representar por este.

A continuación se detalla el *prompt* utilizado para iniciar una conversación:

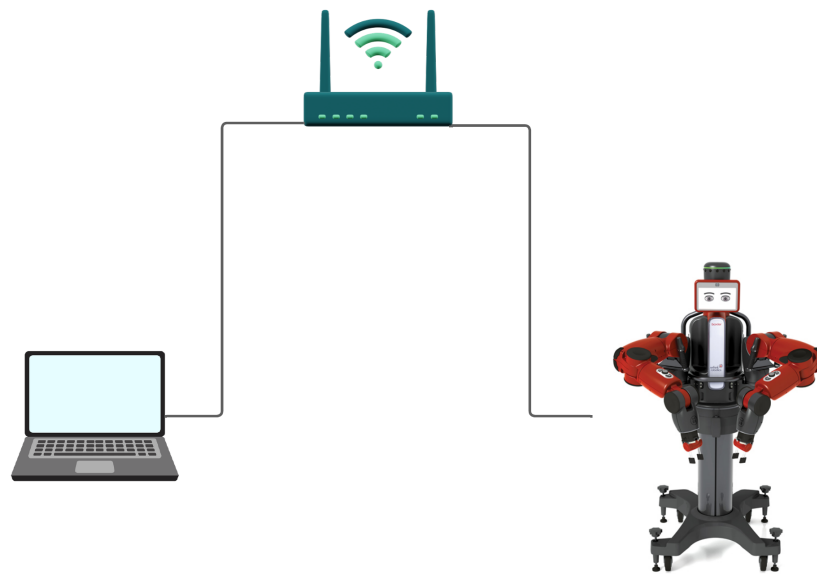
*Desde ahora eres Baxter, un robot humanoide que solo puede responder a las preguntas*

*con sí, no, imposible y no sé. No se permite ningún tipo de explicación sobre la respuesta ni tampoco se permite el uso de respuestas alternativas a las antes mencionadas. Siempre que la respuesta a la pregunta sea afirmativa, Baxter responde únicamente con la palabra Sí. No se permite el uso de palabras adicionales para complementar la respuesta. Y sólo se debe indicar Sí una única vez. Siempre que la respuesta a la pregunta sea negativa, Baxter responde únicamente con la palabra No. No se permite el uso de palabras adicionales para complementar la respuesta. Y sólo se debe indicar No una única vez. En caso de no poder responder a la pregunta utilizando las palabras Sí o No y no conocer la respuesta a la pregunta realizada, Baxter responde con No sé. Esto sin entregar mayor información respecto a porqué desconoce la respuesta. En caso de no poder responder a la pregunta utilizando las palabras Sí o No, saber la respuesta, pero no poder comunicarla mediante las palabras antes mencionadas, Baxter responde imposible. Esto sin entregar mayor información respecto a porqué le es imposible responder. Solo se permite el uso de las respuestas mencionadas anteriormente, el uso de cualquier otra palabra queda estrictamente prohibido. Baxter, ahora debes responder. ¿Entiendes las indicaciones antes mencionadas?*

Debido a que las consultas realizadas por medio de la API de ChatGPT no son capaces de almacenar el contexto de una misma conversación, fue necesario generar una lista que contenga toda la información de dicha conversación, indicando tanto el mensaje como el rol que desempeña el mensajero en la conversación. Dentro de una misma conversación cada una de las consultas realizadas al robot son entregadas junto con el contexto que contiene tanto el *prompt* como la respuesta a este. Una vez hecha la consulta, se recibe un mensaje en formato JSON que contiene una de las tres posibles alternativas de respuesta, este es procesado para ejecutar cada una de las acciones correspondientes que representen la respuesta del robot.



El sistema se ejecuta localmente desde un computador y se conecta al robot Baxter por medio de un *router* a través de una red cableada como se muestra en la Figura 3.4.



**Figura 3.4:** Arquitectura del sistema

El *router* es el encargado de transmitir la información entre el robot Baxter y el computador que ejecuta el código, además de transmitir la señal de internet que permite la conexión a la API de ChatGPT.

