industriales
etsii

Escuela Técnica Superior de Ingeniería Industrial

UNIVERSIDAD POLITÉCNICA DE CARTAGENA

Escuela Técnica Superior de Ingeniería Industrial

Arquitectura cognitiva para integración de usuarios humanos dentro de un sistema de mapas multinivel en una plataforma robótica asistencial

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

Autor: **Marta Jiménez Muñoz**
Director: Nieves Pavón Pulido
Codirector: Jorge Juan Feliu Batlle

Cartagena, 16 de julio de 2024.

Universidad Politécnica de Cartagena

AGRADECIMIENTOS

Al equipo de JUNO+. A Nieves, Juan Antonio, Jorge y Jesús Damián, por toda la ayuda y las oportunidades brindadas, y a Pablo, Adrián y Gabriel por la compañía en el laboratorio y, sobre todo, por prestarme sus caras para entrenar el modelo de reconocimiento.

A mis padres, por apoyarme y creer en mí dos años más, aunque cada vez hable más de cosas que son más difíciles de entender. A mi hermana, por ser un referente del trabajo duro y de la constancia.

A Ana, Clara, Vic y Javi, por estar siempre.

RESUMEN

En este Trabajo Fin de Máster se ha abordado el diseño y el desarrollo de una aproximación a una arquitectura cognitiva para la identificación y ubicación de usuarios humanos de una plataforma robótica dentro de un mapa semántico. El sistema permite registrar y entrenar el modelo de reconocimiento facial con nuevos usuarios. Se genera un mapa de obstáculos con la ubicación de las personas más cercanas y los objetos detectados, al mismo tiempo que se crea un mapa semántico con la ubicación de cada persona siguiendo un código de colores. La arquitectura cognitiva, a través de un mapa semántico con una estructura de datos, es capaz de memorizar a los usuarios reconocidos junto a las coordenadas y la franja horaria correspondiente, así como de olvidar a aquellos que no ha reconocido desde hace tiempo en los lugares donde solían estar. Esta aproximación a una arquitectura cognitiva ofrece la posibilidad de poder predecir la ubicación de una persona buscada en un momento concreto, lo que habilita al robot para realizar tareas de vigilancia y búsqueda. Este sistema ha sido probado con un robot asistencial real en un entorno de laboratorio, quedando preparado para una fase de despliegue en el entorno real de una residencia de mayores.

ÍNDICE GENERAL.

ÍNDICE DE FIGURAS.

1 INTRODUCCIÓN.

En este capítulo se presentan los aspectos clave que han dado lugar al desarrollo de este TFM (Trabajo Fin de Máster). Se procede a describir los principales objetivos del proyecto junto a las razones más importantes que han motivado su desarrollo. El contenido de los capítulos que forman parte de esta documentación se describe también a continuación.

1.1 Antecedentes.

El desarrollo de plataformas robóticas asistenciales ha ganado un interés significativo en los últimos años debido al aumento de la población envejecida y la necesidad de mejorar la eficiencia en el cuidado de personas mayores. La integración de tecnologías avanzadas, como la Inteligencia Artificial y la Robótica, se presenta como una solución prometedora para abordar los desafíos asociados con la atención en residencias de mayores.

Uno de los avances más relevantes en este campo es el desarrollo de sistemas de detección y reconocimiento facial. Estos sistemas han sido ampliamente investigados y aplicados en diversos contextos, desde la seguridad y vigilancia hasta la interacción humano-robot. Tecnologías como FaceNet (Schroff et al., 2015) han demostrado ser altamente precisas y robustas en la identificación de personas, aunque enfrentan desafíos relacionados con el retraso en el procesamiento y la necesidad de hardware potente.

La visión artificial, un componente crucial para la detección y el reconocimiento facial, ha experimentado mejoras significativas gracias a la evolución de los algoritmos y el aumento en la capacidad de procesamiento. Los modelos de aprendizaje automático, como los basados en redes neuronales convolucionales (CNN), han permitido que los sistemas de visión artificial identifiquen y clasifiquen objetos y personas con una precisión sin precedentes.

En el contexto de residencias de mayores, la sobrecarga del personal es un problema crítico. Los trabajadores enfrentan una gran cantidad de tareas rutinarias y repetitivas, lo que puede afectar la calidad del cuidado y la atención proporcionada. La introducción de plataformas robóticas asistenciales con capacidades avanzadas de detección y reconocimiento facial puede aliviar esta carga, permitiendo que el personal se concentre en tareas más importantes y mejorando la calidad de vida de los residentes.

Los sistemas de mapas semánticos han sido utilizados en robótica para mejorar la navegación y la interacción con el entorno. Estos sistemas permiten que los robots comprendan y respondan a su entorno de manera más inteligente, utilizando información almacenada en celdas del mapa. La integración de personas reconocidas en

estos mapas proporciona una representación más rica y contextual del entorno, mejorando las capacidades de vigilancia y búsqueda de personas.

Inspirado en la organización de la memoria humana (Kotseruba & Tsotsos, 2020), el diseño de sistemas que estructuren la información de manera eficiente es fundamental para el desarrollo de arquitecturas cognitivas avanzadas. Estos sistemas deben ser capaces de almacenar y olvidar información de manera similar a como lo hace el cerebro humano, permitiendo una gestión más eficiente de los datos y mejorando la capacidad de respuesta del sistema.

En resumen, los antecedentes en el desarrollo de tecnologías de detección y reconocimiento facial, visión artificial, y sistemas de mapas semánticos, junto con la necesidad urgente de soluciones para la sobrecarga del personal en residencias de ancianos, establecen un marco sólido para la implementación de una arquitectura cognitiva avanzada en plataformas robóticas asistenciales. Esta arquitectura promete mejorar significativamente la interacción, la seguridad y la calidad de los servicios en estos entornos, abordando los desafíos actuales y aportando soluciones innovadoras.

1.2 Motivación y objetivos.

La creciente demanda de servicios asistenciales de alta calidad en residencias para personas mayores ha puesto de manifiesto la necesidad de soluciones tecnológicas innovadoras que puedan aliviar la carga del personal y mejorar la calidad de vida de los residentes. En este contexto, la implementación de plataformas robóticas asistenciales con capacidades avanzadas de reconocimiento y mapeo se presenta como una solución prometedora (Pavón-Pulido et al., 2023). La motivación principal de este trabajo radica en la necesidad de desarrollar una arquitectura cognitiva que permita la integración eficaz de usuarios humanos dentro de un sistema de mapas multinivel, mejorando así la interacción y la seguridad en entornos asistenciales.

El desarrollo de una arquitectura cognitiva en una plataforma robótica asistencial puede proporcionar múltiples beneficios. Por un lado, permite automatizar tareas rutinarias, liberando tiempo para que el personal pueda enfocarse en aspectos más críticos del cuidado. Por otro lado, mejora la seguridad mediante la vigilancia continua y la capacidad de búsqueda de personas, garantizando un entorno seguro y controlado.

Para alcanzar estos beneficios, se han establecido los siguientes objetivos:

- Diseño de un sistema de detección facial como base de la arquitectura: Este objetivo se enfoca en desarrollar un sistema robusto y eficiente para la detección de rostros, que servirá como fundamento para las etapas subsecuentes de reconocimiento y mapeo.
- Diseño de un sistema de reconocimiento facial utilizando el sistema de detección facial previamente desarrollado: Una vez desarrollado el sistema de detección, se procederá a implementar un sistema de reconocimiento facial que pueda identificar y distinguir a las personas en el entorno.

- Ubicación de los rostros reconocidos en un mapa semántico: Este objetivo implica la integración de la información de los rostros reconocidos en un mapa semántico, lo cual permitirá una representación más detallada y contextual del entorno.
- Diseño de un sistema basado en la organización de la memoria humana para estructurar la información obtenida: Inspirado en el funcionamiento de la memoria humana, este sistema organizará y gestionará la información de manera eficiente, facilitando el acceso y el procesamiento de los datos.
- Integración del sistema en la plataforma robótica asistencial: La implementación de todos los componentes desarrollados en la plataforma robótica JUNO+ permitirá evaluar su funcionamiento en un entorno real, asegurando su compatibilidad y eficacia.
- Validación del sistema final en laboratorio: Antes de su implementación definitiva en entornos asistenciales, el sistema será validado en un entorno controlado de laboratorio para verificar su rendimiento, precisión y fiabilidad.

En resumen, este trabajo busca mejorar significativamente la interacción, la seguridad y la calidad de los servicios en plataformas robóticas asistenciales. La validación del sistema en laboratorio asegurará que esté listo para su implementación en residencias de atención a personas mayores, aportando soluciones innovadoras y efectivas a los desafíos actuales en este ámbito.

1.3 Distribución de capítulos.

La memoria de este proyecto contiene los siguientes capítulos:

1) **Introducción**: El capítulo actual describe la motivación y objetivos que se han propuesto
2) **Estado del arte**: Resumen acerca de las tecnologías disponibles más relevantes para dar contexto a este proyecto. Se abordan tres temas principalmente, que consisten en describir el concepto de arquitectura cognitiva, así como presentar sus diferentes capacidades y dos ejemplos, en explicar el concepto de mapas en la robótica móvil y las técnicas más empleadas para ello y, por último, exponer el funcionamiento general de los sistemas de reconocimiento facial junto a los algoritmos más conocidos.
3) **Diseño del sistema**: Se lleva a cabo un análisis de los componentes principales del sistema, además del diseño e implementación de estos. Se presentan los problemas planteados y el procedimiento a seguir hasta su solución con el objetivo de facilitar la compresión de todas las partes que finalmente forman parte de la arquitectura.
4) **Resultados**: En este capítulo se describen los resultados globales del proyecto y las pruebas de validación realizadas.
5) **Conclusiones**: Se detallan las principales conclusiones obtenidas durante el desarrollo del proyecto y se identifican las posibles áreas de mejora para futuras líneas.

2 ESTADO DEL ARTE.

En este capítulo se presentan las principales tecnologías empleadas y relacionadas con este TFE, pues se incluyen los conceptos y ejemplos principales de las arquitecturas cognitivas y de los mapas multinivel, así como de los sistemas de detección y reconocimiento facial.

2.1 Arquitectura cognitiva.

Una arquitectura cognitiva, cuyos comienzos dentro de la investigación de la Inteligencia Artificial tuvo lugar en 1950 (Kotseruba & Tsotsos, 2020), es el conjunto esencial de componentes que hace de infraestructura de un sistema inteligente. Esta, puede incluir aspectos constantes en el tiempo y en múltiples dominios de aplicación (Langley et al., 2009). Se puede decir que una arquitectura cognitiva aporta, a un sistema inteligente, la capacidad de resolver tareas de forma parecida al comportamiento humano (*Andres-Ubeda_Practicas_RoboticaServicios.pdf*, s. f.).

Dentro de los componentes pueden encontrarse las memorias a corto y a largo plazo, encargadas de almacenar la información sobre conocimientos, objetivos y creencias, la representación de los elementos que forman parte de los "recuerdos" de la memoria y su estructura de organización y los procesos con los que actúa, ya sean mecanismos de desempeño o de aprendizaje. (Langley et al., 2009).

En la actualidad, existen cientos de arquitecturas cognitivas y pueden clasificarse en cuatro tipos diferentes: sistemas que piensan como los humanos, sistemas que actúan como los humanos, sistemas que piensan de forma racional y sistemas que actúan de tal forma. Por ejemplo, una arquitectura cognitiva que piense como un humano debería cometer los mismos errores que cometería un humano en su misma situación, mientras que, si se trata de un sistema racional, este debería ofrecer soluciones robustas independientemente del dominio en el que se encuentre.

Por otro lado, existe una forma de agrupar las arquitecturas en función de cómo procesan y representan la información, encontrándose principalmente divididas en tres grupos: simbólicas, emergentes e híbridas.

Las arquitecturas simbólicas usan símbolos para representar los conceptos, de manera que estos pueden ser manipulados a través de un conjunto de instrucciones como normas de decisión *if-else*. Se trata de uno de los sistemas más comunes gracias a lo natural e intuitiva que es su representación. Este tipo de arquitecturas destacan por su capacidad de planificación y razonamiento, aunque su punto débil se encuentra en la flexibilidad y la robustez.

Los sistemas emergentes, en cambio, se adaptan a las situaciones y aprenden de ellas construyendo grandes modelos paralelos, donde la información es representada en un diagrama de flujo con la propagación de diferentes señales de parte de los nodos de

entrada. Esto hace que, aunque sean más sencillas de diseñar que las simbólicas, deben ser entrenadas para que generen el comportamiento deseado, tarea realmente difícil y que puede verse afectada por el continuo entrenamiento para lograr nuevos comportamientos.

Por las ventanas y desventajas que estos dos tipos de arquitecturas presentan, se trata de combinar elementos de ambas en los sistemas híbridos. Son los más elegidos, aunque a veces resulta difícil encasillarlas en alguno de los grupos mencionados. (Kotseruba & Tsotsos, 2020)

En la Figura 1, se puede ver una clasificación de las arquitecturas cognitivas más relevantes donde, además, las emergentes se dividen en función de si implementan modelos biológicos neuronales o si están diseñadas de una forma más similar a las redes neuronales artificiales. Las arquitecturas híbridas aparecen separadas por aquellas que incluyen como mínimo un módulo subsimbólico para realizar el procesamiento sensorial, incluyéndose el resto de los tipos de hibridación en la categoría alternativa.

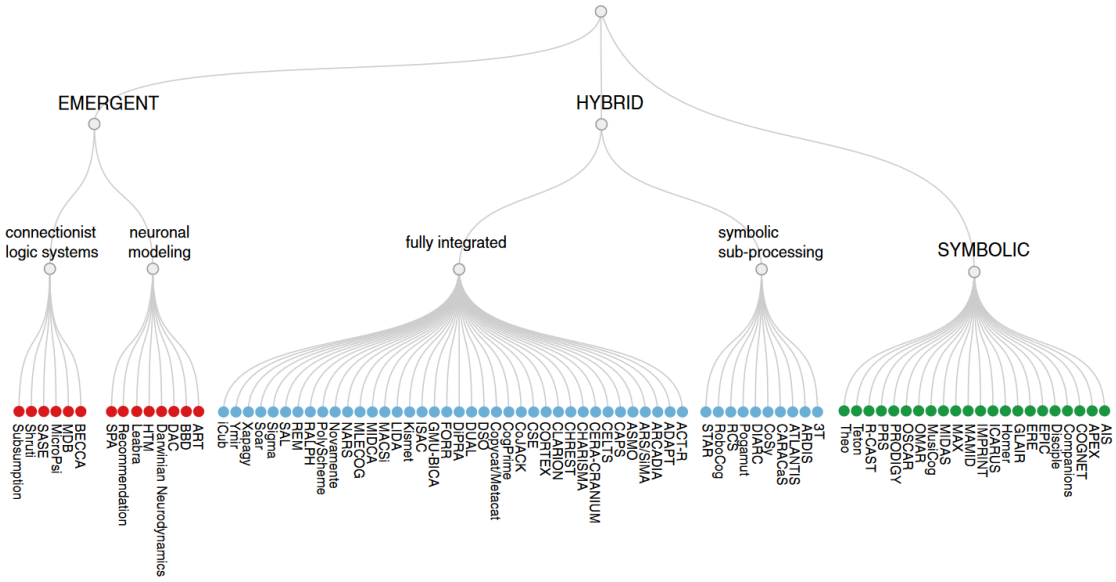

Figura 1. Clasificación de arquitecturas cognitivas según la representación y el procesamiento. (Kotseruba & Tsotsos, 2020)

La tendencia actual indica que las arquitecturas cognitivas híbridas van a crecer aún más de lo que ya lo han hecho en los últimos años.

