Universidad Santo Tomás
Facultad de Ingenieria Electrónica
Bogotá, Colombia
2023

DESARROLLO DE UN SOFTWARE DE ROBÓTICA SOCIAL PARA LA ASISTENCIA DE PERSONAS EN OPERACIÓN DE FUNCIONES DE DOMÓTICA BASADO EN ROS

Laura Alejandra Herrera Quiñones
Marlon Sebastián Rodríguez Ramírez

Proyecto de grado presentado como requisito para optar al título de
INGENIERO ELECTRÓNICO

Director: M. Sc. Armado Mateus Rojas
Coodirectora: M. Sc. Sindy Paola Amaya

GED (Grupo de Estudios y Desarrollo en Robótica)
Universidad Santo Tomás
Facultad de Ingeniería Electrónica
Bogotá, Colombia
2023

Autoridades de la Universidad

RECTOR GENERAL
R.P. ÁLVARO JOSÉ ARANGO RESTREPO, O.P.

VICERRECTOR ADMINISTRATIVO Y FINANCIERO GENERAL
R.P. FRAY HERNÁN YESID RIVERA ROBERTO, O.P.

VICERRECTOR ACADÉMICO GENERAL
R.P. FRAY MAURICIO ANTONIO CORTÉS GALLEGO, O.P.

SECRETARIA GENERAL
DRA. INGRID LORENA CAMPOS VARGAS

DECANO DIVISIÓN DE INGENIERÍAS
R.P. FRAY JAVIER ANTONIO HINCAPIÉ ARDILA, O.P.

SECRETARIA DE DIVISIÓN
E.C. LUZ PATRICIA ROCHA CAICEDO

DECANO FACULTAD DE INGENIERÍA ELECTRÓNICA
ING. CARLOS ANDRÉS TORRES PINZON

Nota de aceptación

Firma del autor

Firma del jurado

Firma del jurado

BOGOTÁ D.C. ——— DE 2023

Advertencia

La Universidad Santo Tomás no se hace responsable de las opiniones y conceptos expresados en el trabajo de grado, solo velará por qué no se publique nada contrario al dogma ni a la moral católica y porque el trabajo no tenga ataques personales y únicamente se vea el anhelo de buscar la verdad científica.

Capítulo III –Art. 46 del Reglamento de la Universidad Santo Tomás.

Dedicatoria

A las personas que han creído en nosotros y nos han apoyado en el proceso, gracias por todo el apoyo.

Marlon Sebastián Rodríguez Ramírez
Laura Alejandra Herrera Quiñones

Índice general

Resumen

Este trabajo de grado tiene como objetivo principal proporcionar asistencia a aquellas poblaciones que enfrentan dificultades en la interacción con la tecnología en su vida diaria. Para lograr este propósito, se ha desarrollado un software diseñado para ayudar a las personas a controlar funciones relacionadas con BAC (Building Automation Control).
Durante el desarrollo de este proyecto, se han integrado modalidades auditivas, visuales y de texto al sistema. La información recopilada a través de estas modalidades ha ampliado la percepción del sistema en su entorno. Esta información adicional se utiliza para deducir detalles importantes y complementarios relacionados con las funciones solicitadas por el usuario.

Para la implementación, se han empleado diversas herramientas y repositorios de código abierto que se han integrado en el entorno de ROS (Robot Operating System). Esto ha permitido establecer comunicaciones entre nodos, procesar datos y extraer características. Posteriormente, los datos se someten a un preprocesamiento y se preparan para un modelo de lenguaje al cual se le aplicó un *Fine-tuning*. Este modelo se encarga de discernir los dispositivos y el nivel de complejidad de las instrucciones proporcionadas. Finalmente, estas solicitudes se envían a un servidor ubicado en un microcontrolador (el cuál sirve como prototipo para Controlador de Automatización Directa - DDC y sistema de Automatización Industrial - ISA) que ejecuta las acciones requeridas en dispositivos reales dentro de una vivienda a escala. Todas las comunicaciones entre el servidor y el cliente se realizan siguiendo el protocolo BACnet.

Palabras Clave:

- **Asistencia**
- **Tecnología**
- ***Software***
- **Domótica**
- **ROS**
- **Multimodal**
- **Representación compartida**
- **Comunicaciones**
- **Procesamiento de datos**
- ***Fine-tuning***
- **Modelo de lenguaje**

- **Microcontrolador**
- **Protocolo BACnet**

Abstract

This thesis aims to provide assistance to populations facing challenges in their daily interaction with technology. To achieve this goal, a software has been developed to assist individuals in controlling functions related to Building Automation Control (BAC).

Throughout the project development, auditory, visual, and text modalities have been integrated into the system. The information gathered through these modalities has expanded the system's perception of its environment. This additional information is used to deduce important and complementary details related to the user's requested functions.

For the implementation, various open-source tools and repositories have been employed and integrated into the Robot Operating System (ROS) environment. This has facilitated communication between nodes, data processing, and feature extraction. Subsequently, the data undergo preprocessing and are prepared for a language model that has undergone fine-tuning. This model is responsible for discerning the devices and the level of complexity of the provided instructions. Finally, these requests are sent to a server located on a microcontroller (which serves as a prototype for Direct Digital Control - DDC and Industrial Automation System - ISA) that executes the required actions on real devices within a scaled-down household. All communications between the server and the client are conducted following the BACnet protocol.

Keywords:

- **Assistance**
- **Technology**
- **Software**
- **Home Automation**
- **ROS (Robot Operating System)**
- **Multimodal**
- **Shared Representation**
- **Communications**
- **Data Processing**
- **Fine-tuning**

- **Language Model**
- **Microcontroller**
- **BACnet Protocol**
- **NLP**

Índice de figuras

Índice de tablas

Glosario

Se presenta un glosario con las palabras con mayor relevancia dentro del proyecto.

- **API (Interfaces de Programación de Aplicaciones):** Conjunto de reglas y protocolos que permiten que diferentes aplicaciones se comuniquen entre sí.
- **Adaptación de Dominios:** La capacidad de un sistema para ajustarse a diferentes dominios o entornos.
- **BACnet Protocol:** Un protocolo de comunicación utilizado en sistemas de automatización de edificios.
- **Codificadores Automáticos Variacionales:** Algoritmos basados en modelos generativos profundos que asumen que los datos son estocásticos y generados por una representación global.
- **Crawling:** Recopilación de datos de la web, comúnmente utilizada para la minería de datos.
- **Dataset:** Un conjunto de datos utilizado para entrenar modelos de aprendizaje automático.
- **Domótica:** Automatización de viviendas a través de sistemas tecnológicos.
- **Fine-tuning:** Ajuste fino, se refiere a ajustar un modelo preentrenado para que cumpla tareas específicas.
- **Google MediaPipe:** Un framework que permite el procesamiento de información de múltiples modalidades, incluyendo audio e imágenes.
- **HVAC:** Calefacción (Heating), Ventilación (Ventilation), Aire Acondicionado (Air Conditioning).
- **Inmótica:** Automatización de edificios a gran escala.
- **Información Heterogénea:** Datos que provienen de diferentes fuentes, como texto, imágenes, audio, etc., y que pueden requerir técnicas de procesamiento diferentes debido a sus diferentes características y distribuciones.
- **Información No Estructurada:** Datos que no siguen una estructura organizada, como texto sin formato, imágenes, audio, y que requieren técnicas avanzadas de procesamiento para extraer información.
- **Interacción Multimodal:** La combinación de diferentes modalidades de entrada, como texto e imágenes.
- **Large Language Model (LLM):** Modelos de lenguaje con un gran número de parámetros utilizados para procesar y generar texto.
- **LSTM (Memoria de Corto y Largo Plazo):** Un tipo específico de red neuronal recurrente que aborda el problema del desvanecimiento del gradiente y permite mantener dependencias a largo plazo en las secuencias de datos.
- **Machine Learning (Aprendizaje Automático):** Un enfoque para el desarrollo de modelos que pueden aprender y mejorar a partir de datos.

- **Minería de Datos:** El proceso de descubrimiento y extracción de patrones útiles en grandes conjuntos de datos.
- **Multimodal:** Referente a la combinación de múltiples modalidades, como texto e imágenes.
- **OpenAI:** Una organización de investigación en IA.
- **Procesamiento del Lenguaje Natural (NLP):** Un campo de la IA que se centra en la interacción entre las computadoras y el lenguaje humano.
- **Procesamiento Natural del Lenguaje (NLP):** Un campo de la IA que se centra en la interacción entre las computadoras y el lenguaje humano.
- **Protocolo BACnet:** Un protocolo de comunicaciones utilizado en la automatización de edificios para el control de dispositivos, compatible con diferentes protocolos de comunicación.
- **Redes Neuronales Convolucionales (CNN):** Son un tipo de red neuronal diseñada para procesar datos de imágenes, caracterizadas por el uso de capas de convolución que extraen características visuales.
- **Redes Neuronales Recurrentes (RNN):** Son un tipo de red neuronal que se utiliza en el procesamiento de secuencias de datos, como el lenguaje natural, debido a su capacidad para mantener información anterior a medida que procesan nueva información en una secuencia.
- **Reconocimiento de Emociones:** La capacidad de un sistema, como un robot, para identificar las emociones de las personas.
- **Representaciones Compartidas:** La creación y utilización de representaciones comunes de datos heterogéneos para su uso en múltiples tareas o dominios.
- **Robot Operating System (ROS):** Es un conjunto de librerías y herramientas de software que ayudan a crear aplicaciones para robots.
- **Robótica Social:** Un campo que busca la coexistencia de robots y humanos en entornos sociales.
- **Scikit-learn:** Una librería de aprendizaje automático de código abierto para Python que ofrece herramientas para análisis de datos y construcción de modelos.
- **Speech-to-Text (S2T):** Conversión de discurso en texto.
- **Superposición:** Capacidad de combinar o fusionar múltiples representaciones o conceptos en un mismo espacio.
- **Tensorflow:** Una plataforma de aprendizaje profundo de código abierto desarrollada por Google.
- **Tokenización:** El proceso de asignar valores numéricos a palabras o elementos de un conjunto de datos, lo que reduce la cantidad de información a procesar y permite el entrenamiento de modelos de lenguaje.
- **Transformers:** Una arquitectura de red neuronal que ha demostrado un rendimiento excepcional en el procesamiento del lenguaje natural y otras tareas, basada en mecanismos de atención.
- **Vosk:** Un conjunto de herramientas de reconocimiento de voz de código abierto basado en modelos de redes neuronales profundas.
- **YOLO (You Only Look Once):** Un algoritmo avanzado de detección de objetos en imágenes en tiempo real basado en redes neuronales convolucionales.

Capítulo 1

Introducción

El proyecto gira en torno a la brecha que existe entre la tecnología y muchas de las personas mayores. A raíz de esto, se formula el proyecto con el objetivo de desarrollar un software de asistencia en la operación de funciones de domótica. Con la integración de redes neuronales, el protocolo de comunicación BACnet y ROS (Robot Operating System).

1.1. Planteamiento del Problema

Existe una carencia de mecanismos que permiten a robots sociales como Pepper, Nao, Romeo o Furthat, y a robots domésticos como Toyota HSR, iRobot Roomba o Astro, el asistir a personas con analfabetismo digital en la operación de funciones de automatización de edificios. La domótica y los sistemas BMS (sistemas de administración de edificios) proveen al hogar de aplicaciones de automatización como HVAC (calefacción, ventilación y aire acondicionado, por sus siglas en inglés), seguridad, iluminación, entre otras que se encuentran más allá del alcance de los robots inicialmente nombrados. Evidencia de esto, es la no existencia dentro de los repositorios oficiales de ROS de funcionalidades específicas para sistemas de automatización de edificios como BACnet, Lonworks o KNX, que son protocolos de comunicación utilizados en este campo.

El rápido desarrollo de la tecnologías de la información y comunicación (TIC) representada tanto por los dispositivos inteligentes como por las diferentes tendencias de desarrollo en software, ha generado que parte de la población, especialmente la población de edad avanzada, no desarrolle las habilidades y competencias digitales necesarias para operarla, manteniendo a este grupo poblacional al margen de la tecnología y enfrentándolo a la angustia de adaptarse a este nuevo entorno tecnológico.

Parte del problema surge de la existencia de una diversidad de dominios, debido a que no es tan fácil para un robot desempeñar múltiples funciones dentro de su entorno porque la información del mundo es difusa y poco clara [1]. Por tal razón, ha sido un objeto de estudio que un sistema pueda adaptarse a distintos dominios [2].

Recientes investigaciones abordan la problemática anterior mediante el trabajo de la representación compartida, la cual es una representación compacta e invariante que integra diferentes tipos de modalidades que en este caso un robot social puede adquirir a través de sus sensores [3]. Por ejemplo, para el contexto de una playa se puede hacer referencia a esta por el sonido de las olas, la sensación de la arena o el sonido que producen las gaviotas. Estas distintas modalidades se complementan para construir una representación. Sin embargo, en la Universidad Santo Tomás, de acuerdo con una revisión de trabajos anteriores, esta solución de representaciones compartidas no ha sido estudiada.

Teniendo en consideración las problemáticas anteriormente expuestas, en el presente proyecto se desarrolla la siguiente pregunta problema:

¿Cómo desarrollar un software para la robótica social que le permita asistir a personas con analfabetismo digital en la operación de funciones de domótica a través de las características distribuidas de ROS?

1.2. Objetivo General

Desarrollar un software para la robótica social orientado a la asistencia de personas con analfabetismo digital en la operación de funciones de domótica a través de las características distribuidas de ROS.

1.3. Objetivos Específicos

- Investigar en bases de datos científicas del CRAI de la Universidad Santo Tomás y en repositorios de software abierto sobre temáticas y posibles herramientas tecnológicas a usar que permitan la definición de un estado del arte.
- Diseñar el algoritmo basado en redes neuronales a partir de la estructura de operación ROS, las características tecnológicas y funciones BACS a trabajar con base en los resultados hallados en el estado del arte.
- Desarrollar el software de asistencia mediante la integración y construcción de artefactos de software basados en ROS.
- Validar el software mediante el desarrollo y ejecución de escenarios de prueba con base en la evaluación de métricas de desempeño.

1.4. Justificación

Con la integración entre la robótica social y la domótica, se pueden desarrollar aplicaciones de alto nivel para personas con analfabetismo digital. Asimismo, por su fácil uso hace posible que las personas integren en su diario vivir la tecnología a través de la asistencia brindada por un robot social. Esto reduciría la intervención de estas personas en tareas repetitivas, o que no puedan realizar, mejorando así su calidad de vida.

La implementación de esta solución, dada su intuitiva interfaz, tiene el potencial de incrementar significativamente la interacción de individuos con la tecnología. Este avance no solo disminuiría la conocida "brecha gris.[en] el acceso tecnológico [4], sino también reduciría la disparidad de desarrollo entre Colombia y otras naciones. Según estimaciones, Colombia requeriría al menos una década para equipararse en términos de adopción tecnológica con países como Chile, y aproximadamente tres décadas para alcanzar los niveles de Holanda [Carvajal, 2021].
La adaptación de dominio potencia las aplicaciones que podrían tener los sistemas inteligentes, y esto es posible gracias a las representaciones compartidas. Sin embargo, esta solución no se ha implementado en la comunidad académica, por tal motivo se considera pertinente abordar dicha temática para aportar nuevo conocimiento a la institución.

Al llevar a cabo este proyecto se busca promover investigaciones en la universidad que incorporen las representaciones compartidas y que hagan posible el desarrollo de proyectos futuros por parte de la comunidad enfocados al bienestar de la sociedad.

También se busca construir un avance en temas que no se han trabajado en el laboratorio de robótica, dentro de los cuales también se encuentra a integración del IoT (internet de las cosas) y la robótica e incluso se busca extrapolar estos conocimientos para que se puedan desarrollar aplicaciones de uso libre para personas con discapacidades.

1.5. Impacto Social

Se busca realizar un primer aporte a un enfoque de investigación y desarrollo en la comunidad académica de la Universidad Santo Tomas que implemente las representaciones compartidas, ya que se ha realizado una consulta dentro de las bases de datos de la institución y no se ha encontrado producción académica de esta universidad que las involucre en robótica e IA. Más aún, se busca que los grupos de investigación elaboren futuros estudios con base en la temática anteriormente expuesta.

El acelerado avance de la tecnología no ha tenido una cobertura uniforme en la población como se demuestra en [5], en donde se evidencia que el 20 % de la población no ha implementado ni siquiera en un nivel básico la tecnología en su vida. Este porcentaje se puede reducir y a su vez generar una transformación social al hacer más accesible la tecnología por medio de una herramienta de fácil entendimiento y uso, impactando positivamente en el bienestar las personas.

Este proyecto también es de interés para el personal de asistencia a cargo de un adulto mayor, ya que gran parte de los analfabetas digitales son de edades avanzadas, por ejemplo, en México, el grueso de la población de usuarios de internet está entre edades de 18 a 24 años, mientras que personas de 55 años o más tienen un índice más bajo de acceso y utilización de servicios basados en la web [6].

Sin embargo, este proyecto no solo busca incidir en la vida de las personas con analfabetismo digital. Ya que para cualquier individuo, sin importar su nivel de habilidades tecnológicas, le sería de gran ayuda reducir su intervención en tareas repetitivas, lo cual se puede lograr con la implementación de este software.

Además, este proyecto también busca llevarse a cabo siguiendo los objetivos de desarrollo sostenible, en los cuales se enmarca la universidad, estos son salud y bienestar, crecimiento económico, trabajo decente para personas, reducción de las desigualdades, paz y justicia, y alianzas para cumplir objetivos. Así mismo, el proyecto se encuentra dentro de las líneas de proyección social definidas dentro del documento marco de proyección social de la universidad [7], donde se encuentra la línea de desarrollo comunitario, la cual dentro de sus funciones busca el bienestar de la vida de un grupo de personas las cuales sufren una problemática [8].

Capítulo 2

Estado del Arte

La convergencia entre la IA y robótica ha permeado diversos campos de estudio en los últimos años. Este fenómeno se hace especialmente evidente en la robótica social, un área que ha experimentado un notable crecimiento en donde se busca la coexistencia armoniosa entre robots y seres humanos. En este contexto, el reconocimiento de emociones desempeña un papel crucial, permitiendo una interacción más efectiva entre los robots y las personas, por tal razón es un tema de interés que ha sido profundizado en trabajos como [9], en el que se presenta un sistema que permite a un robot reconocer emociones para después almacenarlas en un repositorio semántico. La primera versión de este proyecto solo reconoce emociones en datasets de S2T basadas en la API de Google y librerías de Python, herramientas con las que se desarrolla una red neuronal que etiqueta en textos las emociones de las personas con base en procesamiento natural del lenguaje (NLP). Cabe aclarar, que también se limitó en esta versión la cantidad de representaciones que una emoción puede tener; el sistema fue probado en un museo donde el robot desempeñaba la función de guía y a su vez reconocía cómo se sentían los visitantes a lo largo del tour, para después, evaluar dichos resultados [9].

Una de las problemáticas que tienen los robots de servicios es la disponibilidad de información de su entorno para generar una estimación más exacta de su estado, lo cual puede convertirse en un gran obstáculo para el robot a la hora de ejecutar alguna tarea, porque se pueden generar falsos positivos o falsos negativos [1]. Se asume por la planeación clásica que el robot tiene información suficiente del mundo para tomar una decisión, pero se sabe que el mundo es incierto y parcialmente observable para los robots de servicio [1]. Por tal razón, se propone un modelo llamado *representaciones compartidas para sugerencias alternativas con información incompleta* el cual propone una toma de decisiones alternativa por parte del robot al momento de identificar información poco clara y cambiante con el fin de encontrar la manera de realizar su tarea.