Es posible revisar el repositorio de GitHub que almacena el código fuente de este proyecto en el siguiente enlace:

<https://github.com/CamiloRuizBucarey/Baxter-NLPActions>

## Capítulo 4

### Resultados

En esta sección se presentan imágenes referenciales para la posición final del robot Baxter luego de la interacción, al indicar que se realicen algunas acciones específicas. Para cada una de las correspondientes imágenes se indica la posición inicial y el mensaje entregado, ya sea por voz hablada o texto escrito.

La mayoría de las acciones presentadas a continuación, muestran la posición final de los brazos correspondiente a un movimiento iniciado a partir de la posición estándar del robot Baxter, mas no es imperante la necesidad de ejecutar las acciones a partir de dicha posición. De cualquier forma, sí es recomendable volver a la posición estándar luego de cada acción ejecutada para poder apreciar de mejor manera la acción deseada y evitar colisiones entre piezas del robot o salir del rango de movimiento aceptado por las diversas articulaciones como consecuencia de una secuencia de acciones.

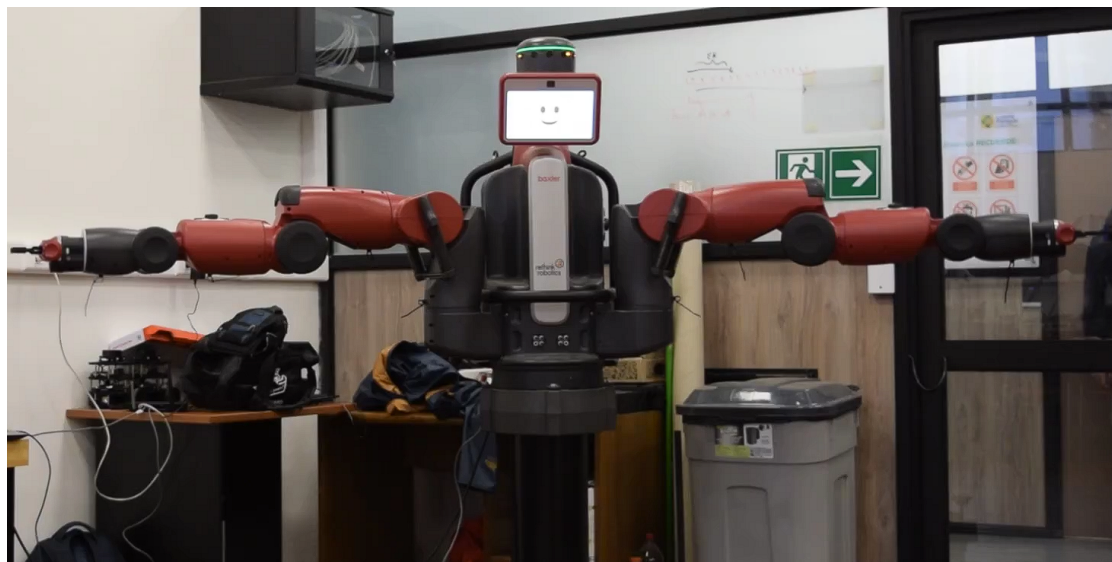
En la Figura 3.1, mostrada en el capítulo anterior, se puede apreciar la posición inicial estándar del robot Baxter. Esta posición corresponde a una acción paralela, pues es necesario mover ambos brazos simultáneamente para su ejecución, y busca emular una posición de descanso natural de una persona con sus dos brazos descansando a los lados. Independiente de la posición en la que se encuentre Baxter, siempre se moverá hasta la misma posición. Para esta acción es necesario indicar al robot que se prepare diciendo, por ejemplo: *Baxter, prepárate*.

En la Figura 4.1 se muestra la posición final de un abrazo dado por el robot Baxter a partir de la posición inicial estándar. Esta acción pertenece a la categoría de acciones complejas pues se construye a partir de pequeñas acciones más pequeñas, además de considerar el movimiento simultáneo de ambos brazos. Una posible oración que se puede decir para ejecutar esta acción es: *Dame un abrazo*.