2.1.1 Capacidades de las arquitecturas cognitivas

Para diseñar una arquitectura cognitiva es importante resolver la manera en la que esta va a acceder a las diferentes fuentes de conocimiento. Esto es posible gracias a capacidades como la percepción, que permite conocer el entorno, la planificación, el razonamiento y la predicción, que aportan conocimiento sobre la situación actual o la memoria, que proporciona el conocimiento del pasado. (Langley et al., 2009)

2.1.1.1 Reconocimiento y categorización

El reconocimiento en términos de sistemas inteligentes consiste en la capacidad de identificar patrones, ya sean de situaciones estáticas o no. Es inherente a la categorización, que incluye la asignación de objetos, situaciones y eventos a conceptos ya conocidos. Al mismo tiempo, está íntimamente relacionado a la percepción.

Esta capacidad es posible lograrla al representar situaciones patrón en la memoria de forma general, para que puedan aplicarse a situaciones parecidas a partir de un grado de coincidencia.

Para que se complete la implementación de esta capacidad, una arquitectura cognitiva debe poder también aprender patrones nuevos a raíz de las situaciones vividas, así como poder refinar los que ya existen cuando sea necesario. (Langley et al., 2009)

2.1.1.2 Percepción, evaluación y atención

Los diferentes tipos de entradas empleadas por las arquitecturas empleadas para generar las respuestas de comportamiento son los que permiten la integración del sistema en un entorno externo. Estas se obtienen a través del componente de la percepción, capaz de transformar datos en crudo al sistema.

Al igual que los seres humanos y sus sentidos, estos sistemas pueden disponer de visión, olfato, gusto, tacto, percepción de la temperatura, nocicepción (percepción del dolor) o propiocepción (capacidad de saber la posición exacta de todas las partes de un sistema) entre muchos otros. Por otro lado, también existen arquitecturas cuyas entradas no tienen nada que ver con los sentidos humanos e incluyen simplemente interfaces de usuarios, entradas de teclado y sensores como LiDAR, acelerómetros o láseres. (Kotseruba & Tsotsos, 2020)

Se trata de un término muy extenso que incumbe muchos tipos de procesamiento, donde destaca la cuestión de la atención, es decir, la capacidad de decidir a qué dedicarle sus recursos perceptibles para así poder discriminar la información más relevante del entorno, aun teniendo en cuenta todo el ruido que pueda venir de este. Este problema, puede resolverse introduciendo mecanismos de reducción de información, capaces de saber qué sensores utilizar, así como cuándo y dónde. (Langley et al., 2009)

Existen tres tipos de mecanismos diferentes: selección (elegir uno de entre muchos), restricción (elegir algunos de entre muchos) y supresión (eliminar algunos de muchos), siendo este último, el menos frecuente. Cada uno de estos está, a su vez, dividido en subtipos. Los más comunes, por ejemplo, son el de selección dentro de la clase de *world model* (selecciona eventos y/u objetos en los que centrarse), presente en todos los sistemas de visión, y *viewpoint/gaze selection* (permite centrarse en una región del entorno y adquirir más información sobre esta, recorriéndola y viéndola desde diferentes ángulos), presente en los sistemas con cámaras y muy necesario para los sistemas de visión activa. (Kotseruba & Tsotsos, 2020)

Hay que destacar, por último, que depende del reconocimiento y de la categorización del entorno en conjunto con la percepción de objetos y condiciones, para contextualizar por completo la situación del entorno. (Langley et al., 2009)

2.1.1.3 Selección de acción

Para asegurar el funcionamiento dentro de un entorno complejo, las arquitecturas cognitivas deben ser capaces de tomar ciertas decisiones y realizar selecciones de alternativas (Langley et al., 2009), ya sea en referencia a un objetivo, a una tarea o a comandos para el hardware (Kotseruba & Tsotsos, 2020).

Existen dos maneras principales para realizar la selección de la acción. La primera de ellas es la selección de acción de planificación, que consiste en el uso de algoritmos de IA tradicionales con el fin de poder establecer los siguientes pasos que son necesarios para así lograr un objetivo en concreto o resolver algún problema que se haya planteado. Este método es más utilizado en las arquitecturas simbólicas, en las que la tarea se descompone en subtareas sucesivamente.

Por otro lado, se encuentra la selección de acciones dinámica, donde la acción se elige de entre todas las alternativas presentes según el conocimiento del sistema en ese instante de tiempo. Se trata de la opción más flexible puesto que puede emplearse para cualquier fenómeno típico. No obstante, ambos mecanismos de selección no son excluyentes, por lo que la gran mayoría de las arquitecturas cognitivas existentes presentan ambos (Kotseruba & Tsotsos, 2020).

2.1.1.4 Razonamiento

El razonamiento es vital para resolver los problemas. Aunque en su origen se trataba únicamente de una cuestión filosófica, en la actualidad, se ha convertido en uno de los temas de mayor interés en la investigación de la ciencia cognitiva (Kotseruba & Tsotsos, 2020). Esta capacidad permite desarrollar conclusiones a raíz de “ideas” que ya posee el sistema (Langley et al., 2009).

Para esto, es necesario establecer y representar las relaciones existentes entre las creencias de la arquitectura (Langley et al., 2009). Esta tarea puede realizarse de una forma clásica, utilizando la inferencia lógica a través de la deducción, inducción y abducción, o de una forma más novedosa, utilizando la heurística, la moral o la analogía entre otras (Kotseruba & Tsotsos, 2020).

Entre los mayores retos que supone el desarrollo de esta capacidad, se encuentra la necesidad de tomar las decisiones correctas pese al desconocimiento, al igual que les sucede a los seres humanos. Además, se añade la problemática de la limitación de recursos como, por ejemplo, la capacidad de procesamiento de información (Kotseruba & Tsotsos, 2020).

Es por ello, que se está haciendo especial hincapié a la hora de diseñar las arquitecturas, en sobrepasar las problemáticas descritas.

2.1.1.5 Memoria

La memoria es una capacidad esencial que suele clasificarse dentro de los mecanismos de metagestión y consiste en la capacidad de codificar los resultados y almacenarlos para así poder tener acceso a ellos más tarde. No obstante, los sistemas no son capaces de recordar situaciones o acciones directamente, sino que deben guardan las estructuras que las representan (Langley et al., 2009).

La gran mayoría de las arquitecturas cognitivas que se han desarrollado poseen un sistema de memoria para así poder almacenar los diferentes resultados de los cálculos, sentando así una base para poder desarrollar la capacidad del aprendizaje y de la adaptación al entorno. (Kotseruba & Tsotsos, 2020)

La memoria suele describirse, en el campo de la arquitectura cognitiva, según su duración (corto y largo plazo) y dentro de esta, según su tipo.

La memoria a corto plazo se divide en sensorial o perceptiva y de trabajo, siendo la sensorial como un pequeño buffer a muy corto plazo que guarda muchas percepciones recientes. Por otro lado, la memoria de trabajo es un almacén temporal para esas percepciones y también para otra información sobre la tarea actual, por lo que suele asociarse con la capacidad de atención ya descrita con anterioridad.

La memoria a largo plazo, por el contrario, sí que almacena conocimiento objetivo, información sobre qué realizar si ocurren ciertas situaciones, así como recuerdos de experiencias personales del sistema. Esta se puede dividir a su vez en semántica (hechos objetivos), procedural (rutinas de movimiento y comportamiento) y episódica (Kotseruba & Tsotsos, 2020).

El resultado obtenido suele conocerse como *memoria episódica,* debido a la similitud que presenta con esta (Langley et al., 2009). En un ser humano, la memoria episódica es la que permite evocar, de una forma clara y nítida, episodios pasados de la vida. (*Memoria episódica y envejecimiento cognitivo*, s. f.)

2.1.1.6 Aprendizaje

El aprendizaje es la capacidad que posee una arquitectura cognitiva de mejorarse a sí misma y sus funcionalidades a lo largo del tiempo. Al igual que en el mundo real, el aprendizaje se basa en las experiencias. Por ejemplo, se puede aprender de los hechos que se han observado que han tenido lugar como parte de ciertas situaciones o como resultados de las acciones.

Los métodos aplicados para el aprendizaje dependen de las múltiples variables que forman parte del desarrollo de una arquitectura cognitiva, como son el paradigma de diseño, el entorno, las estructuras de datos o los algoritmos empleados. No obstante, esta capacidad puede ser dividida por métodos entre los que destacan los siguientes:

- El aprendizaje perceptivo se implementa en aquellas arquitecturas que modifican, de forma activa, la manera en la que la información sensorial es manejada o cómo se aprenden los patrones. El objetivo principal de este método incluye conocer el

entorno a través de mapas espaciales, descubrir características visuales o encontrar asociaciones entre elementos.

- El aprendizaje declarativo es un conjunto de hechos sobre el mundo y las relaciones que estos mantienen entre sí. Suelen implementarse mediante métodos de fragmentación, cuando un nuevo fragmento de información es almacenado en la memoria como, por ejemplo, cuando se cumple un objetivo. Estos fragmentos de información también pueden ser introducidos en el sistema directamente por los desarrolladores, así como ser extraídos del entrenamiento previos realizado.
- El aprendizaje procedural hace referencia al aprendizaje de habilidades a través de la repetición, de manera que, con el tiempo, esa habilidad se convierta en automática. La forma más sencilla de hacerlo es acumular los ejemplos obtenidos tras resolver problemas de forma exitosa para así poder utilizarlos a posteriori. Un ejemplo de ello podría ser la reutilización de un camino para la navegación entre dos puntos ya conocidos.
- El aprendizaje asociativo se caracteriza por estar conformado por procesos de toma de decisiones acompañados de recompensas o castigos. Conocido como el aprendizaje reforzado, es comúnmente utilizado por la gran evidencia existente de que resulta fundamental a la hora de desarrollar la toma de decisiones y la adquisición de habilidades motoras. Además, se trata de un método simple y eficiente, presentando la principal ventaja de que puede ser aplicado en cualquier tipo de arquitectura. (Kotseruba & Tsotsos, 2020)

2.1.2 Ejemplos

A continuación, se van a describir de forma general las dos arquitecturas cognitivas más relevantes que se encuentran en desarrollo en la actualidad.

2.1.2.1 ACT-R

ACT-R es una arquitectura cognitiva clasificada como híbrida (Ritter et al., 2019) que se comenzó a desarrollar a finales de los años 70 (Langley et al., 2009). El origen de ACT-R se encuentra en un simulador de memoria llamado HAM. Tras varias revisiones en las que se fueron implementando nuevos módulos y sistemas al software, se ha alcanzado a versión ACT-R 7. (Ritter et al., 2019)

Está organizada en módulos, de manera que cada uno de ellos procesa un tipo de información (Langley et al., 2009) con el objetivo de predecir y explicar el comportamiento humano, así como la interacción con el entorno (Ritter et al., 2019).

Los módulos principales están dedicados a la identificación objetos en el campo visual (módulo visual), a realizar el seguimiento de metas e intenciones actuales (módulo de objetivos), a la recuperación de información de la memoria (módulo declarativo) y al control de las manos (módulo manual).

Por otro lado, posee un sistema de producción central cuya misión es realizar la coordinación de la comunicación y del rendimiento de los módulos mencionados.

ACT-R tiene ciertos búferes específicos, que almacenan contenido para que sea visible para los módulos. Por ejemplo, dispone de un búfer manual para el control y monitoreo de las manos y un búfer visual para ubicar e identificar los objetos.

El funcionamiento general de este sistema empieza por los búferes, que disponen las representaciones del mundo externo y el interno. Mediante reglas de producción, se reconocen patrones en la información de los búferes y se selecciona y aplica una acción correspondiente con lo almacenado en la memoria de trabajo. Tras ello, los búferes vuelven a actualizarse permitiendo otra vez el comienzo del ciclo, con una duración aproximada de 50 ms. (Ritter et al., 2019)

El diagrama de interrelación de los componentes puede observarse en la Figura 2.

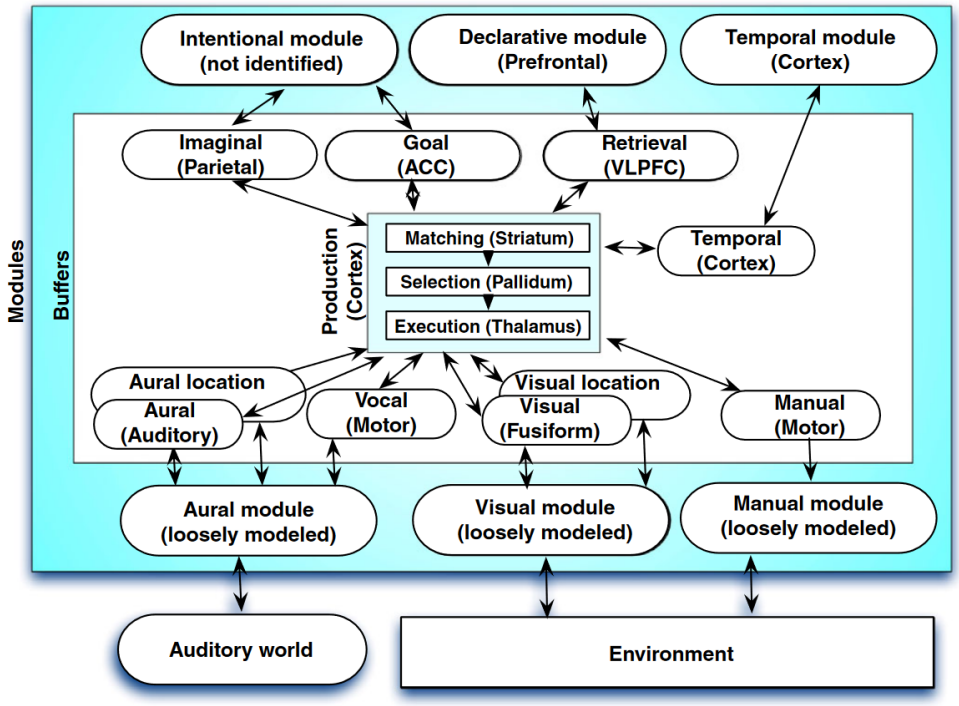

Figura 2. Diagrama esquemático de los componentes de la arquitectura ACT-R. (Ritter et al., 2019)

2.1.2.2 Soar

Soar es otra arquitectura cognitiva de código abierto (Laird, 2022) que se encuentra dentro del grupo de las híbridas (Kotseruba & Tsotsos, 2020). Su desarrollo comenzó a principios de los años 80 y en todo este tiempo, se han ido incorporando nuevos módulos y capacidades.