Dentro del campo de la detección de emociones en [10], se usa el análisis de polaridad de sentimientos para clasificar opiniones en plataformas en linea como positivas, negativas o neutras, para así de manera automática encontrar tendencias de opinión en español. En este estudio, se basan en el enfoque de aprendizaje automático (*Machine Learning*), dado que este no requiere mención directa de la emoción como lo requiere el enfoque basado en palabras clave o en léxico/corpus. Dentro de este enfoque, se encuentran las redes neuronales recurrentes (RNN), las cuales procesan secuencias de datos de grandes dimensiones pero no son aptas para el análisis de sentimientos, a causa de no recordar dependencias a largo plazo por la tendencia del gradiente a desaparecer; por esta razón las RNN son usadas con unidades recurrentes cerradas (GRU por sus siglas en inglés) que mantienen las dependencias a largo plazo. También se encuentran las redes de memoria a corto y largo plazo (LSTM) que son más complejas, y redes neuronales convolucionales (CNN).

Debido a que el contenido utilizado en el análisis de sentimientos es cada vez más multimodal, es decir, no solo incluye texto si no también imágenes, y que estas modalidades se complementan para realizar tareas [11], en [12] proponen un modelo de red neuronal denominado Red de Atención al Aspecto Visual o VistaNet de tres capas, en donde los componentes visuales desempeñan un papel complementario que señala aspectos más destacados dentro de una reseña a los que el modelo les presta especial atención al momento de clasificar sentimientos. En la capa inferior se transforman las palabras en representaciones de frases; en la intermedia, primero se codifican las imágenes de entrada con la red neuronal convolucional preentrenada VGG-16, se transforma la representación de frases en una a nivel de documento utilizando la atención visual en donde se asigna mayor peso a las frases más destacadas; y en la superior, se asigna la etiqueta de sentimiento.

Por otra parte, en [13], se realizó un estudio similar a [10], en donde el objetivo es por medio del procesamiento del lenguaje natural analizar tendencias en Twitter y su enfoque es dar información para la toma de decisiones enfocadas a la Investigación, desarrollo e innovación [13]. Con el objetivo de realizar dicha tarea se emplean distintas herramientas para el análisis de textos, dichas herramientas, tienen cómo especialidad el Procesamiento natural del lenguaje, categorización, etiquetado del texto y razonamiento semántico, además de hacer uso de la técnica *Crawling* para la minería de datos. El resultado de este proyecto fue un sistema capaz de realizar el análisis con el enfoque deseado, facilitando la extracción de información heterogénea y no estructurada [13].

Es fundamental destacar que los sistemas encargados de recolectar y procesar datos del entorno se enfrentan a una realidad compleja. En este contexto, adquirir información no es una tarea sencilla debido a la presencia constante de ruido, objetos que pueden distorsionar la información, así como diferentes contextos y otros factores. Esta complejidad se refleja en la afirmación de [1], quien señala que los robots de servicio pueden experimentar falsos positivos o negativos al adquirir y ejecutar tareas. De manera similar, [3] respalda esta observación al destacar que el contexto no siempre proporciona la información suficiente para que un sistema pueda llevar a cabo una adaptación de dominio efectiva.

Reconocer a qué dominio adaptarse en un contexto real es una tarea desafiante. El único criterio disponible para el sistema es una estimación basada en la entrada, la cual puede estar sujeta a distorsiones. Para abordar esta problemática, se ha propuesto un enfoque que implica la creación de un sistema de representaciones compartidas y heterogéneas para la adaptación de dominio. La mayoría de los sistemas utilizan una adaptación de dominio homogénea [3], es decir, estiman la función objetivo basándose únicamente en la entrada. Por esta razón, se plantea una representación heterogénea que tiene como objetivo propagar información a través de distintos dominios, aprovechando imágenes, palabras clave, contextos y otras características específicas de cada dominio.

El sistema propuesto por [3] tiene como objetivo mejorar e iterar sobre las técnicas existentes basadas en características. Introduce subetiquetas para facilitar la discriminación de dominios y se implementa de manera iterativa, midiendo la consistencia de las etiquetas en todos los dominios. Cabe destacar que este clasificador de dominios opera de manera no supervisada y ha demostrado un rendimiento muy satisfactorio en comparación con otras herramientas similares.

Por otro lado, Google MediaPipe ha desarrollado un *framework* diseñado para mejorar la percepción del entorno, lo que permite el procesamiento de información proveniente de sensores de manera versátil y adaptable [14]. MediaPipe abarca diversas funcionalidades, incluyendo el procesamiento de audio, reconocimiento facial, localización de objetos, entre otros. Esta herramienta se presenta como un recurso valioso en la construcción de sistemas que requieran

procesar información de manera multimodal. Por ejemplo, en la modalidad de audio, se pueden emplear redes S2T como Vosk [15], mientras que para la modalidad de imagen, se puede aprovechar la potencia de herramientas como YOLO, que permiten la identificación y ubicación precisa de objetos. Este enfoque multimodal ofrece un mayor potencial para la comprensión y análisis de datos para diversas aplicaciones.

Por otra parte, en [2], un estudio previo relacionado con la adaptación de dominios, proponen las representaciones compartidas en grupos escasos para solucionar la adaptación de dominios. Por consiguiente, se desarrolla un diccionario de representaciones compartidas en grupos escasos para entrenar un clasificador cuyo propósito es acertar con base en la información específica del sistema y la del diccionario. La gran ventaja de este sistema es que se elimina el error por estimación y al hacer este sistema no supervisado se adicionan ventajas cómo el ahorro del tiempo y recursos, a razón de que no se tiene que anotar el dominio objetivo. Al ser de grupos escasos se puede reconocer información con muy pocos datos. Por tal razón, se obtienen resultados de reconocimiento de caras muy superiores en comparación con otras herramientas.

Según [11], las representaciones compartidas son representaciones compactas y de modalidad invariable que integran diferentes modalidades sin etiquetar. Los desafíos del aprendizaje multimodal son inferir representaciones de la integración de modalidades para generar modalidades a partir de ella y la generación intermodal que es la traducción de datos entre modalidades; es necesario tener en cuenta la naturaleza heterogénea de la información adquirida, es decir, que tienen diferentes espacios de características y distribuciones. Estos desafíos se pueden afrontar con el uso de los codificadores automáticos variacionales, que son algoritmos basados en modelos generativos profundos los cuales asumen los datos como estocásticos provenientes de la representación compartida vista como una variable global. Estos modelos generativos profundos a diferencia de los convencionales manejan datos de grandes dimensiones y usan redes neuronales profundas con aprendizaje de extremo a extremo con el uso de *backpropagation*. Una buena representación logra ejecutar diferentes tareas a partir de la entrada original, preferiblemente de manera no supervisada.

Para construir las representaciones compartidas, en [16], proponen el mecanismo de *superposición*, aplicado en un entorno virtual a 1 de 2 agentes, el agente 1 no posee información que permita hacer distinción previa entre ellos. A través de entrenamientos en los cuales el agente 2 se mueve aleatoriamente, el agente 1 logra desarrollar gradualmente dos representaciones diferentes pero compartidas de las localizaciones espaciales del agente 1 y 2 por medio del aprendizaje predictivo recibiendo sensaciones visuales y motoras de él mismo. Las sensaciones de entrada codificadas, se propagan a través de dos vías de procesos diferentes con memorias independientes en un mismo módulo compartido, que es una red neuronal recurrente con una memoria a largo plazo (LSTM), que puede retener recuerdos, generando simultáneamente dos interpretaciones diferentes de una sola sensación usando la misma forma de procesamiento, un módulo posterior (Predictor Visual) integra las dos salidas del módulo compartido de forma mixta para predecir futuras sensaciones visuales.

En [17], entrenan un modelo no supervisado que segmenta imágenes multimodales 2D y 3D a través de la adaptación de dominios (LMISA) para aplicaciones médicas. Los modelos de los estudios clínicos se construyen a partir de la obtención de datos de distintas modalidades que tienen la capacidad de manejar cambios de dominio. Se cuenta con las anotaciones de imagen del origen para el entrenamiento. Aplican el método de *magnitud de gradiente normalizado localmente* (MLNGM) para el preprocesamiento, después una red codificadora-decodificadora aprende una representación conjunta de las imágenes de ambos dominios a la vez e implementan una restricción de forma en el aprendizaje adversarial, con esto se logra una red eficiente de adaptación de dominio de extremo a extremo. Para el origen, se usó un conjunto de datos públicos de KiTS19 y para el destino datos locales de RM para la

segmentación del riñón.

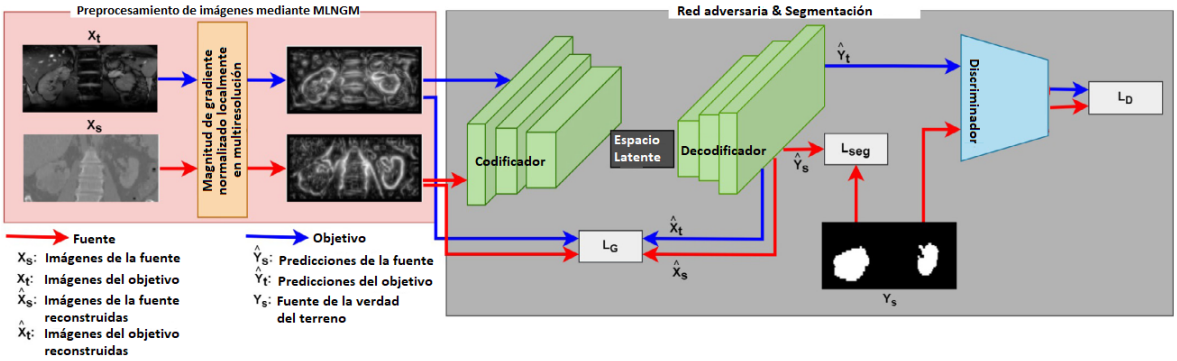


Figura 2.1: Esquema general de LMISA traducido del original. Fuente: [17].

Los robots de servicio tienen aplicaciones en entretenimiento, asistencia, seguridad, vigilancia, educación y ayuda en tareas domésticas [18]. Otra área relacionada con el entorno doméstico es la domótica, que permite automatizar una vivienda a través de un conjunto de sistemas [19], por ejemplo, la programación del apagado y encendido de las luces de acuerdo a un horario o por detección de movimiento. A gran escala se encuentra la inmótica, que funciona de manera similar a la domótica y usa los sistemas para el control y automatización de edificios (BAC), es decir, automatiza y conecta los elementos del inmueble a través de protocolos de comunicación como BACnet, el cual está estandarizado bajo la normativa ISO 16484-5 [20], la cual integra un listado de funciones BACS que ayudan a determinar la funcionalidad de una planta.

Dicho lo anterior, es evidente que existen diversas aplicaciones para la implementación de robots de servicio y sistemas de automatización. Sin embargo, se reconoce que en entornos no controlados, estas tareas pueden volverse especialmente desafiantes, sobre todo cuando implican el Procesamiento del Lenguaje Natural (NLP). A pesar de estos desafíos, se han logrado avances significativos en el campo de NLP. Por ejemplo, Meta AI [21] ha desarrollado recientemente un *Large Language Model* (LLM) con una impresionante gama de 7 a 70 billones de parámetros. Este sistema es altamente competente en la interacción con el lenguaje humano, lo que le permite sostener conversaciones, responder preguntas, operar en varios idiomas, abordar problemas y más [21]. Además, OpenAI [22] ha desarrollado un LLM que va más allá de los logros de Touvron. Este sistema no solo es capaz de procesar imágenes, sino que también puede generar respuestas y opiniones sobre las imágenes. Incluso tiene la capacidad de identificar aspectos humorísticos o absurdos en las imágenes, relacionando la falta de sentido con el humor [22].

Capítulo 3

Marco teórico

3.1. Arquitecturas de redes neuronales

3.1.1. Redes Neuronales Convolucionales (CNN)

Las CNN han demostrado ser altamente eficientes en el procesamiento de imágenes y caracteres. Estas redes son entrenadas previamente con extensos conjuntos de datos para adquirir la capacidad de identificar diversos objetos en una imagen [23].

Para el correcto funcionamiento de una red CNN se necesita definir los datos que se reciben en la capa de entrada, teniendo esta distintas neuronas según la cantidad de píxeles que tenga una imagen, considerando aspectos como la resolución y el tipo de imagen (blanco y negro o a color).[23]. Como parte de la preparación de los datos, se realiza la crucial normalización de los valores de los píxeles, escalandolos al rango de 0 a 1. Esta etapa es esencial para lograr una entrada coherente y efectiva a la red neuronal.

na vez establecida la configuración inicial, , los datos son procesados por diferentes capas posteriores llamadas capas de convolución [24]. El objetivo de las capas convolucionales es extraer en matrices grupos de píxeles característicos de una imagen. Estos se obtienen con una función de activación como la *Rectifier Linear Unit* (RELU) en una matriz con dimensiones cada vez más reducidas, llamada Kernel. Este proceso se conoce como convolución [24], es cíclico y se repite tantas veces como sea posible contra la convolución anterior en cada capa convolucional hasta alcanzar el tamaño más pequeño posible respecto a la imagen original [25].

Este proceso se lleva a cabo para que se puedan identificar y almacenar solo características y patrones más importantes de la imagen, ahorrando memoria y costo computacional. La figura 3.1, enseña de una manera gráfica la arquitectura básica de una red CNN.

$$S(i,j) = (I * K)(i,j) = \sum_{n=m} \sum_{n=n} I(m,n)K(i-m, j-n) \tag{3.1}$$

La ecuación 3.1 habla acerca de la convolución, donde I es la imagen de entrada que a su vez es una matriz y K es la matriz kernel, K tiene dimensiones *(i, j)* e I tiene dimensiones *(m, n)*. Ya que se le hace convolución a una imagen, esta es tiempo discreto, por tal motivo se usa una sumatoria y no una integral.

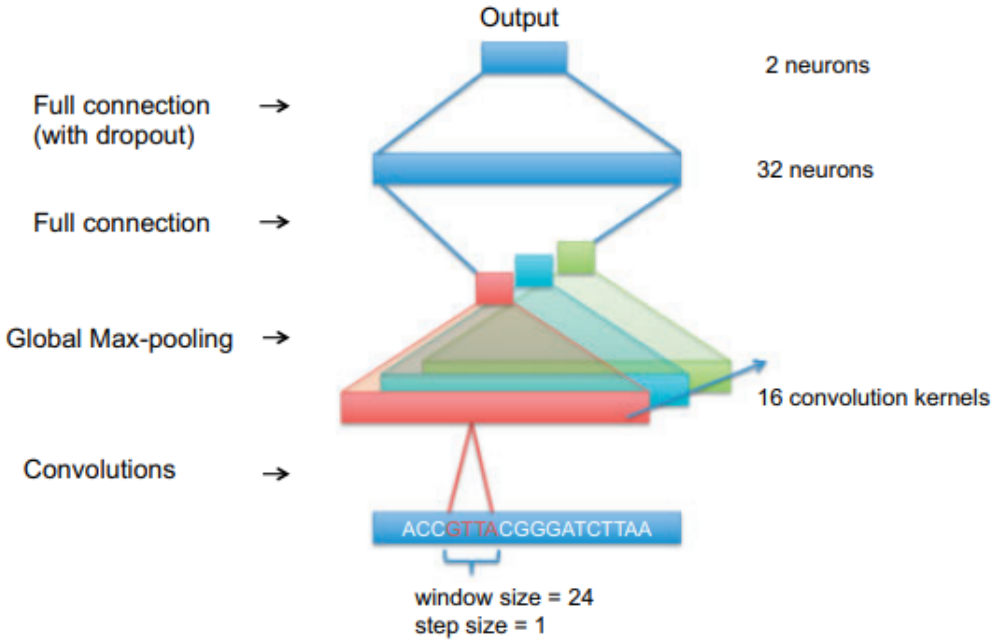


Figura 3.1: Red neuronal CNN. Fuente: [26].

3.1.2. Redes Neuronales Recurrentes (RNN)

Las RNN son herramientas altamente efectivas para procesar secuencias de datos debido a sus conexiones retroactivas, lo que les permite mantener y utilizar información de entradas anteriores en la secuencia. En su estructura, se maneja una secuencia de entrada X_(t):X_(1),X_(2),...X_(T) en un instante t, y una salida Y_(t) en el mismo momento. En cada paso, la red utiliza una combinación lineal de la secuencia y la activación anterior h_(t-1) como entrada, y produce tanto una predicción actual como una activación actual h_(t), la cual se conoce como el estado oculto y es la "memoria"que se utiliza como entrada para la capa recurrente en la siguiente iteración [27].

Las conexiones de retroalimentación forman ciclos dentro de la red (ver Figura 3.1). Estas conexiones son fundamentales para la actualización de los pesos W_(hh),W_(xy) y W_(hy) durante el proceso de entrenamiento. En términos generales, cada red posee su propia matriz de pesos y sesgos, junto con su respectiva función de activación, como se ilustra en las siguientes ecuaciones [28].

$$h_t = f_w(h_{t-1}, x_{(t)}) \tag{3.2}$$

$$h_t = tanh(W_{hh}^T h_{t-1} + W_{xh}^T X_t) \tag{3.3}$$

$$Y_t = W_{hy} h_t .... \tag{3.4}$$

Esta arquitectura presenta la compartición de parámetros entre los distintos módulos de la red, lo cual es crucial cuando una información específica puede ocurrir en múltiples posiciones dentro de una secuencia [10]. Esta característica mejora la generalización y el rendimiento en datos de prueba con patrones no vistos en los datos de entrenamiento.

Por lo tanto, resulta muy útil en aplicaciones como el procesamiento del lenguaje natural (NLP), la traducción automática, el reconocimiento de voz y la generación de texto, entre otros. Aunque estas redes funcionan bien en

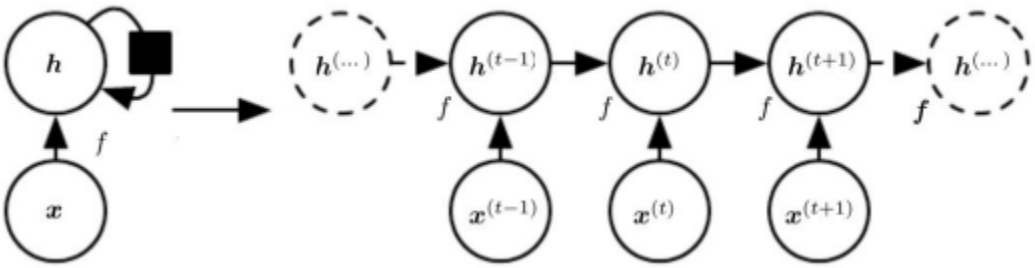


Figura 3.2: Red neuronal recurrente. Fuente: [28].

teoría, se enfrentan al desafío del desvanecimiento del gradiente, donde este disminuye rápidamente al calcularse con respecto a entradas de varios pasos en el pasado, limitando la capacidad para modelar dependencias a largo plazo en secuencias de datos [10].

Para abordar este problema, surge un tipo de celda de memoria más compleja llamada Memorias de Corto y Largo Plazo (LSTM), diseñada para extraer patrones de secuencias de mayor longitud [28].

3.1.3. Redes Memoria de largo y corto plazo (LSTM)

Las redes LSTM se destacan por su capacidad para procesar secuencias temporales y superar el desafío de la desaparición del gradiente, común en las RNN. La característica distintiva radica en la incorporación de una celda de memoria adicional que permite un flujo ininterrumpido de información a lo largo de la red.

En este contexto, la información fluye a través de tres compuertas fundamentales, cada una actuando como una red neuronal independiente. La compuerta de entrada decide qué nueva información se integra a la celda de memoria, la compuerta de olvido determina qué datos deben ser descartados de la celda, y la compuerta de salida determina qué información de la celda se utilizará como salida [29]. Por lo que cuentan con la capacidad de discernir y preservar información crucial a lo largo del tiempo, gracias a sus unidades de memoria. Al mismo tiempo, tienen la capacidad de descartar información superflua, permitiendo así mantener dependencias de largo plazo en secuencias temporales.

Su versatilidad las convierte en herramientas fundamentales en tareas como traducción automática y generación de texto. Además, en el reconocimiento de voz, las LSTM destacan por su habilidad para capturar patrones temporales en señales de audio. En el ámbito financiero y la predicción de series temporales, las LSTM demuestran su eficacia al proporcionar pronósticos precisos al comprender patrones históricos [29]. La figura 3.3 muestra mejor la arquitectura de una red LSTM, donde los valores de C son los correspondientes a la memoria, H representa las salidas y X las entradas. Las celdas en amarillo son redes neuronales simples con sus funciones de activación y los puntos rosa son operaciones específicas para descartar, actualizar y calcular el nuevo estado oculto de la memoria.