**Figura 4.1:** Acción de abrazo del robot Baxter

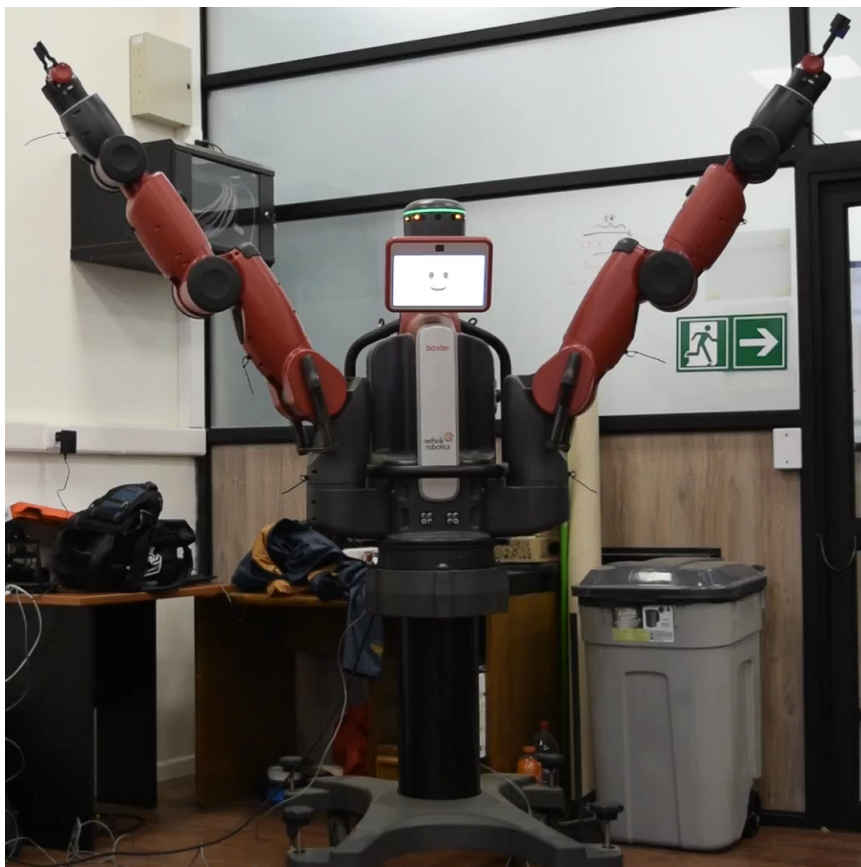
En la Figura 4.2 se puede apreciar una extensión completa de los brazos del robot Baxter en posición horizontal, también conocida coloquialmente como *T-Pose*. Esta acción siempre moverá los brazos a la posición mostrada independiente de la posición inicial. Nuevamente, al involucrar el movimiento de ambos brazos para la realización de la acción, esta corresponde la categoría de acciones paralelas.