Diseñada para la creación de agentes de inteligencia artificial con características y capacidades cognitivas similares a las de los humanos, consiste en módulos interactivos e independientes para cada tarea (Laird, 2022), que a su vez se expresan como intentos para lograr objetivos (Langley et al., 2009).

A nivel general, Soar dispone de memoria a corto plazo (de trabajo) y de largo plazo, así como de módulos de procesamiento, como sistemas de decisión, de percepción

visual o de control motor, y de aprendizaje, como aprendizaje semántico, episódico y reforzado. El diagrama de componentes interrelacionados puede verse en la Figura 3.

A grandes rasgos, el funcionamiento de esta arquitectura cognitiva comienza con la entrada de información del entorno a través de los módulos de percepción que, tras convertirse en representación simbólica, se almacena en la memoria de trabajo. Tras ello, se intentan reconocer patrones y se selecciona la regla de producción más adecuada. Tras aplicarla, se evalúa el resultado obtenido pudiendo dar lugar a crear nuevos objetivos, así como al aprendizaje de nuevas reglas (Laird, 2022).

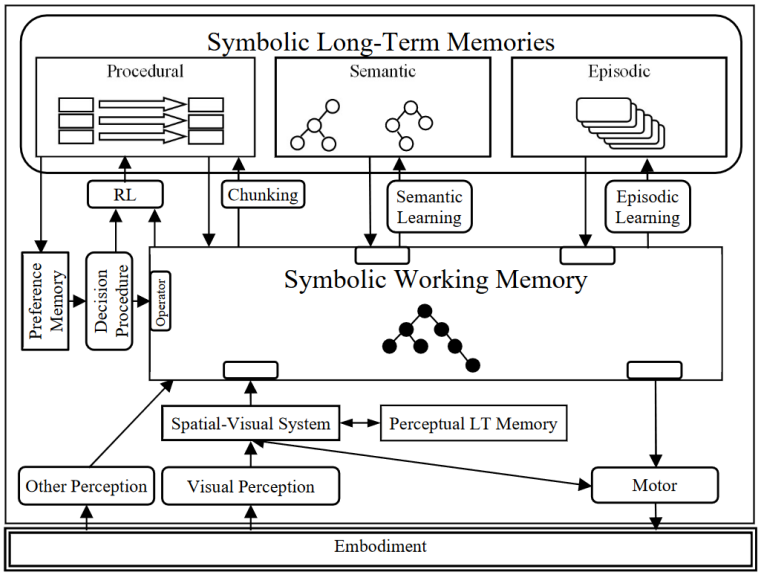

Figura 3. Diagrama esquemático de los componentes de la arquitectura Soar. (Laird, 2022)

2.2 Mapas en robótica móvil.

La representación del entorno en la robótica móvil es una parte esencial dentro de la navegación autónoma. Dentro de las tareas que se deben realizar durante el proceso de navegación se encuentran el mapeo y la localización entre otras. Ambas conforman el término conocido como SLAM (Simultaneous Localization and Mapping – localización y mapeo simultáneos) (Sebastian Thrun, 2005). Este hace referencia a la capacidad que tiene un robot de poder crear un mapa del entorno en el que se encuentra al mismo tiempo que es capaz de determinar su posición.

El algoritmo de representación de mapas más habitual es mediante *grid,* donde el mapa está discretizado y formado por celdas de una resolución determinada. Según el valor binario asignado a cada celda se establece su ocupación (0 → libre, 1 → ocupada). La determinación de la ocupación de cada celda se puede realizar empleando visión artificial (cámara) o, por ejemplo, con un láser 2D haciendo barrido.

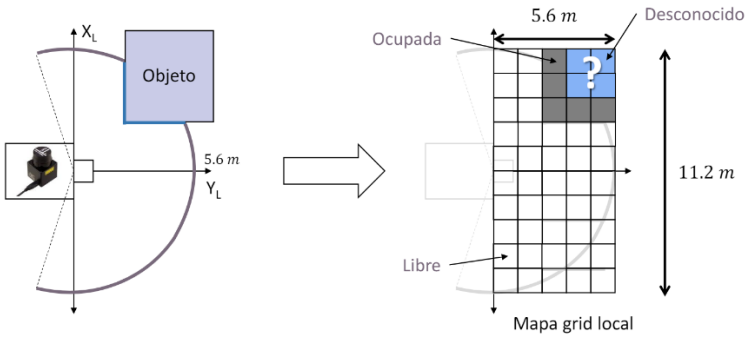

Figura 4. Representación mapa tipo grid

Son muchos los algoritmos de mapeo basados en SLAM que se han desarrollado a lo largo de la evolución de la robótica móvil. Por su compatibilidad con ROS (Robot Operating System) y su fácil integración, destaca GMapping.

2.2.1 GMapping

(Ratul et al., 2021) GMapping es un algoritmo que basa el mapeo en los datos provenientes de sensores como un LiDAR y los relaciona con modelos de movimiento para poder realizar ambas tareas (localización y mapeo) de forma simultánea.

El flujo del funcionamiento de GMapping puede describirse a través de varias etapas:

1. Adquisición de datos del sensor: se obtiene información sobre la estructura del entorno en forma de nubes de puntos con los datos que se adquieren del LiDAR.
2. Inicialización del mapa y posición del robot: GMapping inicia con un mapa vacío y una estimación inicial de la posición del robot en el entorno. A medida que el robot se mueve y recibe datos del sensor, el algoritmo actualiza el mapa y ajusta la posición estimada del robot.
3. Asociación de datos y actualización del mapa: GMapping utiliza técnicas de asociación de datos para relacionar las lecturas del sensor con las características del entorno y actualizar el mapa en función de esta información. Esto permite al algoritmo construir un mapa detallado y preciso a medida que el robot se desplaza.
4. Estimación de la posición del robot: A través de la integración de datos del sensor y modelos de movimiento, GMapping estima continuamente la posición y orientación del robot en el entorno. Esta estimación se actualiza en tiempo real a medida que se recopilan más datos del sensor.
5. Resampling: Durante el proceso de resampling, se ajusta la distribución de partículas utilizadas para estimar la posición del robot. Las partículas con baja importancia se reemplazan por aquellas con mayor peso, lo que ayuda a mejorar la precisión de la estimación de la posición del robot.

GMapping es un algoritmo realmente eficiente y preciso para llevar a cabo la tarea de SLAM de robots móviles en entornos dinámicos.

2.3 Sistemas de reconocimiento facial.

El reconocimiento facial es la técnica biométrica más empleada para la autentificación de personas en todos los ámbitos necesarios, por lo que es una línea de estudio que ha evolucionado durante décadas, desde los años 90 (Wang & Deng, 2021). Este avance se ha debido principalmente al aumento y la mejora de las tecnologías relevantes para su desarrollo, especialmente de la inteligencia artificial (Adjabi et al., 2020).

Aun siendo menos preciso que otros datos biométricos como el iris o las huellas dactilares, proporciona la ventaja de no ser una técnica intrusiva, por lo que puede hacerse sin contacto físico y es aceptada por la mayoría de las personas (Adjabi et al., 2020).

Existen tres módulos indispensables para desarrollar un sistema de reconocimiento facial. Primeramente, es necesario emplear un sistema de detección facial para poder extraer los rostros que aparecen en fotos o vídeos. Tras ello, se deben extraer las características de estos rostros y normalizarlas para, finalmente, clasificarlas a través de la verificación o identificación (Wang & Deng, 2021).

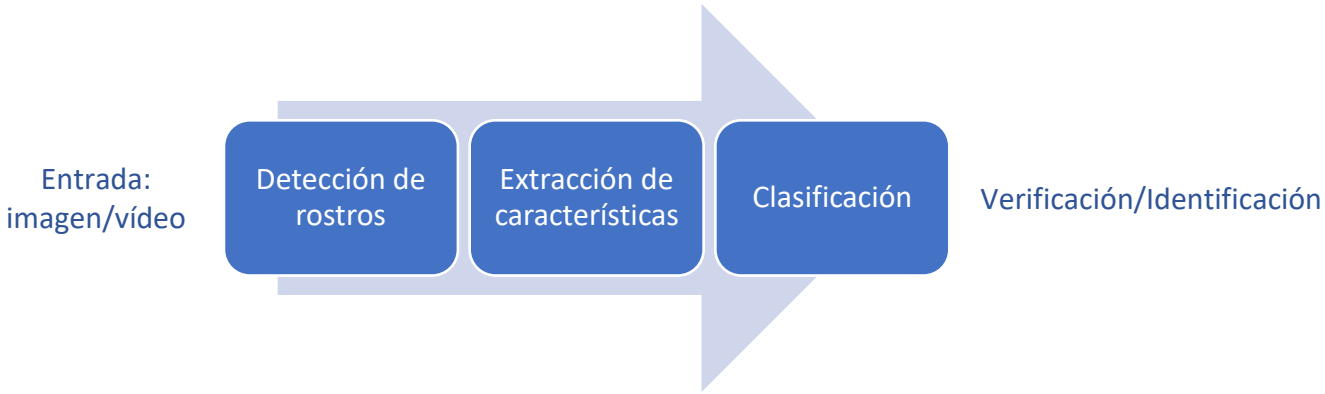

Figura 5. Diagrama módulos principales de un sistema de reconocimiento facial

2.3.1 Detección facial

La detección facial debe hacerle frente al desafío de detectar los rostros en las imágenes a pesar de las variaciones en la iluminación, de los cambios de postura y la complejidad del entorno. Por ello, se han desarrollado una amplia variedad de algoritmos y enfoques, que permiten aplicarla en los diferentes contextos y necesidades (Hasan et al., 2021).

Los enfoques para la detección de rostros humanos se pueden clasificar en dos categorías principales: basados en características y basados en imágenes. Los enfoques basados en características se centran en identificar elementos específicos en una imagen, como bordes, esquinas y estructuras bien localizadas en dos dimensiones. El proceso de detección se basa en la extracción y análisis de características visuales distintivas para identificar rostros en una imagen. Algunos ejemplos de métodos

incluyen modelos activos de forma, serpientes, coincidencia de plantillas deformables y análisis de características como color, bordes y movimiento. Estos métodos son eficaces para detectar características específicas en entornos controlados, aunque son sensibles a variaciones en la iluminación, oclusiones y cambios en la escala y orientación (Hasan et al., 2021).

Por otro lado, los enfoques basados en imágenes se basan en el análisis global de la imagen mediante el escaneo de ventanas para identificar rostros. Estos enfoques procesan la imagen en su totalidad para detectar patrones característicos de rostros. Algunos ejemplos de métodos incluyen el uso de redes neuronales y métodos estadísticos para el reconocimiento facial. Estos métodos son eficaces en entornos complejos con variaciones en la iluminación, oclusiones y fondos complejos. Sin embargo, requieren mayor poder computacional y pueden ser menos precisos en la detección de características específicas en comparación con los enfoques basados en características (Hasan et al., 2021).

2.3.2 Reconocimiento facial

Los métodos de reconocimiento facial han evolucionado significativamente en los últimos años, con un enfoque creciente en el uso de tecnologías avanzadas como el aprendizaje profundo (en Inglés, Deep Learning), para mejorar la precisión y la eficiencia de los sistemas de FR. Pueden clasificarse principalmente en métodos 2D y 3D (Adjabi et al., 2020).

Los métodos 2D han alcanzado altas tasas de reconocimiento en entornos controlados donde los parámetros de adquisición están regulados, pero su rendimiento puede verse afectado con las condiciones variables como cambios de iluminación o expresiones faciales. Ejemplos de métodos 2D empleados en el reconocimiento facial son:

- Método Holístico: se describe toda la cara como un conjunto de características globales. Se utilizan técnicas como el análisis de componentes principales y modelado de deformaciones para identificar similitudes entre rostros.
- Método Local: se centra en examinar las características geométricas específicas de la cara, como los ojos, la nariz y la boca. Se busca la correspondencia de estas características para la identificación facial.
- Método Híbrido: Combina elementos de los enfoques holístico y local para mejorar la precisión del reconocimiento facial. Se pueden utilizar tanto características globales como locales para lograr una identificación más precisa.

Por otro lado, los métodos 3D se presentan como una solución alternativa para superar las limitaciones de los enfoques 2D, ya que son invariantes a condiciones como la pose y la iluminación, lo que mejora la eficiencia de los sistemas de reconocimiento. Se encuentran, por ejemplo, los siguientes métodos:

- Imágenes de Profundidad: Se basa en el uso de sensores de profundidad para capturar la información tridimensional de la cara. Estas imágenes permiten obtener detalles precisos de la forma y la estructura facial, lo que facilita la identificación y el reconocimiento.
- Modelado de Nubes de Puntos: se representa la cara como una nube de puntos en un espacio tridimensional. Se utilizan algoritmos de procesamiento de nubes de puntos para extraer características distintivas y realizar comparaciones entre diferentes rostros.
- Mallas 3D: se representar la cara como una malla tridimensional de vértices y aristas. Esta representación permite capturar con precisión la forma y la textura facial, lo que facilita el reconocimiento en entornos 3D.

2.3.3 Algoritmos

A continuación, se van a describir dos algoritmos ampliamente utilizados que representan enfoques diferentes en el reconocimiento facial (Adjabi et al., 2020).

2.3.3.1 Eigenfaces

Eigenfaces es un algoritmo clásico de reconocimiento facial basado en el análisis de componentes principales (PCA) y en la idea de que las caras humanas comparten ciertas características comunes. Fue propuesto por Alex Pentland y Matthew Turk del MIT en 1991.

Utiliza una técnica de reducción de dimensionalidad para representar las caras como combinaciones lineales de vectores propios (eigenfaces) que capturan las variaciones más significativas en un conjunto de imágenes faciales. El algoritmo calcula los vectores a partir de un conjunto de imágenes de entrenamiento y luego proyecta nuevas imágenes faciales en este espacio de eigenfaces para realizar comparaciones y reconocimiento. La distancia entre la proyección de la nueva cara y las caras de referencia en el espacio de eigenfaces se emplea para determinar la identidad.

Es relativamente simple y computacionalmente eficiente, aunque solo es realmente efectivo en entornos controlados donde las condiciones de iluminación y pose son estables.

2.3.3.2 Redes Neuronales Convolucionales (CNN)

Las redes neuronales convolucionales son algoritmos de aprendizaje profundo que han demostrado un rendimiento excepcional en tareas de visión por ordenador, incluido el reconocimiento facial. Estas redes están compuestas por capas convolucionales que aplican filtros para extraer características locales de las imágenes y capas de agrupación que reducen la dimensionalidad de las características extraídas. Las capas totalmente conectadas al final de la red utilizan las características aprendidas para realizar la clasificación final.

Las CNN aprenden automáticamente las características relevantes de las imágenes faciales a partir de datos de entrenamiento, lo que les permite realizar tareas de reconocimiento facial con alta precisión y robustez. Pueden capturar así características complejas y no lineales de las imágenes faciales, lo que las hace adecuadas para entornos variables y desafiantes. Son altamente adaptables y pueden mejorar su rendimiento con grandes conjuntos de datos.