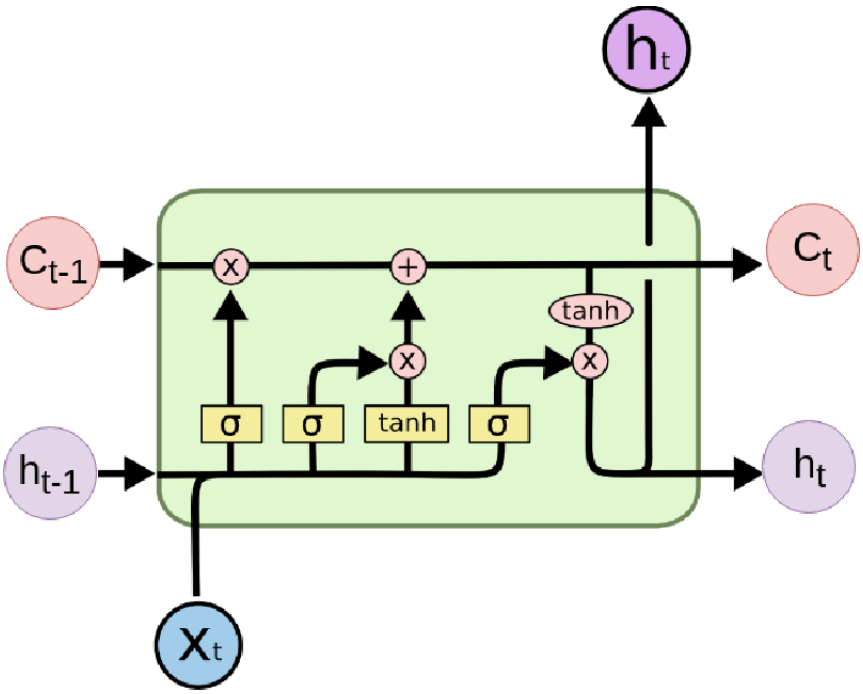


Figura 3.3: Celda de una red neuronal recurrente LSTM. Fuente: [30].

3.2. Procesamiento de Lenguaje Natural (NLP)

El NLP es un campo que se centra en capacitar a las máquinas para comprender y procesar el lenguaje humano de manera efectiva. En este espacio, exploraremos conceptos clave como la transferencia de aprendizaje, transformadores y otros temas relevantes que impulsan los avances en la interacción entre humanos y sistemas computacionales.

3.2.1. Transferencia de aprendizaje

La transferencia de aprendizaje es una técnica en el campo del machine learning que consiste en aprovechar el conocimiento adquirido por un modelo entrenado en una tarea para aplicarlo en otra tarea relacionada pero diferente. Este enfoque permite reducir los costos de entrenamiento y facilita la utilización de modelos avanzados sin la necesidad de iniciar el entrenamiento desde cero. El *Fine-tuning* es la etapa clave de este proceso de transferencia. Consiste en ajustar los pesos del modelo preentrenado para que se adapten a las particularidades del conjunto de datos y la tarea específica de la nueva tarea [31].

Existen diversas formas de llevar a cabo el proceso de ajuste fino según Hugging Face. Una de estas metodologías implica utilizar la librería Transformers junto con su módulo Trainer, que proporciona una interfaz optimizada y facilita el proceso al encargarse de tareas de entrenamiento como el manejo de épocas, cálculo de gradientes y administración de recursos de manera automática. Otra alternativa emplea TensorFlow junto con su interfaz Keras para llevar a cabo el *Fine-tuning* de modelos preentrenados. Esto brinda flexibilidad en la configuración del modelo y otorga un mayor control sobre el proceso de entrenamiento. Por último, se puede optar por utilizar PyTorch, un popular marco de trabajo de aprendizaje profundo. Esta alternativa permite un nivel máximo de control y personalización en el proceso de *Fine-tuning*, ya que se trabaja directamente con los tensores y operaciones fundamentales de PyTorch [32].

3.2.2. Transformadores

Los transformadores son una arquitectura de red neuronal desarrollada por Google que ha demostrado un rendimiento excepcional en una amplia gama de tareas de NLP. Esta arquitectura se basa en el mecanismo de atención, que permite a la red procesar secuencias de texto de manera paralela y capturar relaciones de largo alcance entre palabras. Son tan poderosos que pueden hacer una supervisión automática a sí mismos y recibir billones de parámetros nuevos sin etiquetar y aprender de ellos.

Sin duda los transformadores han impactado el mundo y la forma en la que se desarrollan aplicaciones, pues gracias a ellos, se hace más corto el tiempo de desarrollo de ciertos proyectos, debido a su gran versatilidad. En consecuencia, para los desarrolladores los transformadores son herramientas idóneas ya que los existentes como por ejemplo en [21, 22] tienen modelos que pueden ser adaptados según las necesidades de el desarrollador.

Cabe mencionar, la importancia que tiene la comunidad en la continua mejora de los transformadores y de los innumerables usos que esta le da. Modelos como el desarrollado por Meta AI (Llama 2) son de código abierto y varias versiones del modelo son de uso inclusive comercial [21].

3.2.3. Tokenización

La tokenización es un proceso que es muy recomendado a la hora de procesar lenguaje, pues lo que hace la tokenización es asignar valores numéricos a palabras. Esto, en consecuencia, reduce el tamaño de la información a procesar.Gracias a este proceso, se pueden preparar datos para el posterior entrenamiento de modelos de NLP en gran volumen. En consecuencia, se pueden generar corpus mucho más enriquecidos y robustos.

3.2.4. Redes neuronales para NLP

Las RNN y CNN son comúnmente usadas en el NLP, pues condichas redes se puede procesar audio e imagenes y hacer reconocimiento y procesamiento de elementos del lenguaje de los humanos. Se pueden combinar de muchas formas, creando modelos híbridos y multimodlaes, ampliando mucho más la prescepción de los sistemas.

3.3. Características de la información

3.3.1. Información heterogénea

La información heterogénea se refiere a datos que tienen diferentes espacios de características y distribuciones. Esta información puede provenir por ejemplo de texto, imágenes, audio, vídeo, entre otros, denominadas modalidades. La heterogeneidad dificulta el procesamiento ya que cada modalidad puede requerir técnicas de procesamiento y análisis diferentes para extraer información útil [11].

3.3.2. Información no estructurada

La información no estructurada se refiere a datos que no tienen una estructura definida y organizada, como la que se encuentra en bases de datos relacionales o en hojas de cálculo, la mayoría de aplicaciones usan información estructurada [33]. Estos datos pueden ser de diferentes tipos, como texto, imágenes, vídeos, archivos de audio, etc. Al

igual que con la información heterogénea, el análisis de la información no estructurada requiere técnicas avanzadas de procesamiento del lenguaje natural, como lo sugiere [33], cuando se habla de la tokenización.

3.3.3. Dominios

En el ámbito del Machine Learning, un "dominio"se refiere a un área o tema específico que un agente inteligente puede abordar, como la visión artificial, el procesamiento del lenguaje natural, la recomendación de películas, la clasificación de imágenes médicas, la detección temprana de enfermedades [34], la automatización de edificios, entre otros [1]. Es importante destacar que para los sistemas de Machine Learning, abordar múltiples dominios simultáneamente no es una tarea sencilla, ya que implica un gran gasto de recursos computacionales. Sin embargo, gracias a desarrollos más recientes como el presentado por [21], se ha observado una tendencia hacia la mejora constante de la capacidad multimodal.

3.4. Representaciones compartidas

Una representación compartida se refiere a la creación y utilización de una representación común o de características extraídas de información de naturaleza heterogénea, con el propósito de emplearlas en múltiples tareas o dominios diferentes [3]. En este enfoque, una vez que la información ingresa en la capa de entrada, otra capa de procesamiento toma esos mismos datos y genera una representación general de dicha información [11]. Sin embargo, es importante destacar que no todas las arquitecturas de red utilizan una capa específica para generar la representación compartida. En algunos casos, la representación compartida puede generarse mediante una combinación de capas específicas que procesan los datos de entrada de manera diferente [24].

3.5. Protocolo BACnet

El Protocolo BACnet se centra en la automatización de edificios para controlar dispositivos como sistemas de iluminación, calefacción y refrigeración [35]. Es compatible con varios protocolos de comunicación estándar, como Ethernet. En la estructura de BACnet, los dispositivos tipo BACnet contienen objetos con propiedades que pueden ser leídas y escritas por otros dispositivos. Cada dispositivo tiene un nombre, una dirección con puerto, y los objetos también tienen nombres asociados.

3.6. Métricas del proceso de entrenamiento y evaluación de los modelos

Las métricas son herramientas fundamentales para evaluar el desempeño de los modelos durante su proceso de entrenamiento y evaluación. En el contexto del entrenamiento de modelos de NLP, como los que se someten a *Fine-tuning*, varias métricas son cruciales para entender cómo se está comportando el modelo.

La métrica *"global_step"* representa el número de actualizaciones de parámetros del modelo durante el entrenamiento. *"training_loss"* indica cuán precisa es la predicción del modelo, donde valores bajos indican un buen rendimiento. El *"train_untime"* denota el tiempo total de entrenamiento, y *"train_samples_per _second"* muestra la velocidad de procesamiento de datos por segundo. Además, *"trains_steps_per_second"* refleja la velocidad a la que se realizan pasos de entrenamiento por segundo. *"total_flops"* representa el número total de operaciones en punto flotante realizadas durante el entrenamiento, lo que indica la carga computacional y "*epoch*"muestra cuántas veces se recorrió

el conjunto de entrenamiento.

Por otro lado, en el caso de las pruebas, es común utilizar métricas como el tiempo de respuesta y la tasa de éxito. El tiempo de respuesta se refiere al tiempo que tarda el modelo en procesar una entrada y proporcionar una salida. La tasa de éxito indica la proporción de predicciones correctas realizadas por el modelo en un conjunto de datos de prueba.

3.6.1. Matriz de confusión

3.7. Librerías, repositorios y herramientas tecnológicas

3.7.1. Tensorflow

Es una plataforma de aprendizaje profundo de código abierto desarrollada por Google. Permite la creación de modelos de aprendizaje profundo utilizando redes neuronales, se centra más en la ejecución que en los detalles de implementación y se puede utilizar para una amplia gama de aplicaciones de IA, como la clasificación de imágenes y la detección de objetos [36]. Incluye varios algoritmos de optimización para acelerar el entrenamiento de modelos de aprendizaje profundo, como el descenso de gradiente estocástico y la retropropagación.

3.7.2. Python

Python es un lenguaje de programación que tiene muchas librerías populares para el aprendizaje profundo, como TensorFlow, PyTorch, Keras y scikit-learn, que son fáciles de usar y tienen una amplia comunidad de usuarios, es flexible y puede ser utilizado para una amplia variedad de tareas, incluyendo la visualización de datos, el procesamiento de imágenes y el procesamiento del lenguaje natural [37].

3.7.3. Robot Operating System (ROS)

ROS es un meta-sistema operativo que proporciona una gama de servicios como abstracción de hardware, control de dispositivos de bajo nivel, implementación de funcionalidad comúnmente utilizada, comunicación de mensajes entre procesos y gestión de paquetes. También proporciona herramientas y librerías para el desarrollo de software en robótica [38].

ROS 1 se lanzó en 2010 y está diseñado para ser modular y escalable, lo que permite a los usuarios crear aplicaciones de robótica complejas mediante la integración de paquetes de software preconstruidos y personalizados. Sin embargo, ROS 1 tiene algunas limitaciones, como problemas de escalabilidad y seguridad, lo que hace que sea difícil de usar en sistemas grandes y complejos [39].

ROS 2 se lanzó en 2015 con el objetivo de abordar las limitaciones de ROS 1 y proporcionar un sistema más robusto y escalable.[39] ROS 2 es compatible con múltiples sistemas operativos, lo que lo hace más versátil y fácil de integrar en diferentes plataformas de hardware. También ofrece mejoras en la seguridad, la fiabilidad y la escalabilidad, lo que lo hace más adecuado para aplicaciones de robótica industrial [40].

3.7.4. Scikit-learn

Scikit-learn es una librería de aprendizaje automático de código abierto para el lenguaje de programación Python. Proporciona herramientas simples y eficientes para el análisis de datos y así como para la construcción de modelos de aprendizaje automático y la evaluación de su rendimiento [41].

3.7.5. Hugging Face

Hugging Face es una plataforma destacada en el ámbito del NLP. Su librería *Transformers* ofrece acceso a modelos preentrenados basados en *transformers* como GPT (Generative Pre-trained Transformer) y BERT (Bidirectional Encoder Representations from Transformers), así como herramientas para el ajuste fino y la personalización de estos modelos para aplicaciones específicas Esta metodología facilita tareas de entrenamiento como el manejo de épocas, cálculo de gradientes y administración de recursos de manera automática [32].

3.7.6. BAC0

BAC0 es una librería de Python basada en *BACpypes* que facilita el uso del protocolo BACnet para procesar mensajes y crear redes BACnet en una red. Ofrece una capa de conexión entre diferentes puntos. Además de la capa de conexión, esta librería permite la creación de dispositivos y objetos, la identificación de elementos en la red, y la escritura y lectura de propiedades en los objetos, entre otras funcionalidades [42].

3.7.7. Vosk

Vosk es un conjunto de herramientas de reconocimiento de voz de código abierto y acceso gratuito que aprovecha modelos acústicos y de lenguaje basados en redes neuronales profundas para llevar a cabo el reconocimiento de voz sin requerir una conexión en linea en más de veinte idiomas, incluyendo español, inglés, alemán y ruso. Vosk se puede usar con Python y otros lenguajes de programación, además, ofrece modelos preentrenados para diferentes dominios y tamaños [15].

3.7.8. Yolo

YOLO es un avanzado algoritmo basado en redes neuronales convolucionales diseñado para la detección de objetos en imágenes en tiempo real. Basicamente divide la imagen en una cuadrícula, y cada celda puede predecir múltiples cajas delimitadoras y sus correspondientes probabilidades de clase . A través de sus diversas versiones, como YOLO V2, V3, V4 y V5, se han realizado mejoras significativas en términos de precisión y velocidad. YOLO es ampliamente utilizado en aplicaciones de visión por computadora [43].

Capítulo 4

Diseño Metodológico

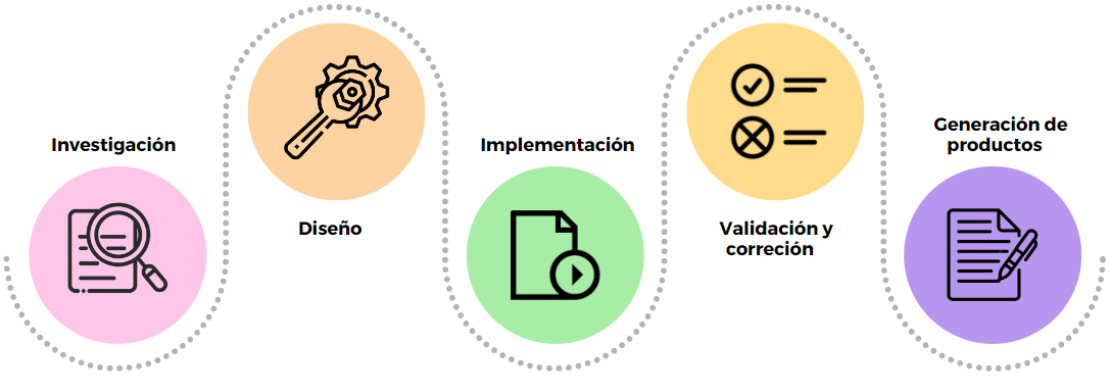


Figura 4.1: Metodología del proyecto. Fuente: autores.

Como se puede observar en la Figura 4.1, se propone el desarrollo de la metodología en 5 fases, las cuales se enuncian a continuación.

4.1. Revisión bibliográfica

Inicialmente se realiza una revisión de diferentes recursos bibliográficos en bases de datos científicas del CRAI de la Universidad Santo Tomás y en repositorios de software abierto de naturaleza similar como GitHub, GitLab, los cuales contienen información relevante sobre representaciones compartidas, adaptación de dominios, información multimodal, técnicas de NLP, recursos abiertos para implementar dichas técnicas y herramientas que permiten desarrollar el proyecto. Se identifican diferentes elementos de los artículos que permitan comparar técnicas utilizadas, herramientas de software y hardware, entre otros, lo cual permitirá la construcción del estado del arte relacionado.

4.2. Diseño del software de asistencia

Tras realizar el proceso de investigación, se define un listado de funciones BACS con base en la norma ISO 16484-3. De igual manera, en el diseño del software de asistencia de acuerdo a sus necesidades se elige la arquitectura de red neuronal, las herramientas de software entre las que están los lenguajes de programación como: Python o C++; IDE´s como: Atom, Spyder y PyCharm, soportadas por el sistema operativo en el cual se trabajará. Se elige la base de datos que se va a utilizar para el entrenamiento y de ser necesario se realiza el acondicionamiento del

mismo, Adicionalmente se establecen las métricas de desempeño con las cuales se evaluará el sistema, entre las que se encuentran: la matriz de confusión el tiempo de respuesta, la tasa de éxito, la precisión y la exactitud. Conforme avance el proyecto es posible que estas métricas se modifiquen.

4.3. Implementación del software de asistencia

Una vez establecida la arquitectura de la red neuronal, esta es entrenada de forma iterativa hasta que se logre el ajuste de los pesos de cada capa, con el fin de minimizar el error en la salida respecto al resultado esperado. Luego, por medio del software ROS, la IDE, el lenguaje de programación y herramientas de desarrollo como Keras y TensorFlow, se integra la red con las dos partes del sistema: el software que permite la interacción de la persona con la maquina de manera gráfica y el hardware, con el que se realiza principalmente la recolección de información, hasta que se logre una correcta comunicación entre ambas partes. En esta etapa, de ser necesario se sigue investigando para complementar los conocimientos relacionados con la integración de las mismas.

4.4. Evaluación de desempeño del sistema

Posteriormente, se valida el funcionamiento del sistema por medio de diferentes escenarios de prueba, en los cuales se busca evaluar mediante las métricas de desempeño su comportamiento ante distintos contextos, y que cumpla puntualmente con cada una de las funciones BACS definidas para el mismo. En caso de que la evaluación de desempeño esté por debajo de lo esperado, se realiza una retroalimentación para hacer los ajustes respectivos en las distintas etapas del sistema.

4.5. Escritura del documento y generación de productos

Paralelo al desarrollo de cada etapa, se documenta el avance de manera detallada en una bitácora, para después redactar el documento final con base en los datos anotados. De esta manera, se tendrán como productos finales el documento, y productos asociados al mismo como: el soporte lógico y la aplicación informática.

Capítulo 5

Desarrollo Conceptual

5.1. Síntesis del estado del arte

Los artículos presentados en las tablas 5.2 y 5.4 son el resultado de la consulta documentada en el estado del arte, tal como se describe en el capítulo 2. El contenido de estas tablas constituye una síntesis en la que se han seleccionado los artículos más relevantes para el proyecto.

La selección de artículos utilizados para la construcción del estado del arte se basa en estadísticas visibles sobre las temáticas abordadas en cada uno de los artículos, lo que evidencia su relevancia actual con respecto a una temática principal: el Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés). Se buscaron palabras clave relacionadas con los temas que engloba el proyecto, y esto se refleja en la figura 5.1, que muestra una clara tendencia en los últimos 5 años.

En el período de tiempo analizado en la figura 5.1, se observa una marcada tendencia en lo que respecta al NLP, seguido de los modelos de lenguaje, la multimodalidad, el *Fine-tuning* y, en último lugar, las representaciones compartidas. En comparación con las dos primeras tendencias, se aprecia un crecimiento casi paralelo en los primeros 5 años. La multimodalidad ha experimentado un crecimiento similar, aunque no tan pronunciado como las dos primeras tendencias. El *Fine-tuning* ha experimentado un aumento reciente, lo que se atribuye a investigaciones actuales relacionadas con los modelos de lenguaje, que también muestran un pico en el mismo período, al igual que la multimodalidad. Esto se debe en gran medida a trabajos como el de [22], un modelo de lenguaje grande y potente, que está estrechamente relacionado con la práctica del *Fine-tuning*.