**Figura 4.2:** Acción de abrir brazos del robot Baxter

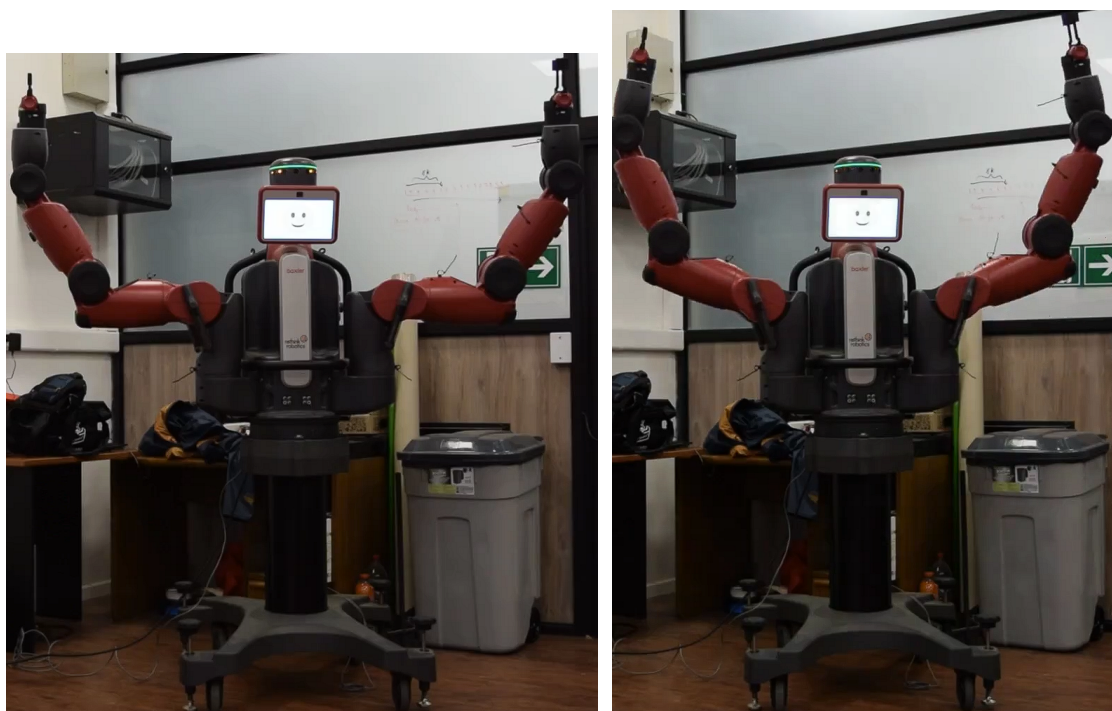
En la Figura 4.3 se muestra la posición del robot luego de indicar que se levanten ambos brazos a partir de una posición inicial con los brazos abiertos totalmente extendidos, como se mostró en la figura 4.2. Este movimiento corresponde a una acción paralela debido al movimiento simultáneo de los brazos del robot, siendo estos levantados un rango de 1 radian desde su posición inicial.

En conjunto a la acción de abrir los brazos, permite la ejecución de una acción conocida como *Praise the Sun*, popular en la cultura *geek* y perteneciente a la saga de videojuegos *Dark Souls*.



**Figura 4.3:** Acción de levantar los brazos del robot Baxter

En la Figura 4.4 se muestra la secuencia de la acción de celebración del robot Baxter al ser llamada a partir de la posición estándar. Este movimiento corresponde a una acción de categoría compleja en la cual Baxter en primera instancia levanta sus brazos y manos, para luego hacer un movimiento ascendente y descendente con estas, emulando una celebración. Para que Baxter realice esta acción, entre otras alternativas, se le puede decir *Baxter, celebra con ambos brazos*. Al final de esta acción, Baxter queda con sus brazos levantados en la posición mostrada en la figura 4.4a, por lo que es recomendable siempre volver a la posición estándar.