La principal desventaja que presentan es que requieren grandes cantidades de datos y recursos computacionales para el entrenamiento. Además, la interpretación de sus decisiones puede ser más compleja en comparación con métodos tradicionales como Eigenfaces.

2.4 Justificación del trabajo.

Los avances recientes en el ámbito de la Inteligencia Artificial y la Robótica han sido significativos. La evolución de los modelos de aprendizaje automático ha permitido una mayor precisión y eficiencia en la toma de decisiones autónomas por parte de los robots. Estos modelos son esenciales para el desarrollo de sistemas capaces de adaptarse a entornos dinámicos y complejos. La mejora en las tecnologías de visión artificial ha incrementado la capacidad de los robots para reconocer y reaccionar ante diferentes objetos y personas en su entorno, lo cual es crucial para la navegación y la interacción en tiempo real. Asimismo, el incremento en la velocidad de procesamiento de datos permite que las plataformas robóticas realicen tareas complejas de manera más rápida y eficiente, lo cual es fundamental para aplicaciones en tiempo real.

A pesar de estos avances, existen varios desafíos en los sistemas de reconocimiento actuales. Los sistemas de reconocimiento empleados con frecuencia son familiares para los usuarios, lo cual facilita su aceptación y uso. Sin embargo, estos sistemas a menudo carecen de personalización y adaptabilidad a diferentes contextos. Además, la dependencia en empresas especializadas para la implementación de estos sistemas puede ser costosa y no siempre asegura una integración perfecta con otros sistemas existentes. Un problema crítico es la falta de anonimización de datos en muchos sistemas de reconocimiento actuales, lo que plantea serias preocupaciones sobre la privacidad. Los procesos de entrenamiento de estos sistemas también pueden ser lentos y requerir grandes cantidades de datos y tiempo, lo que retrasa su implementación efectiva.

Por otro lado, las residencias de atención a personas mayores enfrentan un problema crítico de sobresaturación de sus trabajadores, lo que impacta negativamente en la calidad de los servicios prestados. La implementación de tecnologías robóticas puede proporcionar un apoyo significativo a los trabajadores, liberándolos de tareas rutinarias y permitiéndoles concentrarse en aspectos más críticos del cuidado. La introducción de nuevas funcionalidades a través de plataformas robóticas mejoraría la eficiencia operativa y la satisfacción de los residentes, creando un entorno más seguro y cómodo.

Este trabajo propone contribuir de manera significativa en los siguientes aspectos. En primer lugar, se desarrollará un sistema de reconocimiento propio que sea rápido y garantice la anonimización de los datos personales. Este sistema estará diseñado específicamente para integrarse con la arquitectura cognitiva propuesta y adaptarse a las necesidades específicas del entorno asistencial. Esta arquitectura cognitiva, será capaz de memorizar y olvidar a las personas reconocidas, así como de ubicarlas en un mapa semántico. Todo esto, daría lugar a la posibilidad de implementar funcionalidades avanzadas como la vigilancia y la búsqueda de personas dentro de la plataforma robótica. Estas características no solo mejorarían la seguridad, sino que también proporcionarán herramientas valiosas para la gestión y el monitoreo de los residentes.

En resumen, la justificación de este trabajo se basa en la necesidad de mejorar y optimizar la plataforma robótica asistencial existente, desarrollada en un contexto de investigación de la Universidad Politécnica de Cartagena (UPCT), mediante la incorporación de una arquitectura cognitiva que integre de manera eficiente a los usuarios humanos, abordando los desafíos actuales y aportando una base para desarrollar soluciones innovadoras que beneficien tanto a los residentes como al personal de las residencias.

3 DISEÑO DEL SISTEMA.

En el presente capítulo se describe la arquitectura software del sistema. Se puntualizan las distintas fases de diseño y desarrollo seguidas, previo a las fases de integración y prueba. Además, se realiza una detallada descripción de todos los componentes implementados, así como de los métodos y de las posibles alternativas que han sido estudiadas para ejecutar el desarrollo del sistema planteado.

3.1 Análisis.

El robot que se ha empleado para el desarrollo del sistema y su integración es el prototipo perteneciente a la plataforma asistencial JUNO, desarrollada por la Universidad Politécnica de Cartagena a través de diferentes proyectos de investigación bajo la dirección de los investigadores responsables, la Dra. Nieves Pavón Pulido y el Dr. Juan Antonio López Riquelme. Actualmente, esta plataforma se encuentra en desarrollo dentro del subproyecto nacional JUNO+, financiado en la convocatoria de «Proyectos Estratégicos Orientados a la Transición Ecológica y a la Transición Digital» 2021, del Ministerio de Innovación, Ciencia y Universidades.

Figura 6. Robot JUNO en una demostración (Opinión, 2024)

En este trabajo, se persigue obtener un sistema para implementar en el robot que sea capaz de detectar y reconocer los rostros de las personas en tiempo real mientras se encuentra, ya sea en movimiento realizando otras tareas, o fijo en un punto. Una vez reconozca a personas, debe ser capaz de ubicarlas en un mapa semántico para, posteriormente, poder predecir donde es posible que se encuentre una persona en un determinado momento.

En este capítulo van a definirse los pasos que se han seguido para lograr el objetivo, junto a todo el software y hardware que ha sido necesario para el mismo fin.

3.1.1 Hardware

Los principales componentes hardware necesarios para el desarrollo de este TFM son los siguientes:

- MSI Cubi 5: mini PC con el que va equipado el robot JUNO. Dispone de un procesador i7, 8GB de RAM y un SSD de 120GB. Lleva instalado un sistema operativo Linux con Ubuntu 20.04 así como la versión de ROS correspondiente, ROS Noetic
- Cámara Astra Pro: cámara RGB-D equipada en el robot de la empresa Orbbec. Dispone de un paquete (*ros_astra_camera)* para ser utilizada cómodamente en ROS.

Figura 7. MSI Cubi 5 (*Cubi 5 12M | Best Mini Desktop PC 0.7L | Be Your Window To The World*, s. f.)

Figura 8. Cámara Astra Pro («Astra Series», s. f.)

3.2 Sistema de reconocimiento facial.

Este sistema es, a nivel general, el encargado de detectar los rostros de las personas que capta la cámara RGB-D y de reconocerlas. Se ha dividido en dos subsistemas para su desarrollo, los cuales se han llevado a cabo de dos formas distintas para poder elegir la opción óptima.

3.2.1 Subsistema de detección facial

El subsistema de detección facial, responsable de segmentar rostros de personas de la imagen obtenida a través de la cámara del robot, constituye la primera etapa del sistema de reconocimiento facial.

A continuación, se van a describir las dos formas en las que se ha abarcado el desarrollo de este subsistema.

3.2.1.1 Detección con MTCNN

Primeramente, se probó la solución para la detección facial propuesta por un proyecto encontrado en GitHub (enlace al proyecto).

En esta guía se propone el uso de MTCNN (Multi-task Cascaded Convolutional Networks). Se trata de un algoritmo ampliamente empleado en la detección de rostros gracias a su precisión y su robustez, lo que se debe a su estructura de redes. MTCNN trabaja en tres etapas usando redes convolucionales en cascada: P-Net, R-Net y O-Net. Primero, la imagen es redimensionada a diferentes escalas para generar una pirámide de imágenes. P-Net realiza la detección inicial de caras, generando cuadros candidatos y ajustándolos mediante un algoritmo de supresión no máxima (NMS). R-Net refina estos cuadros, descartando los candidatos de baja puntuación y produciendo vectores de características más precisos. Finalmente, O-Net realiza una selección final de los mejores cuadros candidatos y detecta cinco puntos clave faciales con mayor precisión gracias a su estructura más profunda (*Research on Face Detection Technology Based on MTCNN*, s. f.).

La detección con MTCNN será complementada inicialmente con un subsistema de reconocimiento facial basado en FaceNet, tal y como sugiere el ejemplo seguido para este apartado.

Para terminar, cabe mencionar que las modificaciones realizadas al *script* del proyecto mencionado se basan únicamente, en este punto, en "rosificar"[1].

3.2.1.2 Detección con Mediapipe

Una alternativa al uso de MTCNN como base del subsistema de detección es Mediapipe.

Mediapipe es un conjunto de bibliotecas y herramientas desarrollado por Google que permite la aplicación de técnicas de inteligencia artificial y aprendizaje automático a aplicaciones propias. Se basa en un amplio grupo de soluciones personalizables y disponibles en diferentes lenguajes de programación (*Guía de soluciones de MediaPipe | Google AI Edge*, s. f.).

Para esta tarea se ha empleado Mediapipe Face Detector. Esta solución permite detectar todos los rostros que se encuentran en una imagen o en un vídeo gracias a que usa un modelo de aprendizaje automático. El resultado del detector de rostros es un marco que delimita los rostros (*bounding box*) y las coordenadas de 6 puntos de referencia faciales para cada uno de los rostros (*Guía de detección de rostro | Google AI Edge*, s. f.).

Esta aplicación se ha desarrollado en Python con el modelo *BlazeFace* (corto alcance), optimizado para imágenes a corto alcance. No obstante, lo más interesante sería poder emplear el modelo *BlaceFace* disperso, pues es más ligero y está optimizado para imágenes de rango completo, como las que se toman con la cámara RGB-D. Este último

[1] "rosificar": modificar un código ya existente para que funcione como un nodo dentro del entorno de ROS

modelo mencionado estará disponible dentro de poco tiempo según los desarrolladores de Mediapipe.

Para configurar el entorno de desarrollo de manera que pueda emplearse esta herramienta para la aplicación deseada se debe seguir la guía para entrada de tipo imagen (*Guía de detección de rostro para Python | Google AI Edge*, s. f.). Cabe destacar que se implementa la solución para entrada tipo imagen en vez de video ya que ROS, entorno en el que se va a integrar la aplicación, trabaja con la entrada de cámaras en modo vídeo por *frames*, por lo que realmente se deben procesar secuencias de imágenes y no vídeo en sí.

3.2.2 Subsistema de reconocimiento facial

El subsistema del reconocimiento facial parte de los resultados obtenidos del subsistema de detección para extraer los rostros e identificarlos. Para poder llevar a cabo el desarrollo de este subsistema de forma más eficiente, se ha desarrollado, además de dos sistemas con herramientas diferentes, un script que permite la fácil creación de un *dataset* con los sujetos deseados para así poder entrenar los modelos.

3.2.2.1 Recopilador de rostros

Se trata de un script que puede ejecutarse fuera del entorno de ROS, programado en Python, y que está diseñado para tomar una secuencia de fotos a una persona. Antes del comienzo de la secuencia solicita su nombre, de manera que se crea un sistema de directorios ordenados con las diferentes fotos tomadas a cada persona, que puede ser usado con posterioridad para las tareas necesarias.

La idea de este este programa es facilitar la creación del *dataset* para el entrenamiento de modelos.

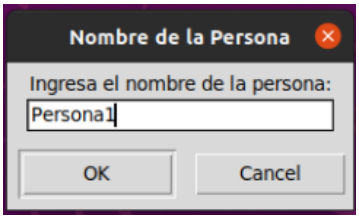

Figura 9. Pantalla inicial recopilador de rostros

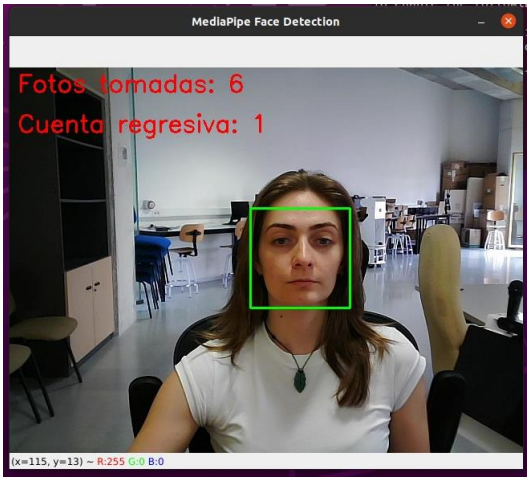

Figura 10. Ejemplo uso recopilador de rostros

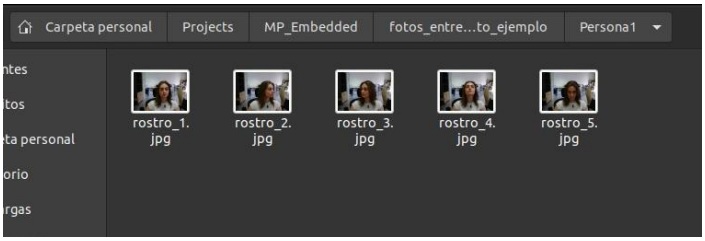

Figura 11. Ejemplo directorio /Persona1

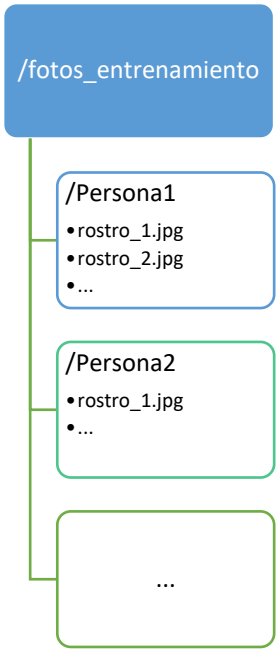

Figura 12. Estructura dataset

3.2.2.2 Reconocimiento con FaceNet

La primera herramienta que se ha empleado es FaceNet, siguiendo la guía descrita en el [proyecto de GitHub] ya mencionado.

FaceNet es un sistema avanzado de reconocimiento facial que fue desarrollado y presentado por Google en 2015. FaceNet aprende directamente a asignar imágenes faciales a un espacio euclidiano compacto donde las distancias entre los puntos representan la similitud entre los rostros. Esto significa que rostros son representados por vectores (*embeddings)* de dimensión 1x128 y que, si dos rostros son similares, estos estarán más cerca en el espacio euclidiano, lo que facilita las tareas de reconocimiento y verificación facial (Schroff et al., 2015).

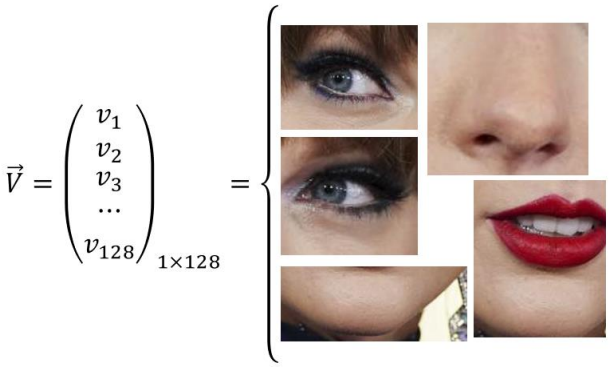

Figura 13. Ejemplo representativo *embedding*

Se basa en una red neuronal convolucional profunda que se entrena para optimizar directamente este mapeo de imágenes faciales al espacio euclidiano. Para este

entrenamiento se utiliza la técnica del *"Triplet Loss"*, que consiste en comparar dos embeddings que son similares y uno que es diferente, y el objetivo es separar las parejas positivas de las negativas por un margen de distancia (Schroff et al., 2015).