Por otro lado, las representaciones compartidas no han tenido tanta visibilidad a lo largo del tiempo, ya que se trata de una temática más reciente y menos explorada. No obstante, ha sido abordada en numerosas ocasiones por investigaciones, como las de [1], [2] o [11].

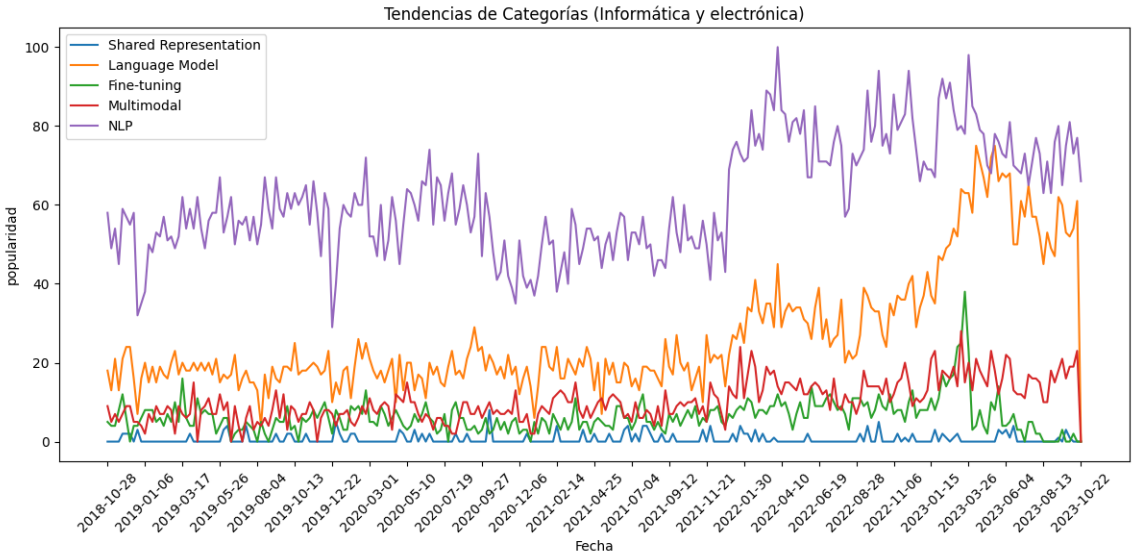


Figura 5.1: Tendencias. Fuente: Google trends.

El primer artículo listado en la tabla 5.2 ofrece una sólida orientación para el desarrollo del proyecto. En este artículo, se presenta una idea general de lo que implica una representación compartida y cómo funciona. Además, se justifica la utilidad de las representaciones compartidas en el contexto del desarrollo de sistemas robóticos inteligentes, abordando las limitaciones a las que se enfrenta un agente en un entorno real debido a su conocimiento parcial del mundo. Se parte del supuesto de que los agentes llevan a cabo acciones en condiciones ideales, con un conocimiento completo de su entorno. Sin embargo, ¿qué sucede si dicho conocimiento es incompleto? ¿Cómo puede un agente realizar su tarea sin incurrir en falsos negativos o falsos positivos? Por esta razón, se plantea la posibilidad de utilizar información complementaria de otras modalidades relacionadas con la ejecución de la función.

El siguiente artículo, que se encuentra en la tabla 5.2, se enfoca en temas relacionados al Procesamiento del Lenguaje Natural (NLP). Este artículo introduce el concepto de clasificación, específicamente en lo que se refiere a la asignación de etiquetas a sentimientos. En el contexto de este proyecto, se realiza una clasificación de dispositivos y niveles en función de las instrucciones proporcionadas por un usuario.

El tercer artículo listado en la tabla 5.2 discute sobre las distintas modalidades que un sistema puede abordar y los desafíos que enfrenta al generar respuestas precisas en presencia de múltiples modalidades de información heterogénea. Además, este artículo aboga por la necesidad de que los sistemas robóticos que operan en entornos del mundo real integren información multimodal. Se plantea la idea de crear un espacio común donde estas modalidades converjan y den lugar a una representación compartida que contemple todas las modalidades.

El primer artículo de la tabla 5.4 aborda la temática del uso de Grandes Modelos del Lenguaje (LLM) para el procesamiento del lenguaje natural. A raíz de esto, surge la idea de utilizar uno de estos modelos de lenguaje a gran escala para procesar las solicitudes del usuario y proporcionar una respuesta. Cabe destacar que también se resalta en el artículo el impacto en cuanto a recursos que implica el entrenamiento de estos modelos de lenguaje, lo que plantea la posibilidad de considerar modelos que requieran menos recursos, y que se ajusten a las capacidades computacionales del proyecto.

El segundo artículo de la tabla 5.4 abre la posibilidad de utilizar redes preentrenadas y ajustar parcial o total

de los pesos de las capas en las arquitecturas de red ya existentes para adaptarlas a necesidades específicas. Este proceso se conoce como "*Fine-tuning*.º ajuste fino, lo cual permite aprovechar un modelo existente que funciona para desempeñar una tarea diferente. En el caso del proyecto, se utiliza la técnica del "*Fine-tuning*"para una tarea de clasificación muy específica.

El tercer y último artículo de la tabla 5.4 introduce la idea de clasificación multimodal. En este caso, se consideran tanto las modalidades visuales como las textuales para evaluar la calidad de un plato de comida. Esta fue la manera en la que se prueba de la convergencia de las modalidades para dar un resultado. El artículo incluye una reseña textual del plato y una imagen del mismo para hacer el test. Luego, se asignan pesos a ciertas palabras de la reseña y a aspectos visuales de la imagen del platillo. Finalmente, se combinan estas modalidades para proporcionar un resultado cuantitativo de la calidad del platillo.

Título	**Autor**	**Año**	**Abstract**	**Producto**	**Referencia**
Shared representations of actions for alternative suggestion with incomplete information	G. H. Lim	2019	Se desarrolla un sistema capaz de extender la capacidad de un robot al realizar un problema, pues la idea propuesta es que en un entorno cambiante, impredecible y poco informativo el robot busque la manera de llevar a cabo su función. Por eso el sistema se llama representaciones compartidas de acciones sugeridas con poca información.	Un sistema capaz de hacer que un robot identifique sus funciones aún en ambientes poco favorables y en constante cambio.	G.H.Lim,«Shared representations of actions for alternative suggestion with incomplete information,» Robotics and Autonomous Systems,vol.116,págs.38-50,jun.2019
Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology	W. Graterol, J. Diaz-Amado, Y. Cardinale, I. Dongo, E. Lopes-Silva y C. Santos-Libarino	2021	Con el procesamiento del lenguaje natural se buscó dotar a un robot social la capacidad de identificar los sentimientos de la gente por medio del habla de la misma, para así convertirlos en texto y procesarlos. El sistema fue testeado en un museo.	Un sistema capaz de reconocer emociones humanas y almacenarlas en un repositorio semántico para interpretarlas a un nivel más profundo.	W. Graterol, J. Diaz-Amado, Y. Cardinale, I. Dongo, E. Lopes-Silva y C. Santos-Libarino, «Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology,» en, Sensors (Basel), vol. 21, n.o 4, feb. de 2021.
A survey of multimodal deep generative models	M. Suzuki e Y. Matsuo	2022	Se abordan los desafíos del aprendizaje multimodal, a través del estudio de diferentes modelos generativos profundos multimodales, en especial los autoencoders variacionales.	Un estudio sobre los diferentes modelos generativos profundos.	M. Suzuki e Y. Matsuo, «A survey of multimodal deep generative models,» Advanced Robotics, vol. 36, n.o 5-6, págs. 261-278, 2022, ISSN: 15685535. DOI: 10.1080/01691864.2022.2035253. arXiv: 2207.02127.

Tabla 5.2: Tabla del estado del arte parte uno

Título	Autor	Año	Abstract	Producto	Referencia
Llama 2: Open Foundation and Fine-Tuned Chat Models	GenAI, Meta	2023	Este estudio presenta Llama 2, una serie de modelos de lenguaje preentrenados y afinados, con escalas de 7 mil millones a 70 mil millones de parámetros. Los modelos afinados, Llama 2-Chat, se centran en diálogos y superan a modelos de código abierto en la mayoría de las pruebas de referencia. Evaluaciones humanas respaldan su utilidad y seguridad, considerándolos sustitutos viables de modelos privados. Ofrecemos una descripción detallada de nuestro enfoque de ajuste fino y mejoras en la seguridad de Llama 2-Chat para fomentar el desarrollo colaborativo y responsable en la comunidad de modelos de lenguaje.	LLM (Gran modelo del lenguaje).	`https://arxiv.org/abs/2307.09288`
Transfer Learning With Adaptive *Fine-tuning*	GREGA VRBANČIČ, (Graduate Student Member, IEEE), AND VILI PODGORELEC, (Member, IEEE)	2020	En aplicaciones de aprendizaje profundo, la escasez de datos confiables es un gran desafío, especialmente en medicina. Para abordar esto, se usa el aprendizaje por transferencia, que ajusta ciertas capas de un modelo preentrenado para una tarea relacionada. Sin embargo, seleccionar las capas adecuadas es problemático. Introducimos DEFT, un método basado en Evolución Diferencial para seleccionar capas en el ajuste fino de datos médicos. Evaluamos su rendimiento en la identificación de osteosarcoma en imágenes médicas, superando otros métodos en precisión (4.45 % - 32.75 %).	El método propuesto, Differential Evolution based *Fine-tuning* (DEFT), que fue evaluado en la tarea de identificación de osteosarcoma en un conjunto de datos de imágenes médicas.	`https://arxiv.org/abs/2307.09288`
VistaNet: Visual aspect attention network for multimodal sentiment analysis	Q. T. Truong y H. W. Lauw	2019	Se propone una red de Atención de Aspecto Visual para realizar análisis de sentimientos en una clasificación de 5 clases, se incluyen para el modelo tanto los componentes textuales como los componentes visuales.	Modelo de deep learning y códigos.	Q. T. Truong y H. W. Lauw, «VistaNet: Visual aspect attention network for multimodal sentiment analysis,» 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, págs.

Tabla 5.4: Tabla del estado del arte parte dos

5.2. Funciones BAC

Con base en los lineamientos establecidos por la norma ISO 16484-3 [44], se han definido las funciones BAC (Building Automation and Control) a trabajar, definidas en el anexo 7.8, un componente normativo y se denomina GA-Funktionsliste (GA-FL), que se traduce al español como 'lista de funciones de automatización de edificios'.

En la norma [44], se proporciona la siguiente definición: *'Die Funktionen werden zur Überwachung, Automatisierung, zur Optimierung und zum Betreiben von elektrischen und mechanischen Anlagen verwendet'*. Esto se traduce al español como: 'Las funciones se utilizan para supervisar, automatizar, optimizar y operar sistemas eléctricos y mecánicos'. Por tal motivo, las funciones a implementar son : control de ventanas (aplicación HVAC), control de puerta (aplicación de control de acceso) y control de lámparas y bombillos (aplicación de iluminación).

De acuerdo a las funciones que se implementan, se construye una representación a escala del entorno: el laboratorio de robótica. En este contexto, se presenta en la figura 5.2 la representación a escala del laboratorio, que se observa en la figura 5.3.

Figura 5.2: Laboratorio de robótica a escala. Fuente: autores.

Figura 5.3: Laboratorio de robótica. Fuente: autores.

En cumplimiento con lo estipulado en la norma, se establece en el anexo 7.8 la definición de los puntos de datos del sistema en formato tabular para suministrar información complementaria y la descripción integral del sistema en cuatro secciones principales, siguiendo las pautas establecidas en la Sección 5.5 de la misma normativa [44].

Figura 5.4: Proceso para la elección de las funciones BACs. Fuente: autores.

5.3. Servidor de Red BACnet

Para llevar a cabo las acciones deseadas en los diversos elementos del entorno a escala, se utiliza la librería BAC0 en Python para configurar un servidor en la Raspberry Pi 4 Model B [42]. De esta forma, se establece la capa de

red que permite la comunicación entre el usuario y el servidor.

Primero se crea un objeto `'servidor'` de BACnet con la dirección IP de la Raspberry Pi 4 y un identificador único. Luego, se crean elementos representados como objetos dentro de un hardware, los cuales se agregan a la aplicación BACnet del dispositivo. Esta configuración permite la interacción del cliente con las propiedades de cada objeto al identificarlos mediante sus nombres de dispositivo en la red.

El servidor supervisa de forma continua los cambios en cada uno de los objetos. Cuando se detecta una modificación en la propiedad del objeto `'presentValue'`, el servidor extrae el valor y lo utiliza como parámetro en una función que activa los elementos correspondientes. La estructura de los elementos dentro de una red BACnet se observa en la figura 5.5.

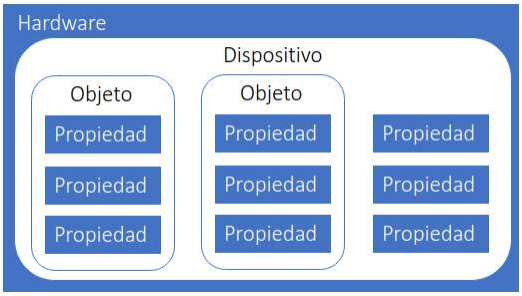


Figura 5.5: Estructura objetos BACnet y sus propiedades en hardware. Fuente: autores.

En el lado del cliente, utilizando la librería BAC0 [42], se crea un objeto 'cliente' de BACnet con la función `'lite'`, que requiere una dirección IP y una máscara de subred. Posteriormente, se configura un dispositivo con la dirección IP y el identificador único del servidor, estableciendo así la conexión con los dispositivos BACnet generados por el servidor. De esta manera, el cliente tiene la capacidad de visualizar y modificar las propiedades de cada objeto en la red. El cliente está integrado en un nodo de ROS y transmite la información a la que se suscribe. En la figura 5.6 se ilustrada el diagrama de conexiones del sistema.

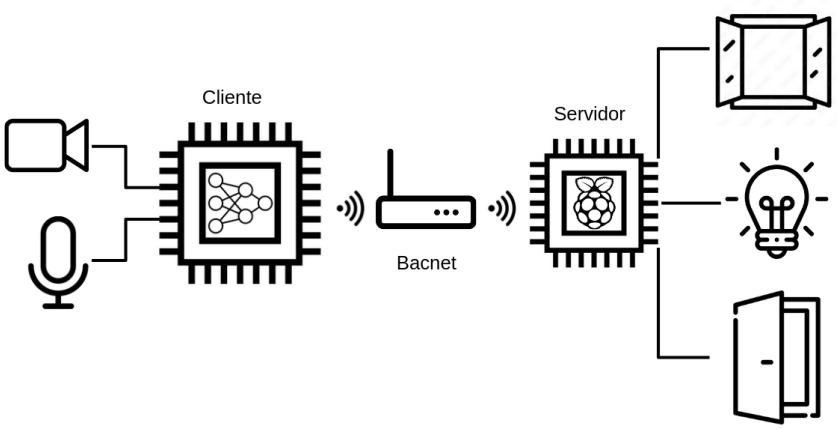


Figura 5.6: Comunicación del sistema. Fuente: autores.

5.4. ROS

5.4.1. Arquitectura y diagrama nodal

La arquitectura de todo el sistema es modular, lo que significa que se utiliza un nodo para cada función específica. Esta organización mejora la legibilidad del código y simplifica las labores de mantenimiento. La primera capa se encarga de la adquisición de información visual y auditiva, tal como se ilustra en la Figura 5.7. La segunda capa procesa esta información y extrae características relevantes. La tercera capa procesa la información mediante los modelos de lenguaje. Finalmente, la cuarta capa adquiere la información procesada y la transmite al servidor.

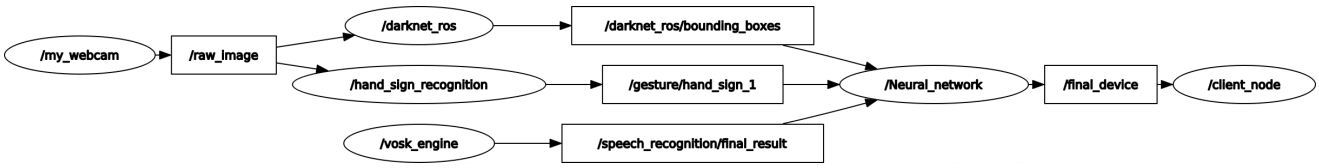


Figura 5.7: Diagrama nodal del sistema RQT. Fuente: autores.

5.4.1.1. Primera capa: Adquisición de datos

Los datos de diferentes modalidades son recopilados por diversos sensores. Para la modalidad visual, se utilizan cámaras IP debido a su alta portabilidad. En cuanto a la modalidad de voz, se emplea un micrófono. Estos datos son procesados en varios nodos de ROS (Robot Operating System).

Para adquirir correctamente la información visual, el nodo `/my_webcam` de la figura 5.7 utiliza OpenCV (Open Computer Vision), una librería de código abierto en Python. Esta permite verificar la disponibilidad de las cámaras, establecer comunicación con ellas y comenzar la adquisición de los datos. Posteriormente, se publica en el tópico `/raw_iamge` la información de la imagen. Es importante destacar que antes de publicar una imagen, esta debe convertirse en un mensaje de ROS mediante la función CVbridge().

Para adquirir correctamente la información de audio, el nodo llamado `/vosk_engine` en el sistema de la Figura 5.7) obtiene el número del dispositivo de entrada de audio predeterminado en el sistema utilizando la librería `sounddevice`, y configura y activa la transmisión de entrada de audio utilizando `RawInputStream`, con una tasa de muestreo de 4000 Hz y bloques de adquisición de 4000 muestras. Estos parámetros indican que se capturan 4000 muestras de audio por segundo y se procesan en bloques de 4000 muestras.

Para ambientes ruidosos se añadió la modalidad de entrada de texto, un nuevo hilo de forma paralela a la adquisición de voz real permite que el usuario ingrese comandos desde la terminal y que sean procesados de la misma manera que si se hubieran recibido como entrada de voz real.

5.4.1.2. Segunda capa: Extracción de características de los datos

El primer nodo que recibe la información adquirida por la cámara, `/darknet_ros`, se encarga de extraer las características de la imagen usando YOLO. Este algoritmo identifica y ubica, en este caso, personas y proporciona las coordenadas de un cuadro delimitador *(bounding box)*. A partir de estas coordenadas, se secciona la imagen en tres partes, dependiendo de la resolución de la cámara, para obtener la etiqueta de la posición (derecha, centro e izquierda). El segundo nodo, `/hand_sign_recognition`, utiliza la librería MediaPipe para el reconocimiento de

gestos de las manos a partir de *landmarks* (puntos clave) como las puntas de los dedos, las articulaciones de las falanges y la base de la palma, ver 5.8.

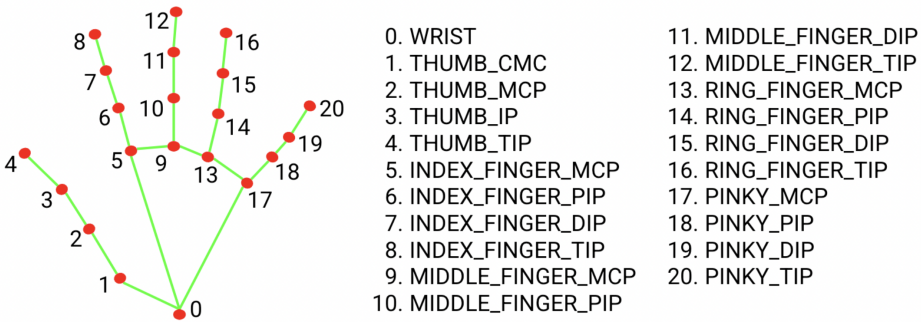


Figura 5.8: Puntos de referencia en la mano. Fuente: [14].