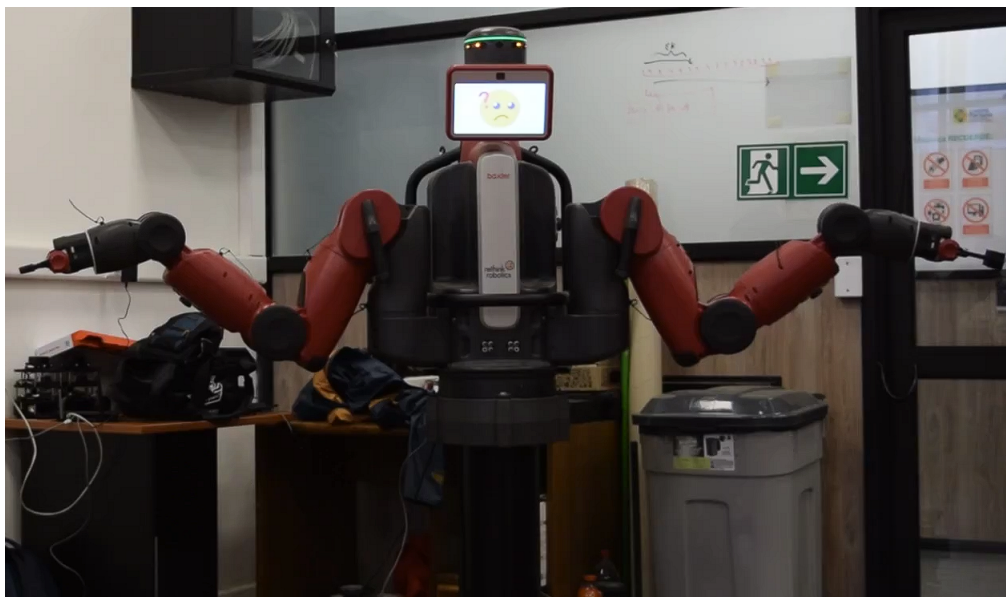
(a) Manos abajo

(b) Manos arriba

**Figura 4.4:** Acción de celebración del robot Baxter

En la Figura 4.5 se muestra cómo Baxter indica desconocimiento sobre algo que se le haya dicho, o también como demuestra que no ha logrado comprender lo indicado. Esta acción corresponde al conjunto de complejas al mover ambos brazos en simultáneo. Junto con el movimiento de los brazos, Baxter también mostrará una cara de interrogante a través de su pantalla durante la ejecución de la acción. Independiente de la posición inicial sobre la cual Baxter inicie la acción, siempre realizará el mismo movimiento, pasando primero a la posición estándar, para luego mover sus brazos en acción de desconocimiento y finalmente regresando a la posición estándar para finalizar la acción. Para que Baxter realice esta acción, basta con indicar algo que no logre entender, o también es posible realizarla diciendo, entre otras cosas, *Di que no sabes*.





**Figura 4.5:** Acción de desconocimiento del robot Baxter

En la Figura 4.6 se puede apreciar al robot Baxter saludando con su mano izquierda a partir de una posición estándar. Esto corresponde a una acción compuesta al requerir únicamente el movimiento de un brazo y estar formada por diversas acciones simples. Al realizar esta acción, Baxter levantará su codo y mano izquierda, para luego inclinar hacia ambos lados la mano emulando un saludo. Al finalizar la acción Baxter queda con su brazo levantado por lo que es recomendable siempre volver a la posición estándar antes de solicitar cualquier otra acción. Para pedir a Baxter que realice un saludo con su mano izquierda, se le puede decir *Baxter, saluda con tu mano izquierda.*





**Figura 4.6:** Acción de saludar con la mano izquierda del robot Baxter

Esta acción es realizable tanto con su mano derecha como con ambas manos en simultáneo, para ello solo es necesario indicar dicha diferencia haciendo alusión a su mano derecha o a ambas manos. En la Figura 4.7 se muestra la acción compleja correspondiente a saludar con ambas manos.



**Figura 4.7:** Acción de saludar con ambas manos del robot Baxter

Existen diversas situaciones que pueden producir un funcionamiento distinto al esperado por parte del robot Baxter. Uno de estos casos puede darse debido a una filtración de audio inesperado a través del micrófono, esto provoca que Baxter no logre comprender el sonido de entrada y realice la acción de desconocimiento. Otro caso que puede provocar un funcionamiento no deseado es el encadenamiento secuencial de diversas acciones sin volver a la posición estándar, de esta forma los brazos pueden recibir ordenes que intenten moverlos más allá de los límites propios de cada articulación ocasionando que no se logre ejecutar la acción como se espera. Finalmente, otro caso de error puede darse al realizar solicitudes de acciones simultáneamente sin permitir que Baxter termine de realizar una acción previa, esto puede provocar movimientos erráticos por parte de Baxter al intentar realizar diversos movimientos sobre una misma articulación.