Para poder ponerlo en funcionamiento, se ha "*rosificado*" el código disponible. Se ha realizado una modificación de manera que publique por un topic de ROS el nombre de la persona que ha detectado. Además, se debe reentrenar el modelo con el *dataset* deseado gracias al *script* dedicado para ello que proporciona el proyecto.

3.2.2.3 Reconocimiento con Mediapipe

La otra herramienta empleada es Mediapipe, esta vez con la solución Image Embedding. Esta tarea permite la creación de una representación numérica de una imagen en forma de lista de vectores. Esta solución está inspirada en el funcionamiento de FaceNet.

3.2.2.3.1 Extractor de embeddings

Previo al desarrollo del modelo de reconocimiento, es necesario obtener los *embeddings* asociados a los rostros para poder entrenarlo. Se ha creado un script que utiliza las imágenes obtenidas con el recopilador de rostros. De nuevo, empleamos la herramienta de Mediapipe Face Detection para obtener la ROI de las fotos de entrenamiento.

Una vez se obtiene el recorte correspondiente a la ROI, se calcula el vector característico de esa imagen con el Image Embedding. Esto se realiza para cada una de las imágenes que se encuentran en el directorio de las fotos para entrenamiento; cada persona que se desea reconocer dispone de una carpeta con su nombre y sus fotos contenidas dentro.

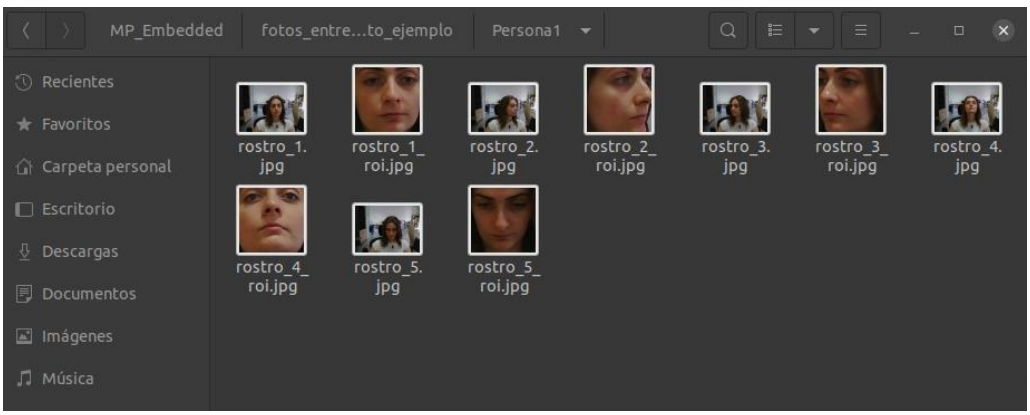

Figura 14. Dataset tras extraer embeddings

Los nombres de las personas a quienes pertenecen los embeddings correspondientes se añaden a unas listas que, al final del código, se exportan a un archivo *.npz* como matrices NumPy.

Para poder visualizar los embeddings extraídos, se ha escrito un script de Python para mostrarlos por consola.

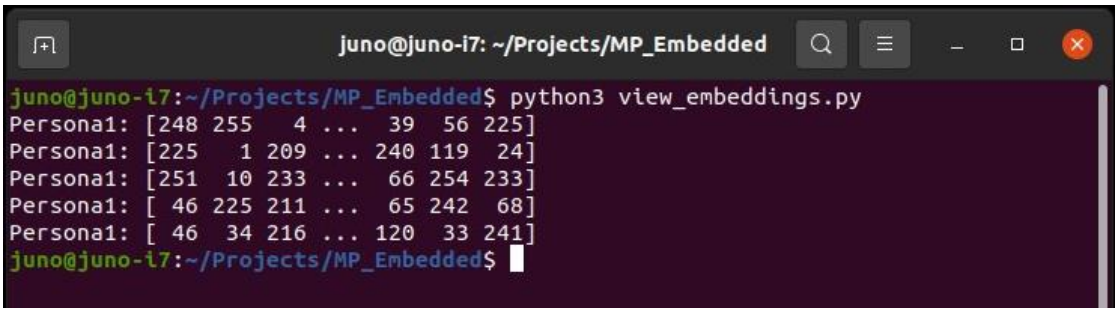

Figura 15. Visualización embeddings extraídos Persona1

3.2.2.3.2 Data augmentation

Tras haber realizado ciertas pruebas con el entrenamiento del modelo, se observa una baja precisión, lo que se asocia a la poca cantidad de datos de entrenamiento. Por ello se desarrolla un código para ampliar el *dataset*.

Para alcanzar este objetivo se emplea una clase de la biblioteca *Keras*, *ImageDataGenerator*, que permite generar nuevas imágenes a partir de las que ya existen con la aplicación de rotaciones, desplazamientos, zoom, etc.

Previo a ejecutar el *script* para ampliar el dataset, se ejecuta otro *script* desarrollado para eliminar las imágenes no ROI para evitar que se amplíen estas también.

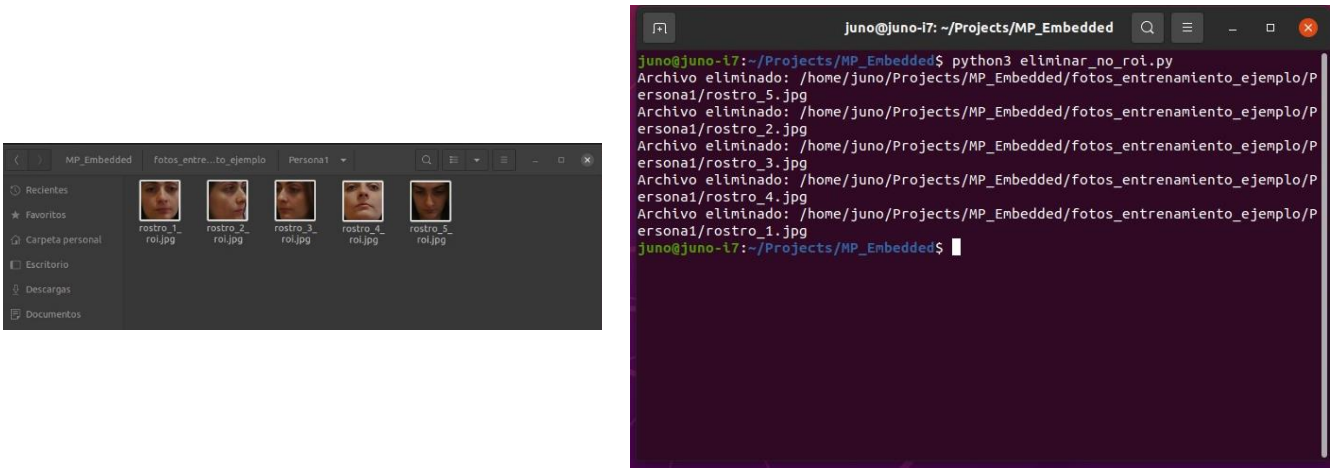

Figura 16. Eliminación de imágenes no ROI

De esta forma, el *dataset* que inicialmente disponía de 60 fotos ahora ha sido mejorado y dispone de casi 300, lo que puede asegurar un mejor rendimiento en el entrenamiento del modelo.

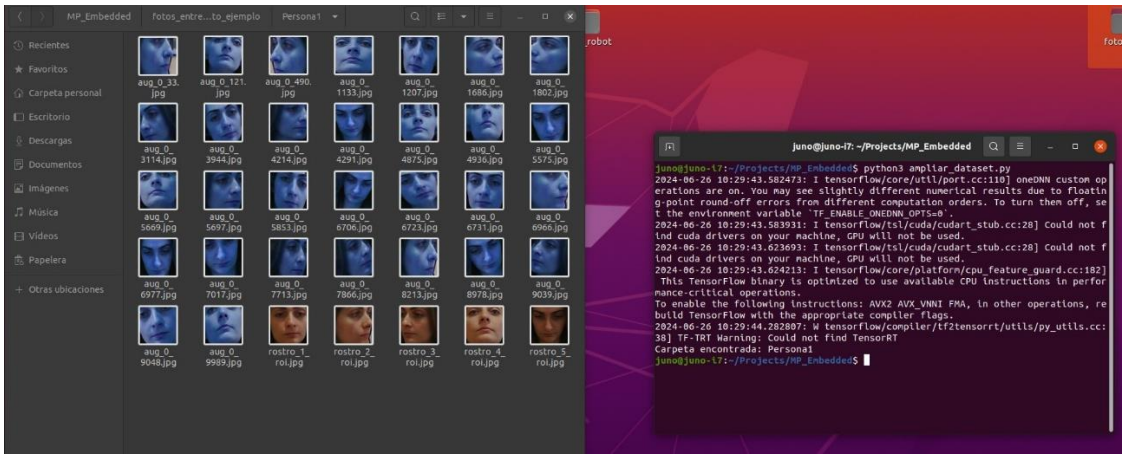

Figura 17. Dataset ampliado

Tras haber ampliado el *dataset* es necesario volver a obtener los *embeddings* para entrenar el modelo con los procedentes de las nuevas imágenes generadas.

3.2.2.3.3 Modelo KNN

Una vez se tiene el archivo *.npz* con los *embeddings* del *dataset* deseado se procede a generar el modelo.

El modelo seleccionado ha sido KNN (K-Nearest Neighbors) por su rapidez y sencillez de entrenamiento, así como su buen rendimiento.

Es un algoritmo cuya idea general se basa en que los objetos que son similares están más cerca en el espacio de características. Para realizar una predicción, el algoritmo encuentra un número predefinido de k vecinos más cercanos y obtiene una predicción que se basa en los valores de estos vecinos.

Con una k = 5, la precisión máxima obtenida ha sido próxima al 100% para el dataset ampliado.

Por último, cabe destacar que también se han realizado pruebas sobre un modelo SVM (Support Vector Machine) pero no se ha obtenido tanta precisión como para el KNN, estando esta siempre entorno al 90%.

```
juno@juno-i7:~/Projects/MP_Embedded$ python3 train_ie.py
Accuracy KNN: 94.83%
Accuracy SVM: 89.66%
```

Figura 18. Precisión modelos entrenados

3.2.3 Solución final

A continuación, se muestran los sistemas completados a partir de los subsistemas mencionados con anterioridad.

3.2.3.1 Sistema de reconocimiento facial con MTCNN y FaceNet

La primera propuesta es el sistema de reconocimiento facial tal y como se lleva a cabo en el proyecto de GitHub, donde la detección facial se realiza con MTCNN y el reconocimiento con FaceNet

Tras haber realizado la *rosificación del código* y el reentrenamiento del modelo, se lanza el nodo de la detección junto con el de la cámara. Además, se añade la publicación del nombre de las personas reconocidas por un *topic* de ROS.

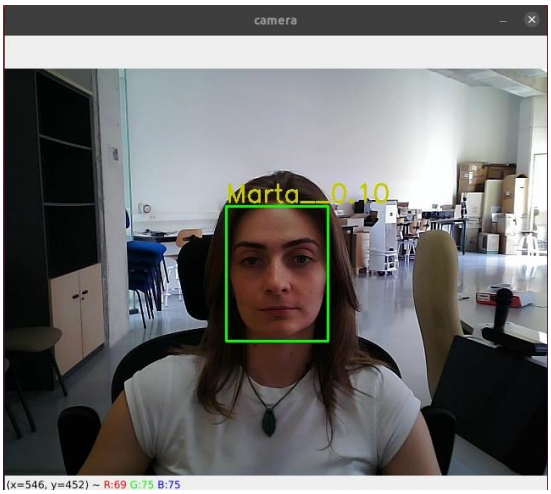

Figura 19. Reconocimiento facial con FaceNet

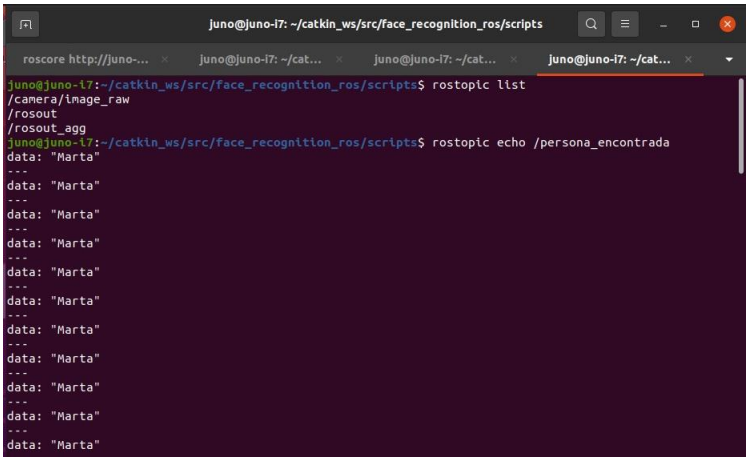

Figura 20. Topic de personas detectadas

El resultado obtenido es correcto y preciso en la gran mayoría de las ocasiones. No obstante, la desventaja que presenta esta solución es la ralentización del vídeo. Se crean varios segundos de retraso entre realizar un movimiento y ser mostrado por pantalla, lo que no lo hace eficiente a la hora del mapeo de personas en tiempo real para la evitación de obstáculos. Este inconveniente lleva al intento de mejorar la velocidad de procesamiento empleando Mediapipe Face Detection para la detección de los rostros en vez de MTCNN, como sugería el proyecto de referencia.

Aunque se consigue una pequeña mejora, no se reduce el delay de forma tan significativa como para ser considerado una solución aceptable para el fin establecido, por lo que se investiga otra manera de realizar el sistema de reconocimiento.

3.2.3.2 Sistema de reconocimiento facial con Mediapipe y modelo KNN

La alternativa propuesta frente al sistema que usaba MTCNN/Mediapipe y FaceNet es un sistema de reconocimiento que emplea Medipipe tanto como subsistema de reconocimiento como de detección, aplicando en este último un modelo KNN.