A partir de esta información, se utiliza un modelo clasificador de *keypoints* que clasifica estos *landmarks* preprocesados en ocho clases diferentes (arriba, abajo, izquierda, derecha, arriba-izquierda, arriba-derecha, centro e Izquierda-atras), ver figura 5.9. La etiqueta de la posición se publica en el tópico `/darknet_ros_bounding_boxescomo` y la del gesto detectado se publica en `/gesturehand_sign_1`.

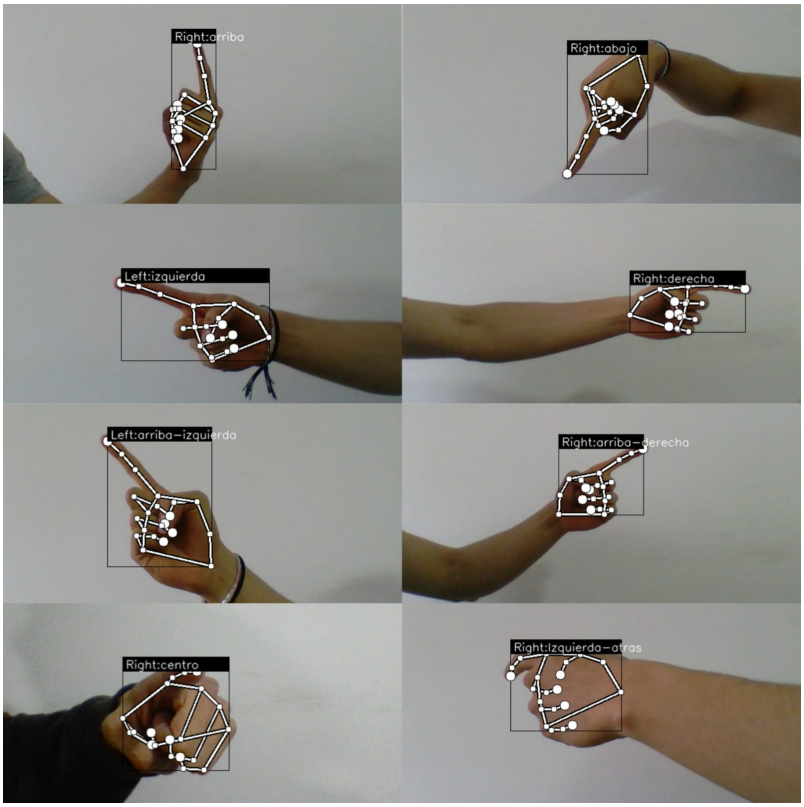


Figura 5.9: Etiquetas de gestos. Fuente: autores.

La información en modalidad auditiva es procesada por el mismo nodo donde se adquiere `/vosk_engine`, que utiliza el modelo preentrenado `'vosk-model-es-0.42'`de AlphaCephei. Este modelo tiene la función de transcribir oraciones en español a texto. El nodo genera el resultado final, que es la transcripción completa y precisa de la

totalidad de la instrucción y lo publica en el tópico `/speech_recognitionfinal_result`.

5.4.1.3. Tercera capa: Procesamiento de datos por modelos de lenguaje

Se emplea el modelo de lenguaje `'dccuchile/bert-base-spanish-wwm-uncased'` que se basa en la arquitectura BERT (Bidirectional Encoder Representations from Transformers), el cual se fundamenta en la arquitectura de redes neuronales *'transformers'* [45] y ha sido entrenado utilizando un extenso corpus en español. Es importante destacar que este modelo no hace distinción entre mayúsculas y minúsculas. La funcionalidad de BERT dentro del sistema resulta esencial debido a su capacidad para comprender oraciones, interpretar el contexto y extraer características cruciales de las palabras. Gracias a estas habilidades, BERT puede clasificar texto en diversas categorías. En este caso particular, las categorías se refieren a los diferentes dispositivos que se encuentran dentro del hogar a escala.

El lenguaje natural del ser humano presenta diversas formas de expresión. Por ello, la identificación de dispositivos y niveles de activación para los dispositivos se realiza a través la información visual y auditiva adquirida en la primer capa. En situaciones donde la información vocal no resulta suficiente, se utiliza la modalidad visual para complementar la instrucción, facilitando la discriminación del dispositivo al que se desea hacer referencia. Es crucial tener en cuenta la posición del sujeto y las señas que realiza con la mano, permitiendo identificar al objeto que desea accionar.

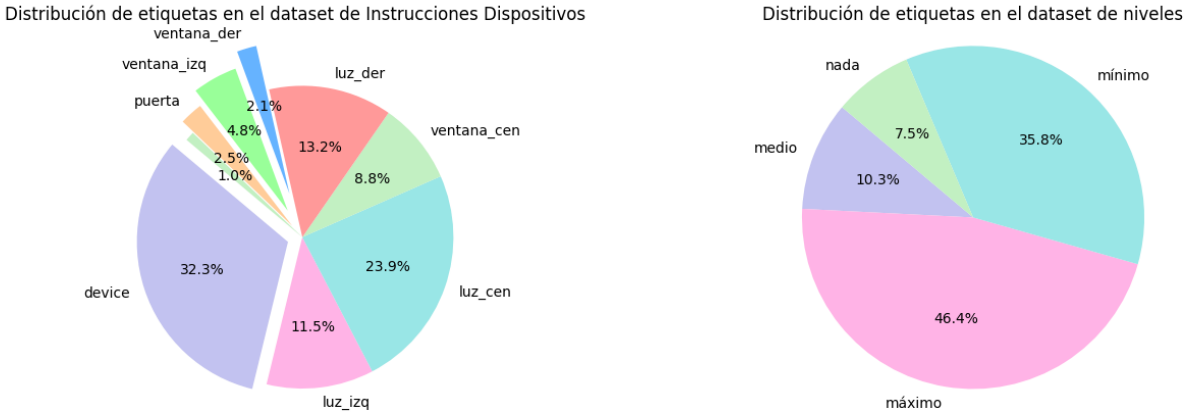
En este sentido, el nodo `/Neural_network` se suscribe a los tópicos que contienen la etiqueta de posición, de gesto y la transcripción. Esta última, si contiene el habilitador 'hola', entra a un primer modelo de BERT al que previamente se le realizó *Fine-tuning* empleando TensorFlow, este modelo clasifica el nivel o la ausencia de este en las siguientes clases: mínimo, medio, máximo y sin nivel. Luego, esta misma transcripción entra a un segundo modelo de BERT al que se le realizo *Fine-tuning* utilizando la librería Transformers de Hugging Face junto con su módulo Trainer, este modelo se encarga de discriminar el dispositivo al que se hace referencia o la ausencia de este en las siguientes clases: ventana derecha, ventana central, ventana izquierda, luz derecha, luz central, luz izquierda, puerta, acción con dispositivo no especificado e instrucción no reconocida.

Cuando el modelo no detecta un dispositivo en la transcripción, las etiquetas generadas por la red de reconocimiento de gestos con la mano y la etiqueta de la posición de la persona entran en juego. A través de una combinación de estas, se determina a qué dispositivo se refiere la persona. El dispositivo y nivel obtenidos se publican al tópico `/final_device` y es recibido por el nodo de cliente `/client_node`. Este nodo se encarga de comunicarse con el servidor y transmitirle el comando con la información necesaria para su ejecución.

El dataset utilizado para el entrenamiento de la clasificación de dispositivo y de nivel sigue la estructura detallada en el repositorio de Hugging Face en el anexo 7.7. La columna `text` contiene los datos con las instrucciones o datos sin sentido, la columna `label_text` contiene la etiqueta correspondiente a cada dato y la columna `label` corresponde a la etiqueta codificada que representa la salida esperada por parte del sistema. Estas etiquetas desempeñan un papel fundamental en el proceso de interpretación y facilitan la correcta asignación de acciones a cada dispositivo en función de las instrucciones proporcionadas. El dataset para la clasificación de dispositivo está compuesto por 1013 datos y el de clasificación de niveles está compuesto por 801 datos. Para la partición, se ha destinado el 20 % para el conjunto de pruebas y el 80 % para el conjunto de entrenamiento en ambos datasets. La distribución de etiquetas de ambos datasets es mostrada en la figura 5.10.

El entrenamiento de cada uno de los modelos, discriminación de dispositivos y niveles, se hacen en archivos `.ipynb`, vistos en el anexo 7.11, después se guardan los pesos del modelo de discriminación de nivel de manera local, y se

guarda el modelo de discriminación de dispositivos para posteriormente importarlos dentro de un nodo de ROS y realizar el proceso descrito anteriormente.



(a) Distribución de etiquetas del dataset (devices). Fuente: autores.

(b) Distribución de etiquetas del dataset (levels). Fuente: autores.

Figura 5.10: Distribución de etiquetas de ambos dataset

5.4.1.4. Cuarta capa: Transmisión de información al servidor

El nodo cliente, denominado `/client_node`, utiliza la función `write`. Esta función procesa la información extraída de los mensajes provenientes del tópico al que está suscrito, permitiendo ajustar los niveles de control de los dispositivos BACnet. Dependiendo del nombre del dispositivo y del nivel de control recibidos en el mensaje, se ejecuta la acción de control específica en el dispositivo BACnet correspondiente.

Capítulo 6

Resultados y Discusión

En este capítulo, se detallan los resultados de entrenamiento de los modelos y las pruebas llevadas a cabo para evaluar tanto el rendimiento como la eficacia del sistema desarrollado. Estas pruebas abarcan dos escenarios principales, cada uno con sus respectivas variaciones relevantes. Las métricas utilizadas para la evaluación se centran en la tasa de éxito, en la identificación de acciones correctas y el tiempo de respuesta del sistema.

6.1. Resultados del entrenamiento

En esta sección, se presentan los resultados de las métricas de evaluación de cada uno de los modelos ajustados finamente condensados en la Tabla 6.1, donde la segunda columna pertenece a las métricas del modelo 1, entrenado con la librería Transformers y la tercer columna pertenecen al modelo 2, entrenamiento con TensorFlow.

Comparando ambos modelos se puede observar que el modelo 2 alcanza un mayor número de iteraciones debido al valor de *"global _step"* y requiere menos operaciones de punto flotante, lo que sugiere una mayor eficiencia de procesamiento. Sin embargo, el modelo 1 logra una pérdida más baja durante el entrenamiento o sea que es más preciso. Además, el modelo 2 de TensorFlow se entrena durante un período más largo debido a un mayor número de épocas pero entrena más rapido en terminos de *"train_samples_per _second"*.

Métricas	**Modelo 1**	**Modelo 2**
global_step	820	1290
training_loss	0.023094856012158	0.0983
train_runtime	7047.3593	8447.67
train_samples_per_second	1.151	2.27
train_steps_per_second	0.116	0.15
total_flop	2133964770600960	4218391910400
epoch	10	30

Tabla 6.1: Métricas de entrenamiento de modelos NLP

6.2. Medición modular de tiempos

En esta sección se miden tiempos de procesamiento en distintos puntos del sistema, para cada medición se evaluaron 7 diferentes instrucciones cada una dirigida a un dispositivo diferente de la vivienda a escala con información

completa en voz.

Primero se mide el tiempo desde que la persona termina de decir la instrucción hasta que el nodo que contiene a la red *S2T* publica el resultado de la transcripción, como se muestra en la Tabla 6.2, se obtuvo tiempo promedio de 4.11 segundos.

Instrucciones	**Tiempo de procesamiento de Vosk [s]**
abre la puerta	3,999876
desactiva la ventana izquierda al cincuenta por ciento	5,604573
abre la ventana del medio	3,651822
cierra la ventana derecha	3,172121
prende luz izquierda al máximo	4,77674
pon la luz central al cincuenta	3,672712
enciende la luz derecha a nivel medio	3,942749
Promedio	**4,11722**

Tabla 6.2: Medición de tiempo de procesamiento de voz

También se mide el tiempo que el sistema tarda en transmitir la información que contiene la instrucción desde el nodo cliente hasta que el nodo servidor ejecuta la acción. Básicamente, esta medición refleja el tiempo que toma la red BACnet en transmitir la información. Como se muestra en la Tabla 6.3, se obtuvo un tiempo promedio de transmisión es de 0.36 segundos.

Instrucciones	**Tiempo [s]**
abre la puerta	0,003977
desactiva la ventana izquierda al cincuenta por ciento	0,685412
abre la ventana del medio	0,552684
cierra la ventana derecha	1,182475
prende luz izquierda al máximo	0,056629
pon la luz central al cincuenta	0,02912
enciende la luz derecha a nivel medio	0,076923
Promedio	**0.3696**

Tabla 6.3: Medición de tiempo de trasmisión red BACnet

Por último, se registra la duración del procesamiento de texto por parte de los modelos NLP. En la Tabla 6.4, se observa que el promedio de tiempo requerido por el modelo con *Fine-tuning* para la discriminación de dispositivos es de 0.17 segundos, mientras que el modelo discriminador de niveles requiere 0.44 segundos en promedio. Es importante destacar que esta diferencia en tiempos de procesamiento se debe, en parte, a las distintas técnicas de entrenamiento empleadas en cada red. Notablemente, el modelo de discriminación de niveles, entrenado con la librería TensorFlow, demuestra una duración de procesamiento que es más del doble en comparación con el modelo entrenado utilizando la librería Transformers.

Instrucciones	Modelo de dispositivos [s]	Modelo de niveles [s]
abre la puerta	0,399102	0,400081
desactiva la ventana izquierda al cincuenta por ciento	0,172322	0,47538
abre la ventana del medio	0,13977	0,435394
cierra la ventana derecha	0,109004	0,440853
prende luz izquierda al máximo	0,126258	0,429005
pon la luz central al cincuenta	0,148536	0,460302
enciende la luz derecha a nivel medio	0,127689	0,451311
Promedio	**0,174**	**0,4417**

Tabla 6.4: Medición de tiempo de procesamiento de modelos de NLP

6.3. Escenarios de prueba

En esta sección, establecemos dos escenarios de prueba que incorporan variaciones para evaluar el funcionamiento del sistema. Estos escenarios abarcan situaciones con información completa en la voz y situaciones con información incompleta en la voz.

Para asegurar un desempeño óptimo en la red de reconocimiento de gestos, es fundamental considerar las siguientes variables del entorno:

1. **Entorno del Laboratorio de Robótica:** La adquisición de datos se realiza en un entorno altamente controlado, específicamente en el laboratorio de robótica.

2. **Dispositivos Utilizados:** Los dispositivos empleados para la captura de datos incluyen la cámara IP de un dispositivo móvil con una resolución de 1280 x 720 pixeles y un micrófono Bluetooth. Estos componentes son parte integral del entorno de prueba y desempeñan un papel fundamental en la recopilación de datos para la evaluación del sistema.

3. **Distancia del Usuario a la Cámara:** Las pruebas se realizaron con una distancia no mayor de 2 metros entre el usuario y la cámara. A esta distancia, la cámara abarca aproximadamente 2.7 metros en horizontal. Esta distancia garantiza una captura efectiva de los gestos y una interacción fluida con el sistema. Fue determinada mediante pruebas de variación de distancia y acierto en el reconocimiento de gestos en cada variación.

4. **Distancia del Usuario al Micrófono:** Las pruebas se realizaron con una distancia cercana al micrófono, donde se alcanza a discriminar la voz del usuario del ruido de fondo.

5. **Iluminación Mínima de 5 Lux:** Es esencial destacar que el reconocimiento de gestos se realiza con una iluminación mínima de 5 lux. Este nivel de iluminación permite un funcionamiento adecuado de los sistemas de captura de gestos y garantiza resultados precisos en condiciones de poca luz, fue determinado realizando pruebas de variación de iluminación, y de acierto de reconocimiento de gestos en cada variación.

6. **Cantidad de Personas Visibles por la Cámara:** La cantidad máxima de personas que pueden ser visibles por la cámara es 1. Esto se debe a que el modelo de detección de objetos YOLO segmenta la clase 'persona', pero no tiene la capacidad de discriminar cuál de las personas en la escena está expresando la instrucción. Por lo tanto, YOLO generará las coordenadas de todas las personas que detecte en la imagen.

7. **Consideraciones Temporales en el Reconocimiento de Gestos y Transcripción:** Por lo tanto, se requiere que la persona continúe realizando el gesto con la mano durante aproximadamente 4 segundos, considerando la medición previa del procesamiento de Vosk. Este intervalo es necesario para garantizar la sincronización adecuada entre la información visual y auditiva.

6.3.1. Escenario de información completa en la voz

Este escenario implica pruebas en situaciones donde la información proporcionada en la entrada de voz es completa permitiendo identificar el dispositivo y el nivel deseado, para este escenario no es relevante la ubicación del usuario.

6.3.1.1. Variación ideal de modalidad auditiva

En esta variante, aprovechando la capacidad del sistema de aceptar entrada de texto, se realiza una prueba de tiempo de respuesta. Se ingresan instrucciones en formato escrito en la consola, teniendo en cuenta que el proceso de conversión de voz a texto (S2T) es la etapa que requiere más tiempo. Los resultados se presentan en la Tabla 6.5, donde se observa que el tiempo de respuesta promedio del sistema en esta variante ideal es de 2.42 segundos.

Instrucciones	**Tiempo de procesamiento del sistema [s]**
abre la puerta	2,295303
desactiva la ventana izquierda al cincuenta por ciento	3,441995
abre la ventana del medio	2,126514
cierra la ventana derecha	1,700718
prende luz izquierda al máximo	2,72184
pon la luz central al cincuenta	2,480352
enciende la luz derecha a nivel medio	2,214456
Promedio	**2,4258**

Tabla 6.5: Tiempo de procesamiento del sistema para diferentes instrucciones información completa

6.3.1.2. Variación voces

En esta sección, llevamos a cabo pruebas con cinco individuos en un escenario de funcionamiento típico del sistema, a través de 21 instrucciones, 3 por cada dispositivo para contemplar los 3 niveles, esto, con el objetivo de evaluar cómo el sistema se desempeña en diversas condiciones vocales.

Los resultados se presentan en la Tabla 6.6. En promedio, el tiempo de procesamiento del sistema no mostró variaciones significativas en diferentes perfiles vocales, en cuanto a la tasa de éxito y al tiepo de procesamiento, obteniendo un tiempo promedio general de 3.97052 segundos y una tasa de aciertos promedio de 68.72 %. Los detalles adicionales de las pruebas con cada sujeto se encuentran en el anexo **??**.

Sujeto	Tiempo de Respuesta Promedio (s)	Tasa de Acierto (%)	Transcripciones Exactas
Sebastian	3.6728	67.74 %	12
Laura	4.3275	65.62 %	8
Juan	4.1081	67.74 %	6
Felipe	3.8666	70 %	16
Gabriela	3.8776	72.41 %	13

Tabla 6.6: Métricas generales de pruebas con información completa

6.3.2. Escenario de información incompleta en la voz

Este escenario implica situaciones en las que la información proporcionada en la entrada de voz es parcial o insuficiente para determinar el dispositivo deseado por lo que se requiere utilizar información adicional de gestos y ubicación para generar la representación compartida e identificarlo. Para este escenario los usuarios se ubican y señalan como se observa en las figuras 6.1, 6.3 y 6.5, algunos de los cambios de estado en los diferentes elementos del entorno a escala se observan en las figuras 6.2, 6.4 y 6.6.

Figura 6.1: Posición usuario para comando 'Abrir puerta'. Fuente: autores.

Figura 6.2: Cambio de estado de elemento. Fuente: autores.

Figura 6.3: Posición usuario para comando 'Prende luz central'. Fuente: autores.

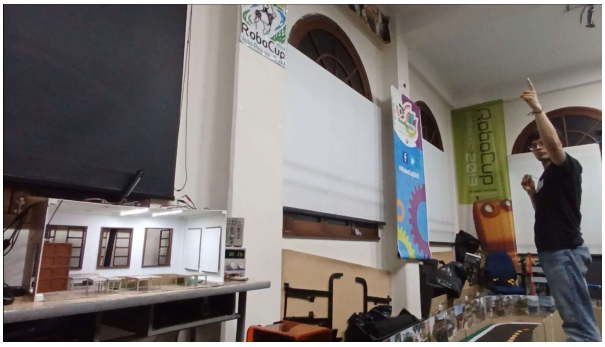


Figura 6.4: Cambio de estado de elemento. Fuente: autores.