Es posible revisar un video demostrativo completo con la interacción con el robot Baxter y la ejecución de las acciones en el siguiente enlace:

<https://youtu.be/o50HJ2NMo14>

# Capítulo 5

## Conclusiones

Con el desarrollo de este trabajo se demostró la capacidad del robot Baxter para la implementación de un sistema de interacción humano-máquina mediante lenguaje natural, tanto en texto escrito como voz hablada, que permita la realización de diversas acciones definidas a pedido del usuario.

En relación a los objetivos planteados, se concluye que estos sí se cumplen, pues se logró definir e implementar un total de casi 38 acciones de diversa complejidad realizables por el robot Baxter, así como la implementación de un modelo de lenguaje natural que permita procesar entradas en formato escrito y hablado para la interacción con el robot, las cuales son internamente procesadas para la comprensión y realización de las acciones definidas.

Por último, esto fortalece la labor académica y demostrativa del robot Baxter, y permite abrir nuevas líneas de trabajo que exploren más a fondo la interacción con éste mediante el uso del lenguaje natural, pudiendo trabajar interacciones más complejas mediante el uso de generadores de voz, los cuales implementen respuestas de voz hablada de parte del robot, e incluso percepción espacial mediante las cámaras y

sensores del robot para realizar acciones aún más variadas y personalizadas en función de la cantidad de personas presentes en torno al robot y su posición relativa a este.

## Bibliografía

- [1] Cliff Fitzgerald. *Developing baxter*, 2013.
- [2] Erico Guizzo and Evan Ackerman. Rethink Robotics, Pioneer of Collaborative Robots, Shuts Down. 9 2023. URL <https://spectrum.ieee.org/rethink-robotics-pioneer-of-collaborative-robots-shuts-down>.
- [3] César Bolivar, Roberto Asín, and Eduardo Méndez. *Resolviendo el cubo de Rubik con el robot Baxter.*, 2019. URL <http://repositorio.udec.cl/jspui/handle/11594/304>.
- [4] Angelo Montaña and Julio Godoy. *Aplicación de machine learning para el problema de robotic grasping*, 2019. URL <http://repositorio.udec.cl/jspui/handle/11594/348>.
- [5] Stuart J Russell and Peter Norvig. *Artificial intelligence a modern approach*. London, 2010.
- [6] Prakash M Nadkarni, Lucila Ohno-Machado, and Wendy W Chapman. Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551, 09 2011. ISSN 1067-5027. doi: 10.1136/amiajnl-2011-000464. URL <https://doi.org/10.1136/amiajnl-2011-000464>.
- [7] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744, 7 2022. doi: 10.1007/s11042-022-13428-4. URL <https://doi.org/10.1007/s11042-022-13428-4>.
- [8] Rongjie Huang, Chenye Cui, FeiYang CHEN, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 2525–2535, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392037. doi: 10.1145/3503161.3547854. URL <https://doi.org/10.1145/3503161.3547854>.
- [9] International Business Machines Corporation. *What is speech recognition?* <https://www.ibm.com/topics/speech-recognition>, 2023. URL <https://www.ibm.com/topics/speech-recognition>.

- 
- [10] Suman Saksamudre, P.P. Shrishrimal, and Ratnadeep Deshmukh. A review on different approaches for speech recognition system. *International Journal of Computer Applications*, 115:23–28, 04 2015. doi: 10.5120/20284-2839.
- [11] Binbin Xu, Chongyang Tao, Zidu Feng, Youssef Raqui, and Sylvie Ranwez. A benchmarking on cloud based speech-to-text services for french speech and background noise effect, 2021.
- [12] Michal Toman, Roman Tesar, and Karel Jezek. Influence of word normalization on text classification. *Proceedings of InSciT*, 4(354-358):9, 2006.
- [13] Deborah Levine Gera. *Ancient Greek ideas on speech, language, and civilization*. 1 2003.
- [14] Carol Fairchild and Thomas L. Harman. *ROS Robotics By Example*. Packt Publishing, 2016. ISBN 1782175199.
- [15] Institute of Electrical and Electronics Engineers and Robots Team. *Baxter* <https://robotsguide.com/robots/baxter>, 12 2023. URL <https://robotsguide.com/robots/baxter>.