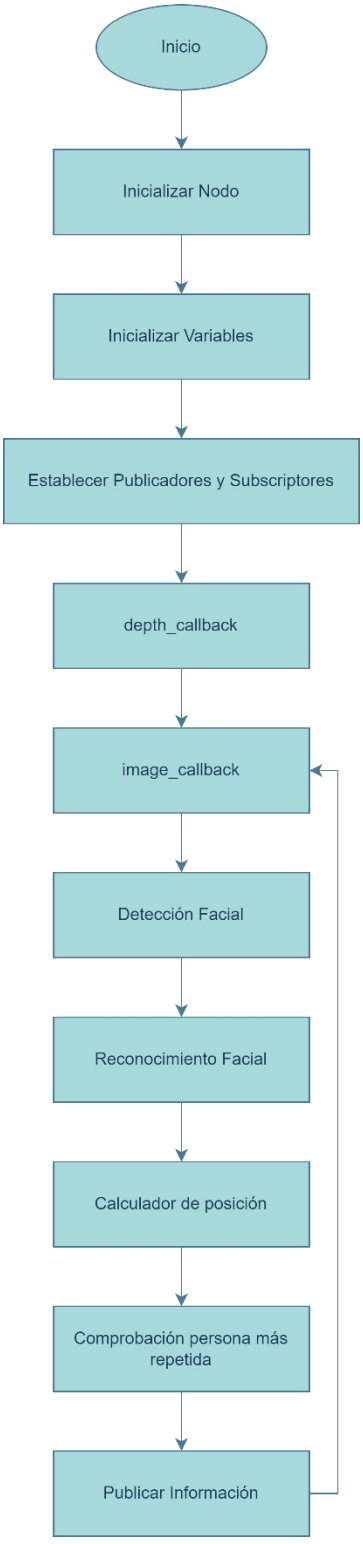

Figura 21. Diagrama de flujo sistema de reconocimiento facial con Mediapipe y KNN

Partiendo del modelo ya entrenado, se debe lanzar el nodo de ROS junto al nodo de la cámara. El funcionamiento del nodo en rasgos generales se describe a continuación.

Lo primero que hace el sistema es llamar al subsistema de detección facial al recibir una imagen por el *topic* de la cámara. Este subsistema obtiene la ROI del rostro

detectado, así como las coordenadas en píxeles de la *bounding box* y de un punto de referencia situado en el centro de la ROI que se utilizará para obtener la distancia.

Esta información se le envía al subsistema de detección facial, que pasa la ROI por el modelo KNN para obtener su predicción. Se almacena el nombre de la persona identificada y se procede a obtener la posición respecto a la cámara en la que se encuentra la persona identificada.

Para ello, se debe obtener la imagen de profundidad que acompaña a la imagen de color empleada para la detección facial. Con el punto de referencia obtenido con el paso por el primer subsistema, se consigue la distancia a este punto con la imagen de profundidad.

Para terminar, aplicando trigonometría, se obtienen las coordenadas reales de este punto de referencia respecto a la cámara, pues anteriormente se encontraban normalizadas. Toda la nueva información conseguida se almacena en una variable.

Además, para optimizar el funcionamiento del sistema se ofrece el resultado de la persona que más veces ha sido reconocida en 10 tomas seguidas de fotos. Para ello se ha implementado una función que mantiene un contador y que finalmente devuelve la persona más veces identificada junto a todos sus datos.

El sistema devuelve el vídeo en tiempo real, esta vez sin *delay* a diferencia de la solución anterior, con la actualización de la etiqueta asignada a cada rostro además de publicar toda la información de forma estructurada por unos *topics* para el uso de esta por parte de otros nodos.

Por último, cabe destacar que sólo se inicia el sistema de reconocimiento con una de cada cinco fotos que se reciben (cada 0.2 segundos) para así poder evitar una ralentización del nodo por procesar demasiada información. Esto, junto al cálculo de la persona más veces reconocida cada 10 secuencias, hace que se publiquen los datos recaudados de las personas reconocidas continuamente cada 0.53 segundos aproximadamente.

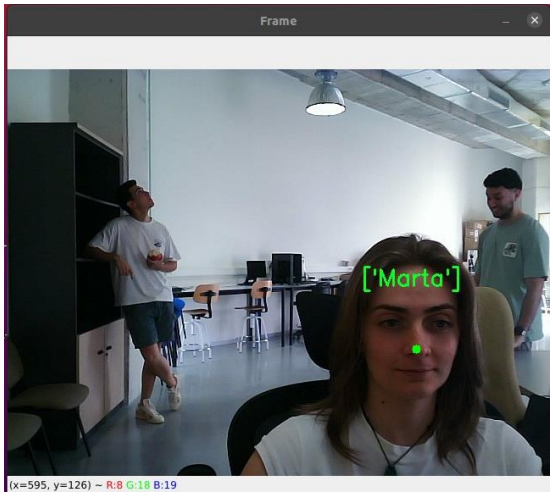

Figura 22. Reconocimiento facial con Mediapipe

La única desventaja que presenta esta alternativa es la menor robustez frente a los cambios de las condiciones de las imágenes. No obstante, esto podría solventarse con un mayor entrenamiento del modelo o con emplear otro tipo de modelo.

3.3 Ubicación de rostros en mapa semántico.

El concepto de mapa semántico puede interpretarse o bien como un mapa en el que lo representado posee cierta información sobre el mundo real, o como la representación de una noción a través cierta información estructurada.

Para crear un mapa semántico con los datos de las personas reconocidas, se han abordado dos alternativas con ambas de las formas descritas.

3.3.1 Mapa semántico mediante estructura de datos

Para llevar a cabo un mapa semántico a través de una estructura de datos se ha decidido crear un nuevo tipo de mensaje de ROS, con el que pueden ser publicados en *topics* y de forma organizada todos los datos de las personas reconocidas. El nodo encargado de ello será el del Sistema de reconocimiento facial con Mediapipe y modelo KNN.

La estructura de datos publicada viene definida por una serie de nuevos mensajes de ROS creados para dar cabida a todos los datos que pueden ser relevantes para otras aplicaciones. El tipo de mensaje definitivo que se publica es *Detectados.msg* que, a su vez, es un array de mensajes tipo *KeyValue.mgs. KeyValue.mgs* es un diccionario que contiene como claves las coordenadas de la *bounding box* de la persona cuyos datos se recogen en valor como un mensaje tipo *Detectado.msg*.

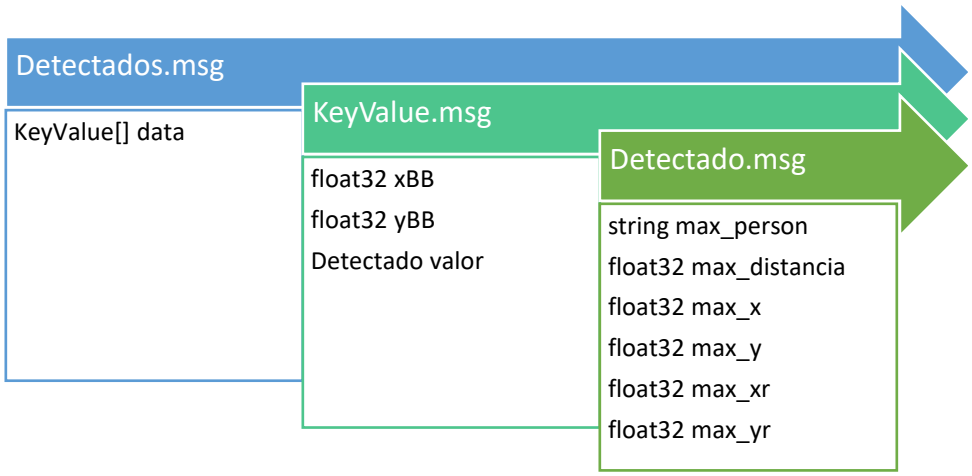

Figura 23. Estructura interna mensaje Detectados.msg

El porqué de esta estructura nace de la necesidad de comprobar que sólo es posible encontrar a una persona en unas coordenadas en concreto. Por ello, *Detectado.msg* contiene todos los datos de la persona más veces identificada en esas coordenadas durante 10 *frames*, y como señal de que esa coordenada ya está "ocupada" se establecen las coordenadas de la *bounding box* como clave del diccionario.

Finalmente, la estructura de datos en el *topic* al que son publicados quedaría, para la detección en este caso de dos personas al mismo tiempo, de la siguiente forma.

```
data:
  -
    xBB: 315.0
    yBB: 336.0
    valor:
      max_person: "['Adrian']"
      max_distancia: 1.392493486404419
      max_x: 1.392493486404419
      max_y: -0.07791940867900848
      max_xr: 356.0
      max_yr: 377.0
  -
    xBB: 498.0
    yBB: 264.0
    valor:
      max_person: "['Marta']"
      max_distancia: 0.0
      max_x: 0.0
      max_y: -0.24365206062793732
      max_xr: 566.0
      max_yr: 332.0
```

Figura 24. Ejemplo mensaje Detectados.msg

A continuación, se muestra una imagen del funcionamiento completo del sistema de reconocimiento facial con la publicación de los datos.

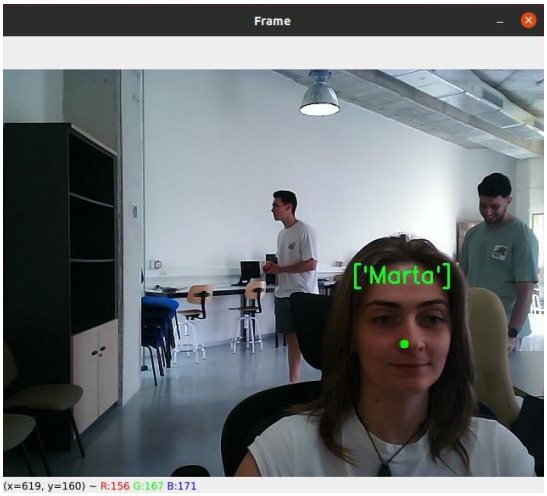

Figura 25. Reconocimiento facial con Mediapipe

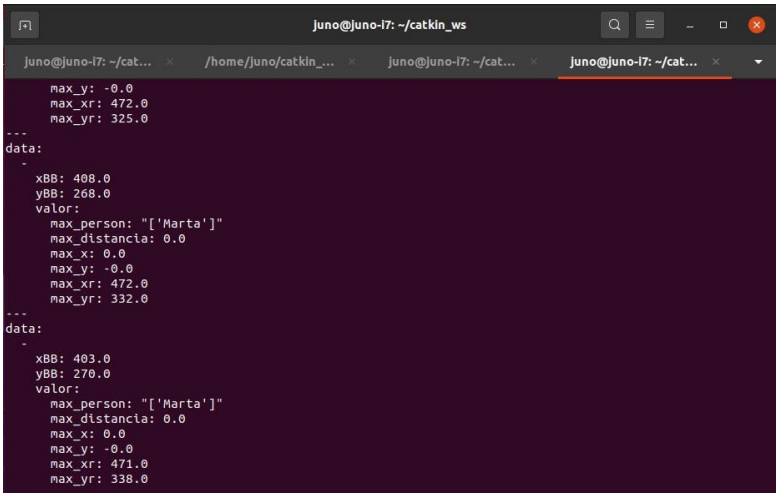

Figura 26. Topic personas detectadas

3.3.2 Mapa semántico mediante *grid*

La idea en la que se basa esta solución consiste en ubicar en un mapa tipo grid a las personas reconocidas aprovechando la estructura de datos *Detectados.msg* creada y descrita en el capítulo 3.3.1 más atrás. Para ello, primero ha sido necesario realizar unas modificaciones en un paquete de ROS desarrollado por la Dra. Nieves Pavón Pulido.

Este paquete inicialmente permite lanzar un nodo que recoge la información de obstáculos detectados por el láser que lleva incorporado el robot y los sitúa en el mapa. Es decir, es capaz de realizar la transformación de coordenadas respecto al láser a coordenadas respecto al mapa y su correspondiente conversión a número de celdas del mapa grid.

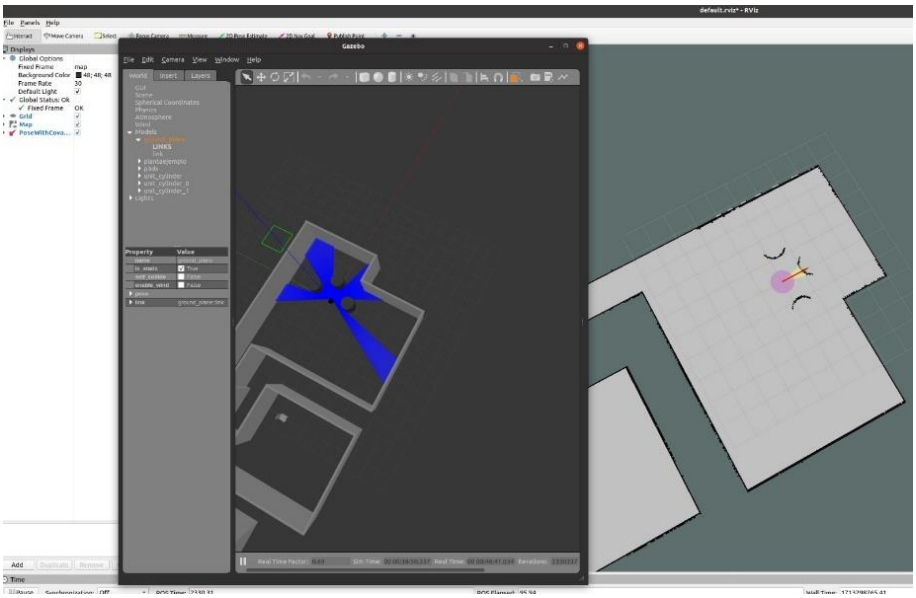

Figura 27. Ejemplo de uso del nodo de mapeo con láser

Aprovechando que realiza esa funcionalidad, se crea una clase nueva en C++ que, en vez de recoger los datos del láser, se subscriba al *topic* donde el sistema de reconocimiento facial está publicando el mensaje *Detectados.msg.* Para ello ha sido necesario realizar las modificaciones oportunas en el CMakeLists.txt del paquete para permitir el uso de ese nuevo tipo de mensaje.

Con la altura proyectada al suelo, tras haber realizado las rotaciones necesarias entre los sistemas de referencia de la cámara y del robot ($Eje\ x_{robot} = Eje\ z_{cámara}$, $Eje\ y_{robot} = Eje\ x_{cámara}$) a la hora de asignar la información de los detectados al mensaje y haber creado un bucle para poder representar en el mapa a todas las personas que se encuentran frente a la cámara a la vez, se recoge la información escuchada del topic y se sitúan en el mapa según el funcionamiento inicial del nodo.

No obstante, de cara a la evitación de obstáculos, se ha añadido una máscara de radio de 5 celdas alrededor de la coordenada en la que se sitúa a la persona, de manera que

se respete un cierto margen entre una posible trayectoria del robot y el espacio personal.

El resultado de las modificaciones de este nodo puede verse en RViz, una herramienta para ROS que permite visualización 3D y simulación para robots.