Figura 6.5: Posición usuario para comando 'Prende luz central'. Fuente: autores.

Figura 6.6: Cambio de estado de elemento. Fuente: autores.

6.3.2.1. Variación modalidad escrita

En esta variante, Se introducen instrucciones en formato escrito en la consola que solo contienen el nivel deseado, Los resultados se presentan en la Tabla 6.7 , donde se observa que el tiempo de respuesta promedio del sistema es de 2.07 segundos.

Instrucciones	**Tiempo de procesamiento del sistema [s]**
prende esa	1,148285
apaga esa	1,29071
pon esa luz a nivel medio	1,558564
desactiva esa al cincuenta porciento	1,96939
abre esa ventana	2,39804
cierra esa	1,074722
abre esa ventana al nivel medio	6,015264
abre esa	1,140684
Promedio	**2,0744**

Tabla 6.7: Tiempo de procesamiento del sistema para diferentes instrucciones información incompleta

6.3.2.2. Variación voces

En esta sección, llevamos a cabo pruebas con cinco individuos en un escenario de funcionamiento típico del sistema, a través de 21 instrucciones con información incompleta, 3 por cada dispositivo para contemplar los 3 niveles, esto, con el objetivo de evaluar cómo el sistema se desempeña en diversas condiciones vocales.

Los resultados se presentan en la Tabla 6.8. En promedio, el tiempo de procesamiento del sistema no mostró variaciones significativas en diferentes perfiles vocales, en cuanto a la tasa de éxito y al tiempo de procesamiento, obteniendo un tiempo promedio general de 4.17 segundos y una tasa de acierto promedio de 65.76 %. Los detalles adicionales de las pruebas con cada sujeto se encuentran en el anexo 7.10.

Sujeto	Tiempo de Respuesta Promedio (s)	Tasa de Acierto (%)	Transcripciones Exactas
Sebastian	4.7124	65.62 %	11
Laura	4.0994	70 %	10
Juan	3.8017	67.74 %	7
Felipe	4.3445	63.63 %	5
Gabriela	4.1082	61.76 %	2

Tabla 6.8: Métricas generales de pruebas con información incompleta

La evidencia de los tiempos tomados se encuentra disponible en el enlace del segundo ítem del anexo 7.11, donde se presenta un video recopilatorio con diversas instrucciones proporcionadas por diferentes sujetos. En el video, se observa que la persona debe realizar gestos con la mano frente a la cámara, dentro del área designada para el escenario de prueba, y simultáneamente expresar la instrucción correspondiente al escenario de información incompleta.

Es importante destacar que en los dos escenarios el sistema transcribe menos de la mitad de las instrucciones de manera precisa. Sin embargo, cabe resaltar que estos casos son favorables, ya que evidencian el sólido desempeño de los modelos de NLP, que son capaces de reconocer y clasificar adecuadamente frases con transcripciones similares pero que no están dentro del dataset de entrenamiento.

También se observa que los tiempos de procesamiento del sistema no muestran una variación significativa al comparar la entrada de información exclusivamente a través de la modalidad auditiva, que contiene información suficiente para discriminar el nivel y dispositivo, con la entrada de información que requiere tanto la modalidad auditiva como la visual para complementarse en la discriminación de nivel y dispositivo.

Capítulo 7

Conclusiones y Trabajos Futuros

7.1. Conclusiones

Basado en la información documentada en el estado del arte, se afirma que las representaciones compartidas e información multimodal han emergido como tendencias significativas en la investigación en los últimos años. Esta afirmación se respalda en artículos clave como [1], [9], [3] y [11], que han abordado temáticas relacionadas con la intersección de datos multimodales y representaciones compartidas. Además, se observa una clara tendencia hacia el desarrollo de modelos de lenguaje, como se evidencia en las investigaciones [21] y [22]. Estos hallazgos resaltan la importancia de abordar proyectos de grado que se alineen con estas tendencias, ya que contribuirán a la vanguardia de la investigación en este campo y de forma transdiciplinar .

Con base en la estructura de operación de ROS, se ha logrado crear un diseño del sistema satisfactorio. Este diseño involucra nodos que integran redes neuronales pre-entrenadas y modelos NLP en español, diversas herramientas tecnológicas obtenidas tanto del estado del arte como de repositorios abiertos y el protocolo de comunicación BACnet basado en la definición de la lista de funciones para la interacción con dispositivos en hogares, como se describe en el anexo 7.8. Gracias a todas estas herramientas y conocimientos adquiridos, se ha logrado llevar a cabo el diseño que se describe en la capítulo 5 y que se sitentiza en las figuras 5.7 y 5.6.

Según lo explicado en la sección 1.4, para el desarrollo de la aplicación de alto nivel destinada a personas con analfabetismo digital y siguiendo los requisitos de diseño, se logra la integración exitosa en ROS de la adquisición de la información con las redes preentrenadas que extraen las características de la información adquirida y a su vez con la transmisión de información de cliente a servidor a través del protocolo BACnet. Esto ha dado lugar a la creación de un repositorio que contiene la configuración del servidor y otro que contiene los distintos paquetes de las redes y toda la configuración implícita de la integración en el cliente, además de diversos datasets, datasets que contienen keypoints de manos para la clasificación de gestos, y datasets que contienen instrucciones de operación de dispositivos en hogares con tres distintos niveles de funcionamiento en el formato adecuado para el entrenamiento del *Fine-tuning* del modelo BETO, ver el anexo 7.11, en general se desarrolla el sistema que asiste en la operación de las funciones de domótica, permitiendo el uso del lenguaje natural y teniendo en cuenta la información multimodal (visual, auditiva y escrita). Para obtener una comprensión detallada de cómo opera el sistema, capa por capa, y cómo se integran y colaboran sus componentes para alcanzar los resultados deseados, en el Capítulo 5 se ven las métricas y situaciones de evaluación expuestas en el capitulo 6

Los resultados de las pruebas detalladas en el Capítulo 6 revelaron que el sistema tarda aproximadamente 4 segundos

en ejecutar una acción en los dispositivos de la vivienda desde que el usuario termina de expresar la instrucción. La tasa de aciertos para instrucciones vocales promedia alrededor del 67.23 %. Es destacable el éxito del proceso de *Fine-tuning* en ambos modelos de lenguaje, ya que a pesar de que menos del 50 % de las instrucciones se transcriben exactamente, los modelos aún clasifican esas transcripciones de manera precisa en sus respectivas clases válidas. También se confirmó el eficiente funcionamiento de la red Bacnet, con un rápido tiempo de transmisión promedio de 0.36 segundos, además de su facilidad de uso y adaptabilidad para nuevos dispositivos. Estos hallazgos respaldan la efectividad y confiabilidad del software desarrollado.

7.2. Trabajos futuros

Desarrollar el software de asistencia mediante la integración y construcción de artefactos de software bas El alcance del proyecto puede extenderse a elementos reales en diferentes ubicaciones. La integración del sistema con dichos elementos lo haría mucho más práctico.

Para mejorar el tiempo de respuesta del sistema, es necesario llevar a cabo un procesamiento de datos más eficiente. Esto podría lograrse mediante la mejora de las redes preentrenadas actualmente en uso o su reemplazo por aquellas con un mejor rendimiento.

Además, se podría ampliar las modalidades del sistema, tales como el reconocimiento de color, temperatura, facial, visión infrarroja, reconocimiento de objetos, entre otros. Esto, junto con una extensa lista de funcionalidades y más sensores, haría que el sistema sea más robusto.

Adicionalmente, se podría incorporar un sistema de retroalimentación que permita monitorear las variables del entorno, donde el sistema podría ajustarse según la voluntad del usuario y verificar automáticamente si un elemento ha sido accionado correctamente. Además, considerando variables como la temperatura, el sistema tomaría acciones de manera autónoma, esta retroalimentación también podría permitir que el sistema sea adaptable a otros dominios.

Bibliografía

[1] Gi Hyun Lim. «Shared representations of actions for alternative suggestion with incomplete information». En: *Robotics and Autonomous Systems* 116 (jun. de 2019), págs. 38-50. ISSN: 09218890. DOI: 10.1016/j.robot.2019.02.005.

[2] Yang Baoyao, Andy J. Ma y Pong C. Yuen. «Learning domain-shared group-sparse representation for unsupervised domain adaptation». En: *Pattern Recognition* 81 (2018), págs. 615-632. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2018.04.027. URL: https://www.sciencedirect.com/science/article/pii/S0031320318301614.

[3] Naimeh Alipour y Jafar Tahmoresnezhad. «Heterogeneous domain adaptation with statistical distribution alignment and progressive pseudo label selection». En: *Applied Intelligence* 52.7 (mayo de 2022), págs. 8038-8055. ISSN: 1573-7497. DOI: 10.1007/s10489-021-02756-x. URL: https://doi.org/10.1007/s10489-021-02756-x.

[4] Silvia Lago Martínez. «Brecha digital, inclusión y apropiación de tecnologías». En: (2019).

[5] Mariana Ramos. *Radiografía de la Era Digital en Colombia (CNC 2020)*. 2020. URL: https://www.centronacionaldeconsu com/post/radiografia-de-la-era-digital-en-colombia-cnc-2020.

[6] Ana Rivoir. «Personas mayores y tecnologías digitales». En: *Miradas críticas de la apropiación en América Latina* (2019), pág. 51.

[7] Juan José Gómez Acosta et al. «Documento marco Proyección social». En: (2016).

[8] *La Asamblea General adopta la Agenda 2030 para el Desarrollo Sostenible - Desarrollo Sostenible*. URL: https://www.un.org/sustainabledevelopment/es/2015/09/la-asamblea-general-adopta-la-agenda-2030-para-el-desarrollo-sostenible/ (visitado 20-10-2022).

[9] Wilfredo Graterol et al. «Emotion Detection for Social Robots Based on NLP Transformers and an Emotion Ontology». en. En: *Sensors (Basel)* 21.4 (feb. de 2021).

[10] Juan David Granados Figueroa. «Aplicación de técnicas de Machine Learning para hacer análisis de polaridad de sentimientos en texto para detectar tendencias de opinión en plataformas online». En: (2020), pág. 70. URL: http://hdl.handle.net/11634/34933.

[11] Masahiro Suzuki y Yutaka Matsuo. «A survey of multimodal deep generative models». En: *Advanced Robotics* 36.5-6 (2022), págs. 261-278. ISSN: 15685535. DOI: 10.1080/01691864.2022.2035253. arXiv: 2207.02127.

[12] Quoc Tuan Truong y Hady W. Lauw. «VistaNet: Visual aspect attention network for multimodal sentiment analysis». En: *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019* (2019), págs. 305-312. ISSN: 2159-5399. DOI: 10.1609/aaai.v33i01.3301305.

[13] Johan David Diaz Mendivelso y Marco Javier Suarez Baron. «Análisis social aplicando técnicas de lenguaje natural a información extraída de twitter». En: *Scientia et Technica* 24.3 (sep. de 2019), págs. 496-503. DOI: 10.22517/23447214.21731. URL: https://revistas.utp.edu.co/index.php/revistaciencia/article/view/21731.

[14] Camillo Lugaresi et al. *MediaPipe: A Framework for Building Perception Pipelines*. 2019. arXiv: 1906.08172 [cs.DC].

[15] Alpha Cephei Inc. *VOSK Offline Speech Recognition API*. URL: https://alphacephei.com/en/.

[16] Wataru Noguchi et al. «Superposition mechanism as a neural basis for understanding others». En: *Scientific Reports* 12.1 (dic. de 2022). ISSN: 20452322. DOI: 10.1038/s41598-022-06717-3.

[17] Mina Jafari et al. «LMISA: A lightweight multi-modality image segmentation network via domain adaptation using gradient magnitude and shape constraint». En: *Medical Image Analysis* 81 (oct. de 2022), pág. 102536. ISSN: 13618415. DOI: 10.1016/j.media.2022.102536. URL: https://linkinghub.elsevier.com/retrieve/pii/S1361841522001839.

[18] Humberto Rojas Pinilla. «Memorias /International Engineering Seminar IES UNISANGIL 2015 Desarrollo rural y sostenibilidad». En: (2015), pág. 70. URL: https://unisangil.edu.co/phocadownload/investigaciones_doc/memorias_IES_2015_ISSN_2422_5088_volumen_2.pdf#page=40.

[19] Luis Vicente Colorado Franco y Narcisa Pilar Colorado Franco. «Cálculo del margen de tolerancia permisible aplicando el factor de experiencia de buque en la transferencia marítima de petróleo crudo y derivados del petróleo». En: *Revista Científica y Tecnológica UPSE* 7.1 (2020), págs. 14-20. ISSN: 1390-7638. DOI: 10.26423/rctu.v7i1.507.

[20] Matías Pizarro. «INMÓTICA Y GESTIÓN ENERGÉTICA BAJO EL ESTÁNDAR KNX». En: (2018).

[21] Hugo Touvron et al. *Llama 2: Open Foundation and Fine-Tuned Chat Models*. 2023. arXiv: 2307.09288 [cs.CL].

[22] OpenAI. *GPT-4 Technical Report*. 2023. arXiv: 2303.08774 [cs.CL].

[23] Alex Krizhevsky, Ilya Sutskever y Geoffrey E. Hinton. «ImageNet classification with deep convolutional Neural Networks». En: *Communications of the ACM* 60.6 (2017), págs. 84-90. DOI: 10.1145/3065386.

[24] Ian Goodfellow, Yoshua Bengio y Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[25] Yann LeCun, Yoshua Bengio y Geoffrey Hinton. «Deep learning». En: *Nature* 521.7553 (2015), págs. 436-444. DOI: 10.1038/nature14539.

[26] Haoyang Zeng et al. «Convolutional neural network architectures for predicting DNA–protein binding». En: *Bioinformatics* 32.12 (jun. de 2016), págs. i121-i127. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btw255. eprint: https://academic.oup.com/bioinformatics/article-pdf/32/12/i121/49020280/bioinformatics_32_12_i121.pdf. URL: https://doi.org/10.1093/bioinformatics/btw255.

[27] Israel Rivera Zárate, Miguel Hernández Bolaños y Patricia Pérez Romero. «REDES NEURONALES RECURRENTES PARA EL PROCESAMIENTO DE SECUENCIAS CODIFICADO EN PYTHON». En: ().

[28] Carlos Arana. *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales*. Inf. téc. Serie Documentos de Trabajo, 2021.

[29] Kamilya Smagulova y Alex Pappachen James. «A survey on LSTM memristive neural network architectures and applications». En: *The European Physical Journal Special Topics* 228.10 (2019), págs. 2313-2324.

[30] Camilo Ferro, Nathalia Celis Mayorga y Alejandro Casallas García. *Llenado de series de datos de 2014 a 2019 de PM2.5 por medio de una red neuronal y una regresión lineal*. Ago. de 2020. DOI: 10.13140/RG.2.2.35092.53126.

[31] Grega Vrbančič y Vili Podgorelec. «Transfer Learning With Adaptive Fine-Tuning». En: *IEEE Access* 8 (2020), págs. 196197-196211. DOI: 10.1109/ACCESS.2020.3034343.

[32] URL: https://huggingface.co/docs/transformers/training.

[33] Sholom Weiss et al. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. 1.ª ed. Springer, 2004. ISBN: 0387954333; 9780387954332. URL: libgen.li/file.php?md5=341a70158173878be25bf6b3d4c0458f.

[34] Tania Pereira, António Cunha y Hélder P. Oliveira. «Special Issue on Novel Applications of Artificial Intelligence in Medicine and Health». En: *Applied Sciences* 13.2 (2023). ISSN: 2076-3417. DOI: 10.3390/app13020881. URL: https://www.mdpi.com/2076-3417/13/2/881.

[35] Hermann Merz et al. *Building Automation: Communication Systems with EIB/KNX, Lon und Bacnet*. Springer, 2018.

[36] *Crea Modelos de Aprendizaje Automático de Nivel de Producción Con Tensorflow*. URL: https://www.tensorflow.org/?gclid=Cj0KCQiApKagBhC1ARIsAFc7Mc7EZw9yJT5a2YSkRk07HexJlIlZarZJ8J84vPkxzsyd2ePZw0tCp5wa
wcB&hl=es-419.

[37] *Welcome to Python.org*. URL: https://www.python.org/.

[38] Bernardo Ronquillo Japon. *Hands-on ROS for Robotics Programming*. URL: https://www.oreilly.com/library/view/hands-on-ros-for/9781838551308/64354160-9145-44e4-9fe6-bbba39ac5385.xhtml.

[39] David Pérez Román. «Robot Móvil Controlado por ROS Mobile Robot Powered by ROS». En: ().

[40] *Robot operating system*. URL: https://www.ros.org/.

[41] *Learn*. URL: https://scikit-learn.org/stable.

[42] *BAC0: BAC0 is a Python 3 (3.6 and over) scripting application that uses BACpypes to process BACnet™ messages on an IP network. This library brings out simple commands to browse a BACnet network, read properties from BACnet devices, or write to them.* https://github.com/ChristianTremblay/BAC0.git. Consultado el 30 de agosto de 2023. 2023.

[43] Peiyuan Jiang et al. «A Review of Yolo Algorithm Developments». En: *Procedia Computer Science* 199 (2022). The 8th International Conference on Information Technology and Quantitative Management (ITQM 2020 2021): Developing Global Digital Economy after COVID-19, págs. 1066-1073. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.01.135. URL: https://www.sciencedirect.com/science/article/pii/S1877050922001363.

[44] *Systeme der Gebäudeautomation (GA) - Teil 3: Funktionen (ISO 16484-3:2005); Deutsche Fassung EN ISO 16484-3:2005*.