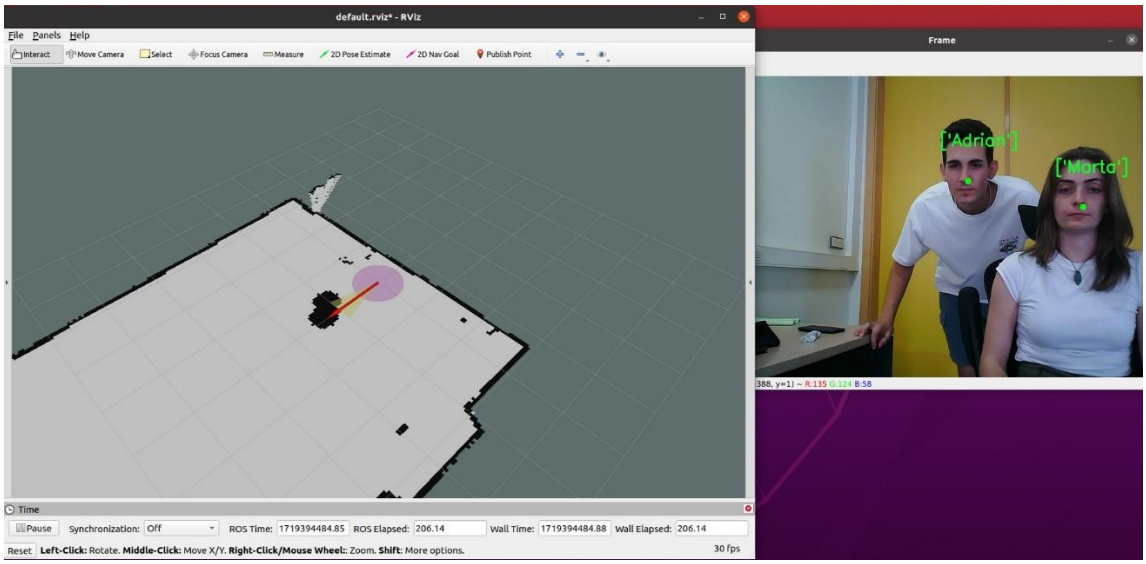

Figura 28. Visualización por RViz del mapa de obstáculos con el sistema de reconocimiento facial

Por último, para intentar ir más allá se ha tratado de dotar de cierto significado a cada celda ocupada. De esta manera, cada vez que se ubique a una persona en una celda, esta se pintará de un color codificado según su nombre, en vez de solo en blanco o negro según su ocupación. Aunque no se ha seguido avanzando más en esa funcionalidad, se muestran los avances conseguidos pues puede ser interesante en un futuro para obtener mapas de ocupación más complejos para cada persona y estudiar así su comportamiento.

En la siguiente imagen se puede observar cómo el mapa en RViz representa con colores las diferentes personas a las que va ubicando a su alrededor. Por ejemplo, el color amarillo representa a “Marta” y el gris a “Adrián”.

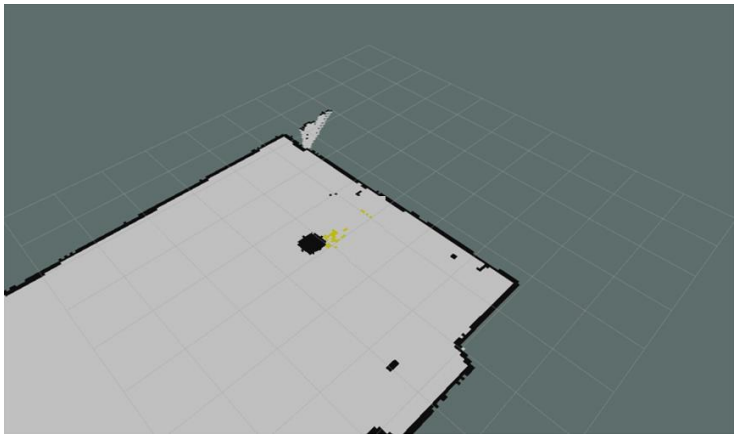

Figura 29. Ejemplo mapa semántico en RViz

Por último, se puede mostrar un esquema proporcionado por *rqt*, una herramienta de ROS que permite visualizar el funcionamiento y la comunicación entre nodos que se encuentran en ejecución.

Figura 30. Diagrama *rqt* para mapeo

3.4 Diseño del sistema de memoria.

En este aparatado se va a explicar cómo se ha realizado una aproximación a la capacidad de memoria de una arquitectura cognitiva, inspirada en el Modelo de Almacenamiento Múltiple de la memoria humana, propuesto por Atkinson y Shiffrin (1968) (Al-Faris, 2021).

Este modelo propone que la información recibida del entorno atraviesa tres etapas: Registro Sensorial (RS), Memoria a Corto Plazo (MCP) y finalmente Memoria a Largo Plazo (MLP). El RS es una forma de memoria donde la información es captada por los receptores y procesada por el sistema nervioso, con una capacidad ilimitada para filtrar información sensorial en una fracción de segundo. El RS forma parte del proceso de percepción y está relacionado con los cinco sentidos, aunque solo la información relevante pasa del RS a la MCP. En el RS, la información se almacena brevemente antes de ser transferida a la MCP, y se subdivide en Memoria Icónica (visual), Memoria Ecoica (auditiva) y Memoria Háptica (táctil).

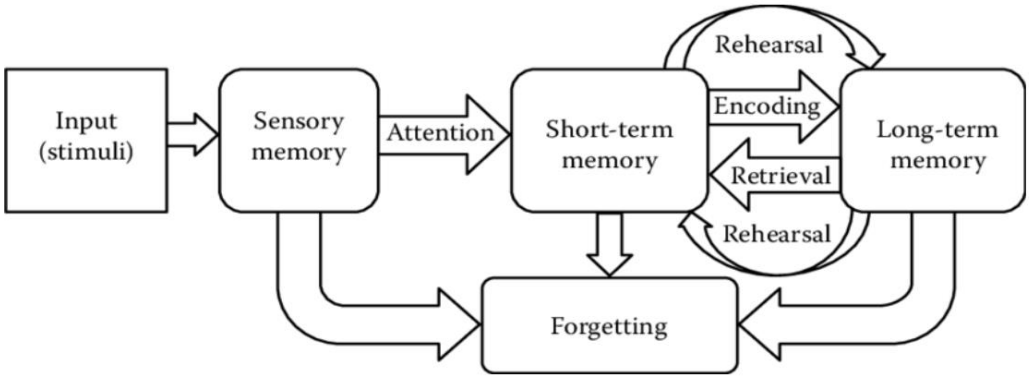

Figura 31. Diagrama Modelo de Almacenamiento Múltiple. (Al-Faris, 2021)

La Memoria Icónica retiene estímulos visuales por menos de un segundo, mientras que la Memoria Ecoica, más duradera, almacena sonidos y ritmos. La Memoria Háptica retiene sensaciones táctiles por 1-2 segundos y depende de otras señales como la visión o el movimiento.

Según Atkinson y Shiffrin, la MCP es un almacenamiento temporal para tareas complejas como la comprensión y el razonamiento. Baddeley y Hitch (1974) sugirieron que la MCP consta de subsistemas interactuantes llamados Memoria de Trabajo (MT), capaz de manejar hasta siete elementos de información relevante durante aproximadamente 10 segundos, a menos que se repase.

La MLP tiene una capacidad y duración virtualmente ilimitadas, permitiendo recordar información de hace mucho tiempo sin necesidad de eliminar datos antiguos para almacenar nuevos. Este modelo de procesamiento de información describe el flujo desde los estímulos ambientales, a través del RS y la MT, hasta la MLP o el olvido.

Para el sistema de memoria, pueden realizarse los siguientes símiles a las etapas descritas.

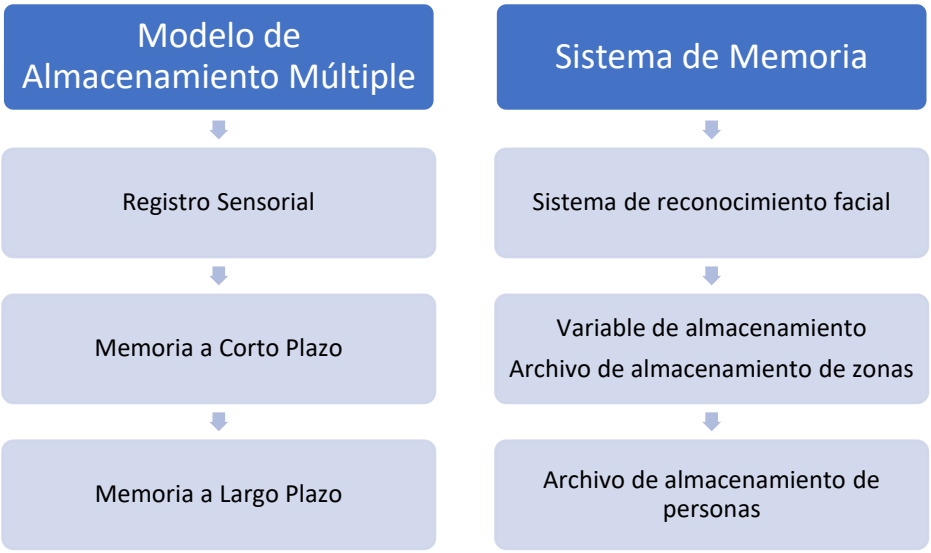

Figura 32. Comparación Modelo Almacenamiento Múltiple y Sistema de Memoria

El sistema de memoria se ha desarrollado como un nuevo paquete de ROS, que consta de un nodo escrito en Python.

A nivel general, este nodo funciona recogiendo la información del topic de las personas detectadas y almacenándola, tras ser procesada, en un archivo teniendo en cuenta 3 franjas horarias.

Cada vez que una persona se identifica se almacena su nombre y se añaden, en la franja horaria correspondiente, las coordenadas donde ha sido encontrada (transformadas a coordenadas reales del mapa), con un contador de cuantas veces se ha reconocido dentro de un área (área con origen esa primera coordenada y límite de un radio de 1 metro) y con la fecha en la que se ha encontrado por última vez.

```
'Marta':
 - 9-13:
 - 13-19:
   -- (0.15275831654032332, 0.040250026781133358); 50 veces, 2024-
06-13
   -- (1.370947372629379, -0.008010837196756675); 38 veces, 2024-
06-13
 - 19-21:
'Adrian':
 - 9-13:
 - 13-19:
   -- (1.36213686789464, 0.05595672494770871); 2 veces, 2024-06-13
 - 19-21:
```

Figura 33. Ejemplo archivo de almacenamiento de personas

Por otro lado, se va creando un registro de todas las zonas por las se pasa ese día y en cada franja. Gracias a este archivo, al lanzar cada nuevo día el sistema, se comprueba las zonas por las que se pasó el día anterior, de manera que si, por ejemplo, el 15/06/2024 se tiene a María registrada en la franja 9-13h en unas coordenadas pertenecientes al área 1 y se pasó el día 21/06/2024 por esa área en esa misma franja, se tiene el indicio de que María no suele encontrarse en esa zona en esa franja normalmente, por lo que se le decrementa el contador para esa posición concreta.

De esta manera, se logra potenciar el recuerdo de las personas que suelen encontrarse en una zona concreta en una franja horaria concreta con regularidad, mientras que el resto de las personas que se van encontrando casuísticamente se van olvidando.

El funcionamiento del nodo, que puede verse resumido en la Figura 34. Flujograma nodo sistema de memoria, comienza por definir la lista de áreas del mapa que se pueden recorrer, así como crear (si no existen de antes) los dos archivos para el almacenamiento de información: *memoria_output.txt* para la MLP de personas y *zonas_memoria_output.txt* para la MCP de zonas recorridas.

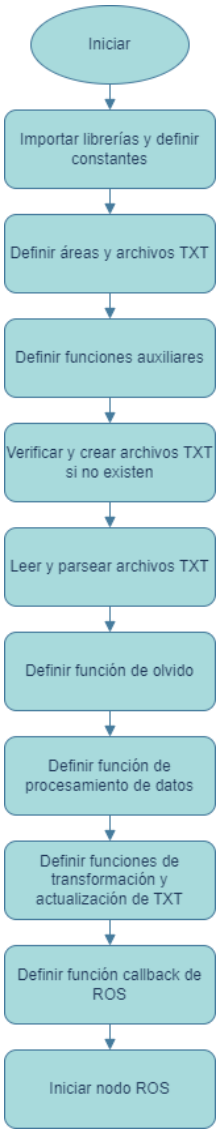

Figura 34. Flujograma nodo sistema de memoria

Tras ello, se *parsean*[2] ambos archivos de almacenamiento y con toda la información que contienen, se inicializan dos variables: *data*_store para personas identificadas y *zonas_store* para zonas recorridas, para así poder realizar la función del olvido progresivo de las personas que se encontraron puntualmente en una ubicación.

Una vez se actualiza el archivo con la "información olvidada", se pone en funcionamiento la función de procesamiento de los datos provenientes del nodo del sistema de reconocimiento facial con el mensaje tipo *Detectados.msg.* Es en esta función donde se gestiona si es la primera vez que se encuentra a esta persona en esas coordenadas y en esa franja o si ya se ha encontrado más veces dentro del área circular de radio 1 metro.

[2] Parsear: proceso de analizar una cadena de texto para identificar su estructura sintáctica y extraer información significativa de ella («Parseo de datos en Internet», 2012)

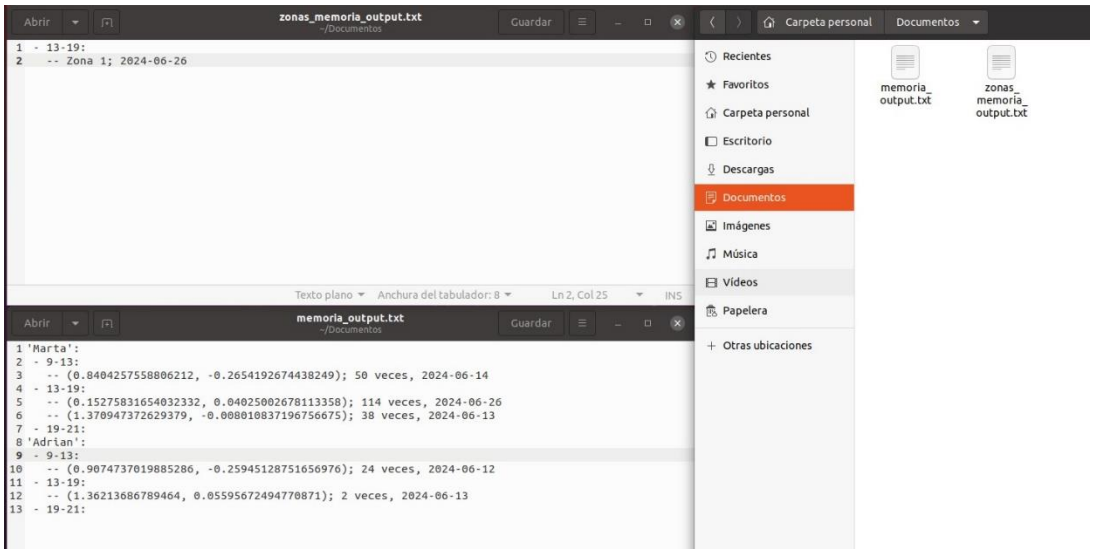

Figura 35. Archivos de memoria de la arquitectura cognitiva inicial

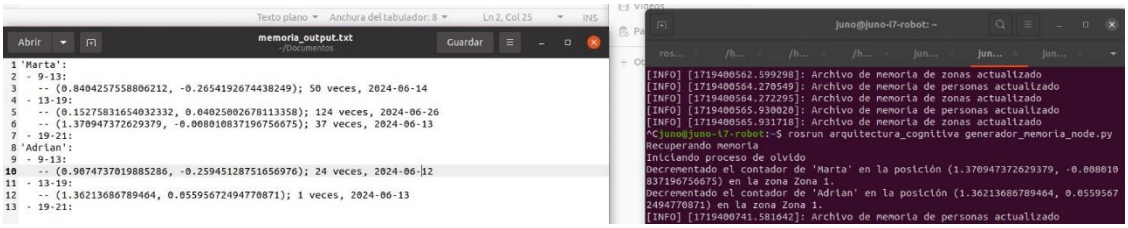

Figura 36. Archivos de memoria actualizados con la función olvido

Con cada iteración del *script* las personas se van, o añadiendo como nuevas entradas, o modificando los contadores de las ya existentes en la variable *data_store,* que a su vez lo vuelca en el archivo de memoria. De esta manera, con el uso prolongado de este sistema, se dispone de una memoria robusta que permitiría predecir la ubicación de una persona dentro de las franjas horarias establecidas con alta seguridad, permitiendo al robot poder realizar tareas de búsqueda y vigilancia de los residentes.