[45] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

ANEXOS

7.3. anexo A:

Listing 7.1: Nodo de adquisición de imagen

```
1
2  #!/usr/bin/python3
3
4  import rospy
5  import cv2
6  from std_msgs.msg import String
7  from cv_bridge import CvBridge, CvBridgeError
8  from sensor_msgs.msg import Image
9
10 from colorama import Fore, Style
11 #from model import KeyPointClassifier
12 def publish_image():
13
14     bridge = CvBridge()
15
16     try:
17         camera1_name = rospy.get_param("my_webcam/camera_name_Pc")
18         print(Fore.BLUE + "camera1_name: %s" + Style.RESET_ALL, camera1_name)
19         image_pub1 = rospy.Publisher("image_raw_alejandro", Image, queue_size=10)
20         capture1 = cv2.VideoCapture(camera1_name)
21
22     except Exception as e:
23         print(Fore.RED + f"Error: {e}"+ Style.RESET_ALL)
24         capture1 = None  # Marcar la camara como no disponible
25
26     try:
27         camera2_name = rospy.get_param("my_webcam/camera_name_2")
28         print(Fore.BLUE + "camera2_name: %s" + Style.RESET_ALL, camera2_name)
29         image_pub2 = rospy.Publisher("image_raw_alejandro_2", Image, queue_size=10)
30         capture2 = cv2.VideoCapture(camera2_name)
31     except Exception as e:
32         print(Fore.RED + f"Error: {e}"+ Style.RESET_ALL)
33         capture2 = None
34
35     rospy.init_node('img_publisher', anonymous=False)
36     rate = rospy.Rate(100)  # Incrementa la tasa a 100 Hz (ajusta seg n sea necesario)
```

```
37
38      while not rospy.is_shutdown():
39          if capture1 is not None:
40              ret1 , img1 = capture1.read()
41              if ret1 is True:
42                  pose_message1 = bridge.cv2_to_imgmsg(img1, "bgr8")
43                  image_pub1.publish(pose_message1)
44          #         cv2.imshow('Camera 1 Image', img1)
45
46          if capture2 is not None:
47              ret2 , img2 = capture2.read()
48              if ret2 is True:
49                  pose_message2 = bridge.cv2_to_imgmsg(img2, "bgr8")
50                  cv2.imshow('Camera 2 Image', img2)
51  #                cv2.imshow(img2)
52          rate.sleep()
53
54
55  if __name__ == '__main__':
56      try:
57          print( Fore.GREEN  + "Image is being published to the topic : " + Fore.RED + " /
                image_raw_alejandro..." + Style.RESET_ALL)
58          publish_image()
59      except rospy.ROSInterruptException as e:
60          print(Fore.RED + f"Error: {e}" + Style.RESET_ALL)
```

7.4. anexo B:

Listing 7.2: Nodo de vosk

```
#!/usr/bin/env python3
# -- coding: utf-8 --

import os
import sys
import json
import queue
import vosk
import sounddevice as sd
from mmap import MAP_SHARED

import rospy
import rospkg
from ros_vosk.msg import speech_recognition
from std_msgs.msg import String, Bool

import vosk_ros_model_downloader as downloader
import datetime
class vosk_sr():
    def __init__(self):
        model_name = rospy.get_param('vosk/model', "vosk-model-es-0.42")

        rospack = rospkg.RosPack()
        rospack.list()
        package_path = rospack.get_path('ros_vosk')

        models_dir = os.path.join(package_path, 'models')
        model_path = os.path.join(models_dir, model_name)


        if not rospy.has_param('vosk/model'):
            rospy.set_param('vosk/model', model_name)

        self.tts_status = False
        self.pub_final = rospy.Publisher('speech_recognition/final_result',String,
            queue_size=0)
        self.rate = rospy.Rate(1000)
        rospy.on_shutdown(self.cleanup)
        self.msg = speech_recognition()
        self.q = queue.Queue()

        self.input_dev_num = sd.query_hostapis()[0]['default_input_device']
        if self.input_dev_num == -1:
            rospy.logfatal('No input device found')
            raise ValueError('No input device found, device number == -1')

```

```
        device_info = sd.query_devices(self.input_dev_num, 'input')
        self.samplerate = int(device_info['default_samplerate'])
        rospy.set_param('vosk/sample_rate', self.samplerate)
        self.model = vosk.Model(model_path)

    def cleanup(self):
        rospy.logwarn("Shutting down VOSK speech recognition node...")

    def stream_callback(self, indata, frames, time, status):
        #"""This is called (from a separate thread) for each audio block."""
        if status:
            print(status, file=sys.stderr)
        self.q.put(bytes(indata))
        #self.current_time = datetime.datetime.now().strftime("%M:%S,%f")
        #print("comeinza reconocimeito"+self.current_time)

    def speech_recognize(self):
        try:
            with sd.RawInputStream(samplerate=self.samplerate, blocksize=4000, device=
                self.input_dev_num, dtype='int16',
                                   channels=1, callback=self.stream_callback):
                rospy.logdebug('Started recording')
                rec = vosk.KaldiRecognizer(self.model, self.samplerate)
                print("Vosk is ready to listen!")
                isRecognized = False

                while not rospy.is_shutdown():
                    data = self.q.get()
                    if rec.AcceptWaveform(data):
                        self.current_time = datetime.datetime.now().strftime("%M:%S,%f")
                        print("reconoce bloque"+self.current_time)
                        result = rec.FinalResult()
                        diction = json.loads(result)
                        lentext = len(diction["text"])
                        if lentext > 2:
                            result_text = diction["text"]
                            rospy.loginfo(result_text)
                            isRecognized = True
                        else:
                            isRecognized = False
                        rec.Reset()

                    if (isRecognized is True):
                        self.msg.isSpeech_recognized = True
                        #self.current_time = datetime.datetime.now().strftime("%M:%S,%f
                            ")
                        #print("publica: "+self.current_time)
                        self.msg.time_recognized = rospy.Time.now()
                        self.msg.final_result = result_text
```

```
                        self.msg.partial_result = "unk"
                        self.pub_final.publish(result_text)
                        self.q.queue.clear()
                        isRecognized = False

        except Exception as e:
            exit(type(e).__name__ + ': ' + str(e))
        except KeyboardInterrupt:
            rospy.loginfo("Stopping the VOSK speech recognition node...")
            rospy.sleep(1)
            print("node terminated")

if __name__ == '__main__':
    try:
        rospy.init_node('vosk', anonymous=False)
        rec = vosk_sr()
        rec.speech_recognize()
    except (KeyboardInterrupt, rospy.ROSInterruptException) as e:
        rospy.logfatal("Error occurred! Stopping the vosk speech recognition node...")
        rospy.sleep(1)
        print("node terminated")
```

7.5. anexo C: Listado de Repositorios

- **alphacep/ros-vosk:** https://github.com/alphacep/ros-vosk
- **TrinhNC/ros_hand_gesture_recognition:** https://github.com/TrinhNC/ros_hand_gesture_recognition
- **ChristianTremblay/BAC0:** https://github.com/ChristianTremblay/BAC0
- **eggedrobotics / darknet_ros:** https://github.com/leggedrobotics
- **Hugging Face Fine-tune a pretrained model:** https://huggingface.co/docs/transformers/training

7.6. anexo D: Heramientas tecnológicas usadas

- **Google Colab:** https://colab.research.google.com/
- **jupyterLab:** https://jupyter.org/
- **ROS:** https://www.ros.org/
- **Python:** https://www.python.org/
- **Hugging Face** https://huggingface.co/
- **Visual Studio Code** https://code.visualstudio.com/
- **github** https://github.com/

7.7. anexo E: Datasets usados

- **cbasconc/instructions_Device:** https://huggingface.co/datasets/cbasconc/instructions_Device
- **cbasconc/levels:** https://huggingface.co/datasets/cbasconc/levels

7.8. anexo J: Tabla de funciones BACS

Example for heterogeneous Systems

BACS Function List

ISO 16484-3

1) Steady-state output, e.g.: 0,I,II=2 BO
Pulsed output, e.g.: 0,I,II=3 BO

Positioning outp. close-0-open=2 BO
Pulse width modulation output=1 BO

2) Active or passive

3) Only shared (networked) I/O data points from foreign systems for interoperable functions

4) Per input point address for a) collected, b) delayed or c) suppressed information

5) Per output point address

System : windows, door, and lighting automation system		I/O functions										Processing functions																			
		Physical					Shared 3), 9)					Monitoring							Interlocks					Closed loop control							
		Binary output switching/positioning 1)	Analog output positioning	Binary input state	Binary input counting	Analog input 2)	Binary value (output), switching	Analog value (output), positioning/setpt.	Binary value (input), state	Accumulated/totalized value (input)	Analog value (input), measuring	Fixed limit	Sliding / Floating limit	Run time totalization	Event counting	Command execution check	State processing 4)	Plant control	Motor control	Switchover 5)	Step control 5)	Safety / Frost protection control	P control loop	PI / PID control loop	Sliding / Floating / Curve setpoint	Proportional output stage	Proportional to on/off conversion 6)	Proportional to pulse width modulation	Setpoint / Output limitation	Switchover of parameters
DATA POINT Section no.		1					2					3						4					5							
Column no.		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	6	1	2	3	4	5	1	2	3	4	5	6	7	8
Control of Lighting Level			X			X		X																						
Window Opening Level Control			X			X		X																						
Door Opening Level Control			X			X		X																						

Figura 7.1: Tabla con funciones BACS parte 1. Fuente: autores.

Example for heterogeneous Systems

BACS Function List

ISO 16484-3

6) For cooling / heating use 2 x on / off conversions

7) Per input point address

8) E.g. device, schedule, life safety, loop, file (see EN ISO 16484-5)

9) If required, indicate whether applies to client devices "A" Or server devices "B" (see BIBBs)

System : windows, door, and lighting automation system		Processing functions														Management functions				Operator functions			
		Calculation / Optimization																					
		h,x directed control 7)	Arithmetic calculation 7)	Event switching	Time schedule	Optimum start / stop	Duty cycling	Night cooling	Room temperature limitation	Energy recovery 7)	Backup power operation	Mains power recovery pogram	Peak load limitation	Energy tariff dependent switching		Input / output / value object types 9)	Complex object types 8), 9)	Event storage	Historical database	Graphic / static plant schematic	Dynamic display	Event instruction text	Remote messaging
DATA POINT Section no.		6														7				8			
Column no.		1	2	3	4	5	6	7	8	9	10	11	12	13		1	2	3	4	1	2	3	4
Control of Lighting Level																							
Window Opening Level Control																							
Door Opening Level Control																							

Figura 7.2: Tabla con funciones BACS parte 2. Fuente: autores.

7.9. anexo F: Repositorios generados

- **cbasconc1/Generic_Fine_tuning_trainer:** `https://github.com/cbasconc1/Generic_Fine_tuning_trainer`

- **Videos de pruebas:** `https://www.youtube.com/watch?v=Zwrg8VFAJwo`

7.10. anexo G: Pruebas

7.10.0.1. Sujeto de prueba N° 1 - Sebastian Rodriguez con información completa en voz

Dispositivo	Instrucción	Transcripción	Tiempo [s]
Luz derecha	Prende esa luz derecha al máximo	Hola, prende esa luz derecha al máximo	3,7539
	Apaga esa luz de la derecha	Hola, apaga esa luz de la derecha	3,4099
	Prende aquella luz derecha a la mitad	Hola, prende aquella luz de la derecha a la mitad	3,7170
Luz central	Prende esta luz central	Hola, prende esta luz central	4,5332
	Desactiva luz del medio	Hola, desactiva la luz del medio	3,3742
	Activa esa luz central a la mitad	Hola, activa esta luz central a la mitad	3,7718
Luz izquierda	Luz izquierda enciéndela	Hola, luz izquierda enciéndela	3,3719
	Bombilla izquierda enciéndela a la mitad	Hola, bombilla izquierda enciéndela a la mitad	3,5573
	Apagar esa luz izquierda	Hola, apagar esa luz izquierda	2,4690
Ventana derecha	Abre esa ventana derecha	Hola, abre esa ventana derecha	3,5132
	Pon la luz derecha al cincuenta	Hola, pon la luz derecha al cincuenta	3,4204
	Cierra la ventana derecha de allá	Hola, cierra la ventana derecha de allá	4,0955
Ventana central	Abre esa ventana central	Hola, abre esa ventana central	3,6790
	Abre a nivel medio la ventana de la mitad	Hola, ponle luz central a cincuenta	3,9699
	Cerrar la ventana del medio	Hola, cerrar la ventana del medio	3,8131
Ventana izquierda	Abre esta ventana izquierda	Hola, abre esa ventana izquierda	3,6862
	Cierra esa ventana izquierda al cincuenta por ciento	Hola, cierra la ventana izquierda al cincuenta por ciento	3,9685
	Cierra esta ventana izquierda totalmente	Hola, cierro esa ventana izquierda totalmente	3,8659
Puerta	Cerrar esa puerta	Hola, cerrar esa puerta	3,4493
	Abrir puerta a nivel medio	Hola, abrir puerta al nivel medio	3,8245
	Abrí esa puerta	Hola, abrir esa puerta	3,8846
		Promedio	**3,6728**

Tabla 7.1: Medición de persona Sebastian información completa

7.10.0.2. Sujeto de prueba N° 2 - Laura Herrera con información completa en voz

Dispositivo	Instrucción	Transcripción	Tiempo [s]
Luz derecha	Prende esa luz derecha al máximo	Hola, aprende y derecha máximo	5.2090
	Apaga esa luz de la derecha	Hola, de salud de la derecha	4.8500
	Prende aquella luz derecha a la mitad	Hola, aprendí aquella el derecho a la mitad	4.5660
Luz central	Prende esta luz central	Hola, aprende esta luz central	4.0685
	Desactiva luz del medio	Hola, desactivar luz del medio	4.3392
	Activa esa luz central a la mitad	Hola, adquirí central en la mitad	3.7628
Luz izquierda	Luz izquierda enciéndela	Hola, luz izquierda enciéndela	3.8694
	Bombilla izquierda enciéndela a la mitad	Hola, ámbito izquierda siempre el elemento	5.1966
	Apagar esa luz izquierda	Hola, apagar esa luz de izquierda	3.7499
Ventana derecha	Abre esa ventana derecha	Hola, ahora esa ventana derecha	6.0446
	Pon la luz central al cincuenta	Hola, ponerle central al cincuenta	4.8235
	Cierra la ventana derecha de allá	Hola, cierre la ventana derecha	4.9956
Ventana central	Abre esa ventana central	Hola, abre la ventana central	4.2805
	Abre a nivel medio la ventana del medio	Hola, abrí la ventana del medio nivel medio	4.0395
	Cerrar la ventana del medio	Hola, cerrar la ventana del medio	4.1207
Ventana izquierda	Abre esta ventana izquierda	Hola, abre la ventana izquierda	4.2608
	Cierra esa ventana izquierda al cincuenta por ciento	Hola, cierre la ventana izquierda el cincuenta por ciento	4.1688
	Cierra esta ventana izquierda totalmente	Hola, cierra la ventana izquierda totalmente	3.5576
Puerta	Cerrar esa puerta	Hola, cerrar esa puerta	3.5863
	Abrir puerta a nivel medio	Hola, abrir puerta a nivel medio	3.5342
	Abrí esa puerta	Hola, abrí esa puerta	3.8555
		Promedio	**4,3275**

Tabla 7.2: Medición de persona Laura información completa

7.10.0.3. Sujeto de prueba N° 3 - Juan Puerto con información completa en voz

Dispositivo	Instrucción	Transcripción	Tiempo [s]
Luz derecha	Prende esa luz derecha al máximo	Hola, aprende salud derecha al máximo	3.7347
	Apaga esa luz de la derecha	Hola, apaga esa luz de la derecha	3.0068
	Prende aquella luz derecha a la mitad	Hola, prenda que ya los derecho a la mitad	4.1850
Luz central	Prende esta luz central	Hola, aprende sobre los centros	4.1246
	Desactiva luz del medio	Hola, desactiva el medio	3.4740
	Activa esa luz central a la mitad	Hola, tío de salud centro de tratamiento	4.9882
Luz izquierda	Luz izquierda enciéndela	Hola, luz izquierda enciéndela	4.2257
	Bombilla izquierda enciéndela a la mitad	Hola, bombilla izquierda enciende de la la mitad	4.4783
	Apagar esa luz izquierda	Hola, apagar esa luz izquierda	4.4067
Ventana derecha	Abre esa ventana derecha	Hola, abre esa ventana derecho	4.8915
	Pon la luz central al cincuenta	Hola, ponerle un central al cincuenta	3.6426
	Cierra la ventana derecha de allá	Hola, cierra la ventana derecha allá	4.3045
Ventana central	Abre esa ventana central	Hola, abre esa ventana derecha central	4.8112
	Abre a nivel medio la ventana del medio	Hola, ara nivel medio la ventana del medio	3.6050
	Cerrar la ventana del medio	Hola, cerrar ventana del medio	4.0907
Ventana izquierda	Abre esta ventana izquierda	Hola, abre esa ventana izquierda	3.7457
	Cierra esa ventana izquierda al cincuenta por ciento	Hola, cierra la ventana izquierda el cincuenta por ciento	4.6699
	Cierra esta ventana izquierda totalmente	Hola, cierra esa ventana izquierda totalmente	4.2350
Puerta	Cerrar esa puerta	Hola, cerrar esa puerta	3.4147
	Abrir puerta a nivel medio	Hola, a abrir puerta a nivel medio	3.9870
	Abrí esa puerta	Hola, hola abrí puertas hola abrí puerta	4.2493
		Promedio	**4,1081**

Tabla 7.3: Medición de persona Juan información completa

7.10.0.4. Sujeto de prueba N° 4- Felipe Lopez con información completa en voz

Dispositivo	Instrucción	Transcripción	Tiempo [s]
Luz derecha	Prende esa luz derecha al máximo	Hola, prende salud derecha al máximo	3.7230
	Apaga esa luz de la derecha	Hola, apaga esa luz de la derecha	3.6717
	Prende aquella luz derecha a la mitad	Hola, aprende aquella luz de la derecha a la mitad	3.7441
Luz central	Prende esta luz central	Hola, aprende esta luz central	3.3548
	Desactiva luz del medio	Hola, desactiva luz del medio	3.5597
	Activa esa luz central a la mitad	Hola, activa esa luz central a la mitad	3.5566
Luz izquierda	Luz izquierda enciéndela	Hola, luz izquierda enciéndela	3.9113
	Bombilla izquierda enciéndela a la mitad	Hola, bombilla izquierda enciende a la mitad	4.0932
	Apagar esa luz izquierda	Hola, apagar esa luz de izquierda	3.7317
Ventana derecha	Abre esa ventana derecha	Hola, abre esa ventana derecha	3.8225
	Pon la luz central al cincuenta	Hola, ponerle central al cincuenta	3.5494
	Cierra la ventana derecha de allá	Hola, cierra la ventana derecha de allá	4.5224
Ventana central	Abre esa ventana central	Hola, abre esa ventana central	3.9012
	Abre a nivel medio la ventana del medio	Hola, abre a nivel medio la ventana del medio	4.3065
	Cerrar la ventana del medio	Hola, cierra la ventana del medio	4.2567
Ventana izquierda	Abre esta ventana izquierda	Hola, abre esta ventana izquierda	3.9291
	Cierra esa ventana izquierda al cincuenta por ciento	Hola, cierra esta ventana izquierda el cincuenta por ciento	3.8943
	Cierra esta ventana izquierda totalmente	Hola, cierra esta ventana izquierda totalmente	3.7365
Puerta	Cerrar esa puerta	Hola, cerrar esa puerta	3.7215
	Abrir puerta a nivel medio	Hola, abrir puerta a nivel medio	4.0419
	Abrí esa puerta	Hola, abrir esa puerta abrir	4.1719
		Promedio	**3,8666**

Tabla 7.4: Medición de persona Felipe información completa

7.10.0.5. Sujeto de prueba N° 5- Gabriela Corredor con información completa en voz

Dispositivo	Instrucción	Transcripción	Tiempo [s]
Luz derecha	Prende esa luz derecha al máximo	Hola, aprendiza de los derechos al máximo	4.2760
	Apaga esa luz de la derecha	Hola, a padecer luz de la derecha	3.3440
	Prende aquella luz derecha a la mitad	Hola, aprende aquella luz derecha a la mitad	3.6591
Luz central	Prende esta luz central	Hola, prende esta luz central	4.6660
	Desactiva luz del medio	Hola, desactivar luz del medio	3.6054
	Activa esa luz central a la mitad	Hola, activas hay luz central a la mitad	4.1877
Luz izquierda	Luz izquierda enciéndela	Hola, luis izquierdo enciéndela	3.5584
	Bombilla izquierda enciéndela a la mitad	Hola, bombilla izquierda enciéndela a la mitad	3.8344
	Apagar esa luz izquierda	Hola, apagar esa luz izquierda	3.6158
Ventana derecha	Abre esa ventana derecha	Hola, abre esa ventana derecha	3.7652
	Pon la luz central al cincuenta	Hola, con la luz central al cincuenta	5.1574
	Cierra la ventana derecha de allá	Hola, cierra la ventana derecha de allá	3.7632
Ventana central	Abre esa ventana central	Hola, abre esa ventana central	3.9732
	Abre a nivel medio la ventana de la mitad	Hola, abre a nivel medio la ventana de la mitad	4.4017
	Cerrar la ventana del medio	Hola, cerrar la ventana del medio	4.0058
Ventana izquierda	Abre esta ventana izquierda	Hola, abre esta ventana izquierda	3.5633
	Cierra esa ventana izquierda al cincuenta por ciento	Hola, cierra la ventana izquierda el cincuenta por ciento	3.9689
	Cierra esta ventana izquierda totalmente	Hola, cierra esta ventana izquierda totalmente	3.7088
Puerta	Cerrar esa puerta	Hola, cerrar esa puerta	3.3295
	Abrir puerta a nivel medio	Hola, abrir puerta a nivel medio	3.7337
	Abrí esa puerta	Hola, abrir esa puerta	3.3138
		Promedio	**3,8776**