3.5 Integración en la plataforma.

Para realizar la integración de todos los elementos que componen la arquitectura cognitiva en la plataforma, se han ido *rosificando* todos los nodos pertinentes para poder compilar todos los paquetes dentro del espacio de trabajo de ROS del propio robot.

Una vez se encuentran todos los paquetes listos, es posible poner toda la arquitectura en ejecución. Para ello se deben lanzar todos los nodos mencionados, así como algunos ya desarrollados con anterioridad por la Dra. Nieves Pavón, como el responsable del mapeo o del controlador de los motores del robot.

Para concluir este apartado, se muestra el esquema que genera la herramienta *rqt* con el funcionamiento de todos los nodos necesarios para activar la arquitectura cognitiva.

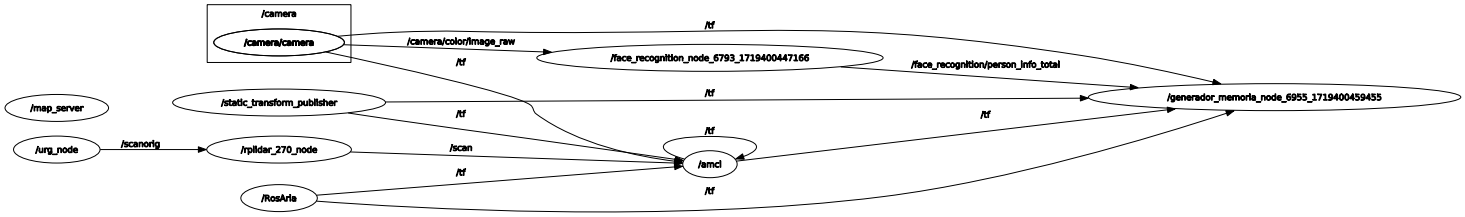

Figura 37. Diagrama rqt del sistema integrado

4 PRUEBAS Y RESULTADOS.

En este capítulo se muestran los resultados más significativos y relevantes que han sido detectados con las pruebas realizadas en laboratorio.

Inicialmente, se realizó una prueba con la rosificación de un sistema de reconocimiento facial utilizando FaceNet. Este sistema demostró ser muy robusto en términos de precisión y reconocimiento facial. Sin embargo, uno de los principales inconvenientes encontrados fue el considerable retardo (delay), incluso después de sustituir MTCNN por Mediapipe para la detección facial.

Posteriormente, se desarrolló un sistema de reconocimiento facial utilizando Mediapipe y un modelo KNN. Este sistema, aunque ligeramente menos robusto en comparación con FaceNet, presentó una ventaja significativa: la ausencia total de delay en el procesamiento. Este aspecto es crucial para aplicaciones en tiempo real donde la velocidad de respuesta es fundamental.

Al comparar el sistema basado en FaceNet con el desarrollado mediante Mediapipe, se observaron varias diferencias clave. El modelo KNN utilizado con Mediapipe tiene un peso muy reducido en comparación con el modelo FaceNet, lo que facilita su implementación y ejecución en dispositivos con recursos limitados. Además, mientras FaceNet presentaba un notable delay, Mediapipe operaba sin ningún retraso, mejorando así la experiencia de usuario. El sistema de reconocimiento facial con Mediapipe destaca por ser una solución innovadora, y de desarrollo 100% propio, lo que la hace totalmente customizable y permite adaptarla a las necesidades específicas del proyecto. Otra ventaja importante es su capacidad para proporcionar anonimización de datos, una característica vital para la protección de la privacidad.

En cuanto a la integración de usuarios en el mapa de obstáculos y el mapa semántico, se logró incorporar exitosamente a las personas reconocidas alrededor en el mapa de obstáculos (Figura 28), utilizando como base un mapa semántico de estructura de datos (Figura 26). Asimismo, se modificó el nodo del mapeo anterior para proponer un mapa semántico que almacena información por celda (Figura 29), lo cual supone otra línea a través de la cual abarcar la tarea de desarrollo de una arquitectura cognitiva para la memorización de usuarios.

En términos de la aproximación a una arquitectura cognitiva, se alcanzaron capacidades de memoria (Figura 35) y de olvido (Figura 36), fundamentales para una operación efectiva y eficiente. El diseño resultante es sencillo e intuitivo, facilitando su uso y mantenimiento.

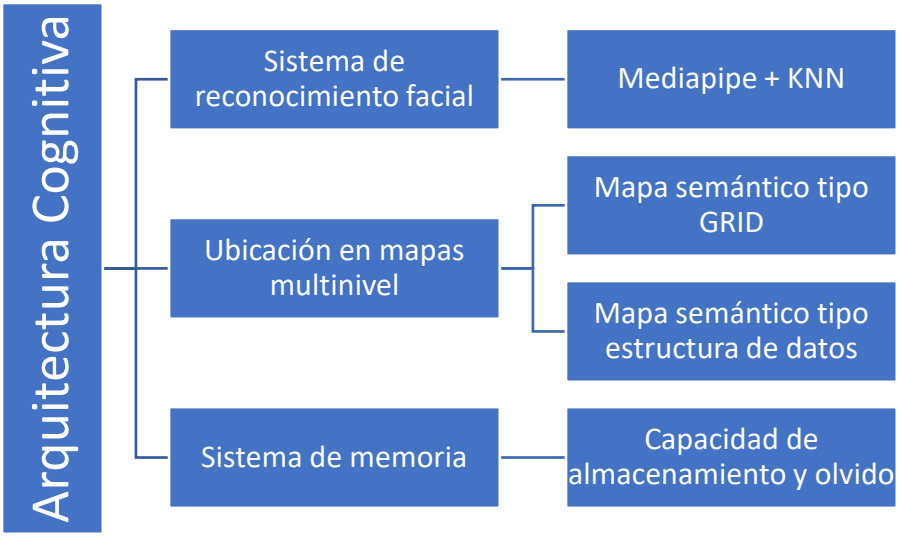

Figura 38. Descripción general de la arquitectura cognitiva

Entre las ventajas del sistema desarrollado destacan las siguientes:

- Modificabilidad de todos los nodos, permitiendo ajustes y mejoras continuas según las necesidades del entorno.
- Portabilidad del sistema. Puede ejecutarse en cualquier PC con TensorFlow y ROS, facilitando su implementación en diversos contextos y dispositivos.
- Sistema de reconocimiento facial sin delay. Se consigue una respuesta rápida y suficientemente robusta.
- Anonimización de datos. Permite establecer una cierta capa de abstracción para proteger los datos de los usuarios.

No obstante, el sistema presenta algunas limitaciones. Pueden enumerarse las siguientes:

- Posible necesidad de almacenamiento en la nube tanto del modelo KNN como de los archivos de almacenamiento de la arquitectura cognitiva, para asegurar una operación continua y eficiente.
- Desarrollo de una interfaz gráfica que permita trabajar con el *dataset* y el entrenamiento del modelo KNN de manera más intuitiva.
- Realizar pruebas en residencias de ancianos reales, imprescindible para validar y ajustar el sistema en entornos operativos concretos.

En conclusión, los resultados obtenidos indican que la arquitectura cognitiva desarrollada tiene un gran potencial para mejorar la eficiencia y la calidad de los servicios de la plataforma robótica asistencial.

5 CONCLUSIONES.

El desarrollo de una arquitectura cognitiva para la integración de usuarios humanos dentro de un sistema de mapas multinivel en una plataforma robótica asistencial ha demostrado ser un avance significativo en la mejora de la eficiencia y calidad de los servicios en residencias de ancianos. A lo largo de este trabajo, se han obtenido resultados relevantes y prometedores a partir de pruebas realizadas en laboratorio, que han permitido validar y refinar los componentes del sistema.

Uno de los principales logros ha sido el diseño y la implementación de un sistema de reconocimiento facial robusto y eficiente. Inicialmente, el uso de FaceNet, aunque preciso, presentó desafíos importantes debido a un considerable delay en el procesamiento. La transición a un sistema basado en Mediapipe y un modelo KNN, a pesar de ser ligeramente menos robusto, eliminó este delay, permitiendo una respuesta en tiempo real crucial para aplicaciones prácticas en entornos asistenciales. Además, la capacidad de este sistema de proporcionar anonimización de datos añade una capa vital de protección de la privacidad, cumpliendo con las normativas de protección de datos.

En términos de integración, se logró incorporar exitosamente a las personas reconocidas en el mapa de obstáculos y se propuso un mapa semántico que o bien almacena información por celda, o bien la publica siguiendo una estructura de datos establecida, enriqueciendo la capacidad del sistema para comprender y reaccionar ante su entorno. Estas mejoras son fundamentales para el desarrollo de una arquitectura cognitiva que emule la memorización y el olvido humanos, lo que facilita una operación efectiva y eficiente.

El sistema desarrollado se considera customizable y portable, características que hacen que sea adaptable y fácil de implementar en diversos contextos y dispositivos. Tiene un gran potencial y los resultados obtenidos hasta ahora son prometedores. Esta arquitectura cognitiva puede ofrecer soluciones innovadoras y efectivas para los desafíos actuales en la atención a personas mayores

En un marco de futuro en el que se trabajen las limitaciones detectadas, como son el almacenamiento en la nube de archivos y la implementación de una interfaz gráfica de entrenamiento, se espera que la arquitectura cognitiva planteada quede desarrollada por completo.

Por último, la validación del sistema en residencias de mayores reales es imprescindible para ajustar y confirmar su eficacia en entornos operativos concretos.

6 BIBLIOGRAFÍA.

Adjabi, I., Ouahabi, A., Benzaoui, A., & Taleb-Ahmed, A. (2020). Past, Present, and Future of Face Recognition: A Review. *Electronics*, *9*(8), Article 8. https://doi.org/10.3390/electronics9081188

Al-Faris, S. (2021). *Perspectives of Human Memory Models: A Critical Review*.

Andres-Ubeda_Practicas_RoboticaServicios.pdf. (s. f.). Recuperado 20 de mayo de 2024, de https://rua.ua.es/dspace/bitstream/10045/87768/1/Andres-Ubeda_Practicas_RoboticaServicios.pdf

Astra Series. (s. f.). *ORBBEC - 3D Vision for a 3D World*. Recuperado 20 de junio de 2024, de https://www.orbbec.com/products/structured-light-camera/astra-series/

Cubi 5 12M | Best Mini Desktop PC 0.7L | Be Your Window To The World. (s. f.). Recuperado 20 de junio de 2024, de https://es.msi.com/Business-Productivity-PC/@Request:fullUrl()

Guía de detección de rostro | Google AI Edge. (s. f.). Google for Developers. Recuperado 11 de junio de 2024, de https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector?hl=es-419

Guía de detección de rostro para Python | Google AI Edge. (s. f.). Google for Developers. Recuperado 11 de junio de 2024, de https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector/python?hl=es-419

Guía de soluciones de MediaPipe | Google AI Edge. (s. f.). Google for Developers. Recuperado 11 de junio de 2024, de https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419

Hasan, M. K., Ahsan, M. S., Abdullah-Al-Mamun, Newaz, S. H. S., & Lee, G. M. (2021). Human Face Detection Techniques: A Comprehensive Review and Future Research Directions. *Electronics*, *10*(19), Article 19. https://doi.org/10.3390/electronics10192354

Kotseruba, I., & Tsotsos, J. K. (2020). 40 years of cognitive architectures: Core cognitive abilities and practical applications. *Artificial Intelligence Review*, *53*(1), 17-94. https://doi.org/10.1007/s10462-018-9646-y

Laird, J. (2022). *Introduction to the Soar Cognitive Architecture 1*. https://doi.org/10.13140/RG.2.2.31776.80644

Langley, P., Laird, J. E., & Rogers, S. (2009). Cognitive architectures: Research issues and challenges. *Cognitive Systems Research*, *10*(2), 141-160. https://doi.org/10.1016/j.cogsys.2006.07.004

Memoria episódica y envejecimiento cognitivo. (s. f.). Recuperado 22 de mayo de 2024, de https://www.fundacioncien.es/noticias/memoria-episodica-y-envejecimiento-cognitivo

Opinión, L. (2024, febrero 2). *La UPCT crea un robot que ayuda a los mayores a prevenir caídas*. La Opinión de Murcia. https://www.laopiniondemurcia.es/cartagena/2024/02/02/upct-crea-robot-ayuda-mayores-97677214.html

Parseo de datos en Internet: Todo lo que deberías conocer. (2012, octubre 17). *Canal Gestión Empresarial*. https://www.inesem.es/revistadigital/gestion-empresarial/el-parseo-de-datos-en-internet-nuevas-oportunidades-comerciales-al-filo-de-la-legalidad/

Pavón-Pulido, N., Blasco-García, J. D., López-Riquelme, J. A., Feliu-Batlle, J., Oterino-Bono, R., & Herrero, M. T. (2023). JUNO Project: Deployment and Validation of a Low-Cost Cloud-Based Robotic Platform for Reliable Smart Navigation and Natural Interaction with Humans in an Elderly Institution. *Sensors*, *23*(1), Article 1. https://doi.org/10.3390/s23010483

Ratul, M. T. A., Mahmud, M. S. A., Abidin, M. S. Z., & Ayop, R. (2021). Design and Development of GMapping based SLAM Algorithm in Virtual Agricultural Environment. *2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*, 109-113. https://doi.org/10.1109/ICCSCE52189.2021.9530991

Research on Face Detection Technology Based on MTCNN. (s. f.). Recuperado 18 de junio de 2024, de https://ieeexplore.ieee.org/abstract/document/9239720?casa_token=K5XUmJ

UbVVwAAAAA:ygk_dcQHaJSBteVRU4fqGLwJVeFEMYIBro260TRveSOim2yFCzN8MahXF6JRebttCxRrFs7k

Ritter, F. E., Tehranchi, F., & Oury, J. D. (2019). ACT-R: A cognitive architecture for modeling cognition. *WIREs Cognitive Science*, *10*(3), e1488. https://doi.org/10.1002/wcs.1488

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 815-823. https://doi.org/10.1109/CVPR.2015.7298682

Wang, M., & Deng, W. (2021). Deep face recognition: A survey. *Neurocomputing*, *429*, 215-244. https://doi.org/10.1016/j.neucom.2020.10.081