Tabla 7.5: Medición de persona Gabriela información completa

7.11. anexo H: Pruebas información completa

7.11.0.1. Sujeto de prueba N° 1 - Sebastian Rodriguez con información incompleta en voz

Dispositiv	Instrucción	Transcripción	Pose	Gesto	Tiempo [s]
Luz derecha	Prende esa	Hola, prende	Derecha	Arriba-derecha	4.1284
	Apaga esa	Hola, apaga eso	Derecha	Arriba-derecha	3.7990
	Prende aquella a la mitad	Hola, aprende a que la mitad	Derecha	Arriba-derecha	4.1376
Luz central	Prende esta	Hola, aprende esta	Centro	Arriba	4.3840
	Desactiva esa	Hola, desactiva esa	Centro	Arriba	4.5099
	Activa esa luz a la mitad	Hola, actividad reduce a la mitad	Centro	Arriba	7.1063
Luz izquierda	Enciéndela	Hola, enciéndela	Izquierda	Arriba	4.4609
	Enciéndela a la mitad	Hola, enciéndela a la mitad	Centro	Arriba-izquierda	7.3234
	Apaga esa	Hola, apaga esa	Centro	Arriba-izquierda	4.6727
Ventana derecha	Abre esa ventana	Hola, abre esa ventana	Centro	Derecha	4.3132
	Cierra esa ventana al nivel medio	Hola, cierra esa ventana el nivel medio	Centro	Derecha	5.0611
	Cierra esa	Hola, cierra esa	Centro	Derecha	3.9228
Ventana central	Abre esa	Hola, abe manera	Centro	Centro	6.3525
	Abre a nivel medio	Hola, abres al nivel medio	Centro	Centro	5.2991
	Ciérrala	Hola, cierren	Centro	Centro	4.7885
Ventana izquierda	Abre esta	Hola, abre esta	Centro	Izquierda	3.5280
	Cierra esa a la mitad	Hola, sierra me esa a la mitad	Centro	Izquierda	5.9419
	Cierra esta total	Hola, cierra esa total	Centro	Izquierda	4.1096
Puerta	Cerrar esa	Hola, cierre	Centro	Izquierda-atrás	3.7805
	Abre a la mitad	Hola, abre a la mitad	Centro	Izquierda-atrás	3.5355
	Abrí esa	Hola, esa	Centro	Izquierda-atrás	3.8070
				Promedio	**4,7124**

Tabla 7.6: Medición de persona Sebastian completo

7.11.0.2. Sujeto de prueba N° 2 - Laura Herrera con información incompleta en voz

Dispositiv	Instrucción	Transcripción	Pose	Gesto	Tiempo [s]
Luz derecha	Prende esa	Hola, prende esa	Derecha	Arriba-derecha	3.9251
	Apaga esa	Hola, apaga eso	Centro	Arriba-derecha	4.6620
	Prende aquella a la mitad	Hola, a aprender la mitad	Centro	Arriba-derecha	3.3095
Luz central	Prende esta	Hola, aprende esta	Centro	Arriba	3.2127
	Desactiva esa	Hola, desactiva esa	Centro	Arriba	3.3590
	Activa esa luz a la mitad	Hola, actividad reduce a la mitad	Centro	Arriba	4.3882
Luz izquierda	Enciéndela	Hola, enciéndela	Izquierda	Arriba	3.5277
	Enciéndela a la mitad	Hola, enciéndela a la mitad	Izquierda	Arriba	3.5236
	Apaga esa	Hola, apaga esa	Izquierda	Arriba	3.5138
Ventana derecha	Abre esa ventana	Hola, abre esa ventana	Centro	Derecha	3.5927
	Cierra esa ventana al nivel medio	Hola, cierra esa ventana a nivel medio	Centro	Derecha	9.9117
	Cierra esa	Hola, cierra esa	Centro	Derecha	3.9156
Ventana central	Abre esa	Hola, abe manera	Centro	Izquierda	4.1932
	Abre a nivel medio	Hola, abres al nivel medio	Centro	Centro	4.1337
	Ciérrala	Hola, cierren	Izquierda	Derecha	3.5674
Ventana izquierda	Abre esta	Hola, abre esta	Centro	Izquierda	3.8480
	Cierra esa a la mitad	Hola, es a la mitad	Izquierda	Centro	4.0145
	Cierra esta ventana total	Hola, cierra esa ventana total	Izquierda	Centro	3.8071
Puerta	Cerrar esa	Hola, cierre	Centro	Izquierda-atrás	3.4244
	Abre a la mitad	Hola, abre a la mitad	Centro	Izquierda-atrás	3.8862
	Abrí esa	Hola, esa	Centro	Izquierda-atrás	3.3231
				Promedio	**4.0494**

Tabla 7.7: Medición de persona Laura completo

7.11.0.3. Sujeto de prueba N° 3 - Juan Puerto con información incompleta en voz

Dispositiv	Instrucción	Transcripción	Pose	Gesto	Tiempo [s]
Luz derecha	Prende esa	Hola, prende esa	Derecha	Arriba-derecha	3.9251
	Apaga esa	Hola, apaga eso	Centro	Arriba-derecha	4.6620
	Prende aquella a la mitad	Hola, a aprender la mitad	Centro	Arriba-derecha	3.3095
Luz central	Prende esta	Hola, aprende esta	Centro	Arriba	3.2127
	Desactiva esa	Hola, desactiva esa	Centro	Arriba	3.3590
	Activa esa luz a la mitad	Hola, actividad reduce a la mitad	Centro	Arriba	4.3882
Luz izquierda	Enciéndela	Hola, enciéndela	Izquierda	Arriba	3.5277
	Enciéndela a la mitad	Hola, enciéndela a la mitad	Izquierda	Arriba	3.5236
	Apaga esa	Hola, apaga esa	Izquierda	Arriba	3.5138
Ventana derecha	Abre esa ventana	Hola, abre esa ventana	Centro	Derecha	3.5927
	Cierra esa ventana al nivel medio	Hola, cierra esa ventana a nivel medio	Centro	Derecha	9.9117
	Cierra esa	Hola, cierra esa	Centro	Derecha	3.9156
Ventana central	Abre esa	Hola, abe manera	Centro	Izquierda	4.1932
	Abre a nivel medio	Hola, abres al nivel medio	Centro	Centro	4.1337
	Ciérrala	Hola, cierren	Izquierda	Derecha	3.5674
Ventana izquierda	Abre esta	Hola, abre esta	Centro	Izquierda	3.8480
	Cierra esa a la mitad	Hola, es a la mitad	Izquierda	Centro	4.0145
	Cierra esta ventana total	Hola, cierra esa ventana total	Izquierda	Centro	3.8071
Puerta	Cerrar esa	Hola, cierre	Centro	Izquierda-atrás	3.4244
	Abre a la mitad	Hola, abre a la mitad	Centro	Izquierda-atrás	3.8862
	Abrí esa	Hola, esa	Centro	Izquierda-atrás	3.3231
				Promedio	**3.8017**

Tabla 7.8: Medición de persona Juan completo

7.11.0.4. Sujeto de prueba N° 4 - Felipe Lopez con información incompleta en voz

Dispositiv	Instrucción	Transcripción	Pose	Gesto	Tiempo [s]
Luz derecha	Prende esa	Hola aprenderla	Derecha	Arriba-derecha	3.6025
	Apaga esa	Hola vaga esa	Derecha	Arriba-derecha	4.3520
	Prende aquella a la mitad	Hola aprendí aquella es la mitad	Derecha	Arriba-derecha	2.4076
Luz central	Prende esta	Hola aprende estas	Centro	Arriba-derecha	4.7869
	Desactiva esa	Hola desactiva esas	Izquierda	Arriba derecha	5.0718
	Activa esa luz a la mitad	Hola actividad esa luz a la mitad	Izquierda	Arriba derecha	5.7467
Luz izquierda	Enciendela	Hola aprender esta	Izquierda	Arriba-izquierda	3.9736
	Enciendela a la mitad	Hola entre la mitad	Izquierda	Arriba-izquierda	3.3665
	Apagar esa	Hola apaga esa	Izquierda	Arriba-izquierda	3.5683
Ventana derecha	Abre esa ventana	Hola haré esa	Centro	Derecha	4.5289
	Cierra esa ventana al nivel medio	Hola sierra está a la mitad	Derecha	Derecha	4.0360
	Cierra esa	Hola apaga esa	Centro	Derecha	3.7385
Ventana central	Abre esa	Hola abre esa	Centro	Centro	5.3869
	Abre a nivel medio	Hola cierren la ventana nivel medio	Centro	Centro	3.8733
	Ciérrala	Hola cierra esa	Centro	Centro	4.1179
Ventana izquierda	Abre esta	Hola abre esa	Centro	Izquierda	3.7572
	Cierra esa a la mitad	Hola cierres a la mitad	Centro	Izquierda	4.1327
	Cierra esta ventana total	Hola cierres	Centro	Izquierda	4.4081
Puerta	Cerrar esa	Hola cerrar esa	Centro	Izquierda-atrás	3.9214
	Abre a la mitad	Hola abre a la mitad	Centro	Izquierda-atrás	3.8442
	Abrí esa	Hola abrir esa	Centro	Izquierda-atrás	3.6523
				Promedio	**4,3445**

Tabla 7.9: Medición de persona Felipe completo

Dispositiv	Instrucción	Transcripción	Pose	Gesto	Tiempo [s]
Luz derecha	Prende esa	Hola, prende esa	Derecha	Arriba-derecha	3.9251
	Apaga esa	Hola, apaga eso	Centro	Arriba-derecha	4.6620
	Prende aquella a la mitad	Hola, a aprender la mitad	Centro	Arriba-derecha	3.3095
Luz central	Prende esta	Hola, aprende esta	Centro	Arriba	3.2127
	Desactiva esa	Hola, desactiva esa	Centro	Arriba	3.3590
	Activa esa luz a la mitad	Hola, actividad reduce a la mitad	Centro	Arriba	4.3882
Luz izquierda	Enciéndela	Hola, enciéndela	Izquierda	Arriba	3.5277
	Enciéndela a la mitad	Hola, enciéndela a la mitad	Izquierda	Arriba	3.5236
	Apaga esa	Hola, apaga esa	Izquierda	Arriba	3.5138
Ventana derecha	Abre esa ventana	Hola, abre esa ventana	Centro	Derecha	3.5927
	Cierra esa ventana al nivel medio	Hola, cierra esa ventana a nivel medio	Centro	Derecha	9.9117
	Cierra esa	Hola, cierra esa	Centro	Derecha	3.9156
Ventana central	Abre esa	Hola, abe manera	Centro	Izquierda	4.1932
	Abre a nivel medio	Hola, abres al nivel medio	Centro	Centro	4.1337
	Ciérrala	Hola, cierren	Izquierda	Derecha	3.5674
Ventana izquierda	Abre esta	Hola, abre esta	Centro	Izquierda	3.8480
	Cierra esa a la mitad	Hola, es a la mitad	Izquierda	Centro	4.0145
	Cierra esta ventana total	Hola, cierra esa ventana total	Izquierda	Centro	3.8071
Puerta	Cerrar esa	Hola, cierre	Centro	Izquierda-atrás	3.4244
	Abre a la mitad	Hola, abre a la mitad	Centro	Izquierda-atrás	3.8862
	Abrí esa	Hola, esa	Centro	Izquierda-atrás	3.3231
				Promedio	**3.8017**

Tabla 7.10: Medición de persona Juan completo

7.11.0.5. Sujeto de prueba N° 4 - Felipe Lopez con información incompleta en voz

Dispositiv	Instrucción	Transcripción	Pose	Gesto	Tiempo [s]
Luz derecha	Prende esa	Hola aprenderla	Derecha	Arriba-derecha	3.6025
	Apaga esa	Hola vaga esa	Derecha	Arriba-derecha	4.3520
	Prende aquella a la mitad	Hola aprendí aquella es la mitad	Derecha	Arriba-derecha	2.4076
Luz central	Prende esta	Hola aprende estas	Centro	Arriba-derecha	4.7869
	Desactiva esa	Hola desactiva esas	Izquierda	Arriba derecha	5.0718
	Activa esa luz a la mitad	Hola actividad esa luz a la mitad	Izquierda	Arriba derecha	5.7467
Luz izquierda	Enciendela	Hola aprender esta	Izquierda	Arriba-izquierda	3.9736
	Enciendela a la mitad	Hola entre la mitad	Izquierda	Arriba-izquierda	3.3665
	Apagar esa	Hola apaga esa	Izquierda	Arriba-izquierda	3.5683
Ventana derecha	Abre esa ventana	Hola haré esa	Centro	Derecha	4.5289
	Cierra esa ventana al nivel medio	Hola sierra está a la mitad	Derecha	Derecha	4.0360
	Cierra esa	Hola apaga esa	Centro	Derecha	3.7385
Ventana central	Abre esa	Hola abre esa	Centro	Centro	5.3869
	Abre a nivel medio	Hola cierren la ventana nivel medio	Centro	Centro	3.8733
	Ciérrala	Hola cierra esa	Centro	Centro	4.1179
Ventana izquierda	Abre esta	Hola abre esa	Centro	Izquierda	3.7572
	Cierra esa a la mitad	Hola cierres a la mitad	Centro	Izquierda	4.1327
	Cierra esta ventana total	Hola cierres	Centro	Izquierda	4.4081
Puerta	Cerrar esa	Hola cerrar esa	Centro	Izquierda-atrás	3.9214
	Abre a la mitad	Hola abre a la mitad	Centro	Izquierda-atrás	3.8442
	Abrí esa	Hola abrir esa	Centro	Izquierda-atrás	3.6523
				Promedio	**4,3445**

Tabla 7.11: Medición de persona Felipe completo

7.11.0.6. Sujeto de prueba N° 5 - Gabriela Corredor con información incompleta en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz derecha	Prende esa Apaga esa	Hola, aprende esa Una apaga eso
Luz izquierda	Apagar esa	Hola, a pagar ir
Ventana central	Abre a nivel medio	Hola
Ventana izquierda	Cierra esa total	Hola, si registra total Hola si redes o twitter Hola Hola sierra es pla total
Puerta Puerta	Cerrar esa Abre a la mitad	Hola, Ferrera Se realizan Obrera mitad

Tabla 7.12: Transcripciones erroneas Sebastian información incompleta en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz central	Prende esta luz central Una prende esta luz central Hola prende esta luz empeorado	Hola, prende esta luz central Una prende esta luz central Hola, prende esta luz empeorado
Luz izquierda	Luz izquierda enciéndela Una apagar esa luz izquierda	Hola, atacar hizo luz izquierda Una apagar esa luz izquierda
Ventana derecha	Abre esa ventana derecha Esa ventana derecha Cierra ventana derecha de allá Cierre la ventana derecha de allá	Una abre esa ventana derecha Esa ventana derecha Una cierra la ventana derecha Cierre la ventana derecha de allá
Puerta	Abrí puerta	Hola, abrí esa puerta

Tabla 7.13: Transcripciones erróneas Laura información completa en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz derecha	Prende aquella luz derecha a la mitad Hola, me interesa la mitad	Hola, prensa entrevista a la mitad Hola, me interesa la mitad
Luz central	Desactiva luz del medio Activa esa luz central a la mitad Pon la luz central al cincuenta	Hola, decepción luz del medio Reciba luz del medio O las luces encendidas
Ventana derecha	Abre esa ventana derecha Cierra la ventana derecha de allá	Hola, esa ventana de derecho Cierra la ventana de derecho de allá
Ventana izquierda	Cierra esa ventana izquierda al cincuenta por ciento	Hola, ciertamente una izquierda el cincuenta por ciento
Puerta	Abrir puerta a nivel medio	Hola, abreu puedes repetir en medio

Tabla 7.14: Transcripciones erróneas Juan información incompleta en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz derecha Luz central	Prende esa Apaga esa Prende aquella a la mitad Prende esta Desactiva esa Activa esa luz a la mitad	Aprender La apaga ella Aprenderá a la mitad Aprende de esta Desactivar Activa esa luz en la mitad
Luz izquierda	Enciendela	La efectividad de esta
Ventana derecha	Apagar esa Cierra esa ventana al nivel medio	Hola, atacar esa La cierra esa ventana al nivel medio
Ventana izquierda	Cierra esa a la mitad	Esa es la mitad
Puerta	Abrí esa	Limpieza

Tabla 7.15: Transcripciones erróneas Felipe información incompleta en voz

Dispositivo	Instrucción	Transcripción
Luz derecha	Apaga esa Prende aquella a la mitad Hola, prende ella La prende a que la mitad	La apaga ella Aprenderá a la mitad Hola, prende a que la mitad Aprende a aquella a la mitad
Luz central	Hola, desactiva esa Hola, activa esta La efectividad de esta	La ciudad está Hola, activa esta
Luz izquierda	Enciéndela Apaga esa Apaga esta	Aprende vendetta Hola, a pagar
Ventana izquierda Hola, ferrari	Cierra esa	Hola, sierra esto

Tabla 7.16: Transcripciones erróneas Gabriela información incompleta en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz derecha	Apaga esa	Hola, paga eso
Luz central	Activa esa luz a la mitad	La activación hice la mitad
Ventana derecha Ventana central	Cierra esa ventana al nivel medio Cierra esa Ciérrala Hola, ciérrela	Hola, sierra es Edson Hola, sierra eso Hola, ciérrela Hola, ciérrela
Ventana izquierda	Cierra esta ventana total	Hola, soy referente natural
Puerta	Cerrar esa	Hola, Larissa Hola, se es

Tabla 7.17: Transcripciones erróneas Sebastian información completa en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz central	Desactiva luz del medio Hola, activa	Hola, efectiva en el medio Hola, activa
Ventana derecha	Cierra la ventana derecha de allá	Pero la a la derecha de young
Ventana central	Abre esa ventana central Hola, bren en medio de la ventana del medio	Hola, ahora oreja linterna central Hola, bren en medio de la ventana del medio
Ventana izquierda	Abre esta ventana izquierda Cierra esa ventana izquierda al cincuenta por ciento Cierra esa ventana izquierda al cincuenta por ciento Cierra esta ventana izquierda totalmente	Hola, agresor en plena izquierda Hola, soy restaurante a la izquierda el cincuenta Hola, si referente a la izquierda el cincuenta por ciento Hola, si a la izquierda totalmente
Puerta	Abrí esa puerta Hola, abrí el soporte	Colador y tuerto nivel medio Hola, abrí el soporte

Tabla 7.18: Transcripciones erróneas Laura información completa en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz derecha	Prende aquella a la mitad	Hola, prenda que ya la mitad
Luz central	Activa esa luz a la mitad Hola, activa esa luz a ver	Esa luz a la mitad Iba esa luz a la mitad
Luz izquierda	Enciéndela Apagar esa	Sí, enciéndela Pagar esa
Ventana derecha	Abre esa ventana Cierra esa ventana al nivel medio	Abre esa Cierre la el nivel medio
Ventana central	Abre esa	Abre esa
Puerta	Cierra esa	Hola, Torres

Tabla 7.19: Transcripciones erróneas Juan información completa en voz

Dispositivo	Instrucción	Instrucción Transcrita
Luz derecha	Apaga esa luz de la derecha	Apaga la luz de la derecha
Luz izquierda	Bombilla izquierda enciendela a la mitad La bombilla izquierda enciende a la mitad	Abuela, la bombilla izquierda enciende a la mitad La bombilla izquierda enciende a la mitad
Ventana central	Cierra la ventana de la mitad En la ventana derecha de allá	Fue la zona central del cincuenta Hola, cierra la ventana del hábitat
Ventana izquierda	Cierra esta ventana izquierda totalmente	Desde que Paula cierre esta ventana izquierda totalmente
Puerta	Cerrar esa puerta	No quería cerrar esa puerta

Tabla 7.20: Transcripciones erróneas Felipe información completa en voz

Dispositivo	Instrucción	Transcripción
Luz derecha	Prende aquella luz derecha a la mitad	Hola, prende aquella luz derecha a la mitad
Ventana derecha	Abre esa ventana derecha	Abre esa ventana derecha
Luz central Ventana central	Pon la luz central al cincuenta Abre esa ventana central De esa ventana central Cerrar la ventana del medio	Hola, con la luz entrar al cincuenta Hola De esa ventana central Cierra la ventana de la mitad
Ventana izquierda	Cierra esta ventana izquierda totalmente	Hola, sierra esta ventana izquierda totalmente
Puerta	Abrir puerta a nivel medio	Hola, abri berta nivel medio

Tabla 7.21: Transcripciones erróneas Gabriela información completa en voz