

Jornadas de Automática

Sistema de toma de decisiones jerárquico para robots de servicio

Hormigo-Jiménez, P.*, Valencia-Rojas, A., Cañete-Baena, A., Galindo, C., Gonzalez-Jiménez, J.

Grupo de Percepción Artificial y Robótica Inteligente (MAPIR), Dept. de Ingeniería de Sistemas y Automática, Instituto Universitario en Ingeniería Mecatrónica y Sistemas Ciberfísicos (IMECH.UMA), Universidad de Málaga, Blvr. Louis Pasteur, 35, 29071 Málaga, España.

To cite this article: Hormigo-Jiménez, P., Valencia-Rojas, A., Cañete-Baena, A., Galindo, C., González-Jiménez, J. 2026. Hierarchical Decision-Making System for Service Robots. *Jornadas de Automática*, 47. <https://doi.org/10.17979/ja-cea.2026.47.13815>

Resumen

El despliegue de robots de servicio en entornos compartidos exige arquitecturas de decisión capaces de gestionar interrupciones y mantener un funcionamiento continuo. Este artículo presenta un sistema de control jerárquico basado en Árboles de Comportamiento (BT) diseñado para la plataforma robótica *Sancho*. El sistema se estructura en cuatro capas de prioridad: gestión de energía, interrupciones conversacionales, ejecución de misiones de alto nivel y rutinas pasivas de reposo. Mediante su implementación en ROS 2 y el uso de una memoria compartida (*Blackboard*), el diseño separa la toma de decisiones de los módulos subyacentes de navegación, percepción e interacción. La evaluación experimental en un entorno físico demuestra la capacidad de respuesta de la plataforma ante cambios de contexto. El sistema permite interrumpir la navegación base para atender comandos de voz, integrar de forma fluida nuevas misiones encoladas y abortar la ejecución ante eventos críticos para asegurar la recarga, validando la viabilidad del diseño para una operación desatendida.

Palabras clave: Robótica inteligente, Robots móviles autónomos, Robótica móvil, Interacción humano-vehículo

Hierarchical Decision-Making System for Service Robots.

Abstract

The deployment of service robots in shared environments requires decision-making architectures capable of managing interruptions and maintaining continuous operation. This paper presents a hierarchical control system based on Behavior Trees (BT) designed for the *Sancho* robotic platform. The system is structured into four priority layers: energy management, conversational interruptions, high-level mission execution, and passive idle routines. Through its implementation in ROS 2 and the use of a shared memory (*Blackboard*), the design separates decision-making from the underlying navigation, perception, and interaction modules. Experimental evaluation in a physical environment demonstrates the platform's responsiveness to context changes. The system allows interrupting base navigation to attend to voice commands, seamlessly integrating newly queued missions, and aborting execution in the face of critical events to ensure recharging, validating the feasibility of the design for unattended operation.

Keywords: Intelligent robotics, Autonomous Mobile Robots, Mobile robots, Human and vehicle interaction

1. Introducción

El despliegue de robots de servicio en espacios compartidos exige sistemas de decisión que equilibren la ejecución de tareas, las normas sociales y la autoconservación Möller et al. (2021). La viabilidad de la plataforma no depende únicamen-

te del éxito en misiones puntuales, sino de su funcionamiento continuo. Esto requiere que el robot gestione interrupciones, respete el espacio personal y administre sus recursos sin necesidad de supervisión.

En la práctica, el despliegue requiere integrar una serie de

*Autor para correspondencia: pablohorjim@uma.es
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

capacidades fundamentales. Por un lado, el robot debe navegar de forma segura en entornos compartidos y adaptar sus trayectorias a la presencia de personas. Asimismo, necesita un sistema de diálogo que le permita comunicarse con los usuarios de forma natural y procesar instrucciones verbales. Por último, para mantener su operatividad continua, debe monitorizar sus niveles de energía y gestionar el retorno a su base para solicitar una recarga asistida. Sin embargo, el funcionamiento fluido de la plataforma no depende únicamente del desarrollo individual de estos módulos, sino de una gestión de prioridades eficaz. El sistema debe tomar decisiones en tiempo real, siendo capaz, por ejemplo, de suspender temporalmente una ruta si un usuario lo interpela, o de abortar una misión en curso para regresar a la estación de carga si la batería alcanza un nivel crítico.

Tradicionalmente, esta toma de decisiones se ha abordado mediante Máquinas de Estados Finitos (FSM). Sin embargo, este enfoque presenta serias limitaciones en cuanto a la escalabilidad y el mantenimiento cuando se trata de abordar interrupciones complejas. Como alternativa estructural, la robótica ha adoptado el uso de Árboles de Comportamiento (BT) Ögren and Sprague (2022). A diferencia de las FSM, los BT separan el control de flujo de la ejecución de las tareas. Utilizan una estructura en árbol donde una señal de control (*tick*) se propaga hacia los nodos hoja, los cuales se limitan a ejecutar su lógica y devolver un estado (*Éxito*, *Fallo* o *En Ejecución*). Al delegar las reglas de transición a los nodos de control superiores, los componentes son totalmente independientes. Esta separación hace que el sistema sea inherentemente modular, facilitando la gestión de interrupciones complejas y la integración de nuevas habilidades.

Aprovechando estas ventajas estructurales, el objetivo de este trabajo es diseñar y validar un sistema de control basado en prioridades para robots de servicio. Como núcleo de decisión, se propone una arquitectura jerárquica basada en un árbol de comportamiento sobre ROS 2 que organiza el flujo de ejecución en cuatro capas: (1) gestión de energía, (2) interrupciones humanas, (3) misiones de alto nivel y (4) reposo. Esta jerarquía asegura que los eventos más urgentes tomen el control del sistema de forma inmediata, permitiendo que la plataforma se adapte a un entorno dinámico sin comprometer su seguridad.

Para evaluar el diseño, el sistema se ha integrado en la plataforma móvil Sancho, del grupo de investigación MAPIR-UMA Cañete (2025). La validación experimental ha consistido en pruebas de operación en un entorno físico compartido, donde el árbol de comportamiento debe coordinar secuencias de navegación, procesar órdenes verbales de distintos usuarios y gestionar alertas de batería. Los resultados han confirmado que el modelo propuesto resuelve de forma determinista y segura los conflictos de prioridad, demostrando su viabilidad para el despliegue desatendido del robot.

El resto del documento se organiza de la siguiente manera. La Sección 2 detalla la plataforma robótica bajo estudio y sus capacidades integradas. A continuación, la Sección 3 expone el núcleo de control desarrollado para coordinar dichos módulos. La evaluación del caso de estudio y los resultados experimentales de la integración se discuten en la Sección 4, culminando en la Sección 5 con las conclusiones principales y las líneas de trabajo futuro.

2. Descripción general del sistema

La autonomía del robot se sustenta en la coordinación de sus distintos subsistemas. Por ello, a continuación se detallan el *hardware* de la plataforma y sus capacidades base.

2.1. Componentes hardware

El robot de servicio Sancho Cañete Baena et al. (2025) (ver figura 2.1 es un sistema móvil basado en un chasis omnidireccional *AgileX*, sobre el cual se integran los dispositivos sensoriales y de comunicación. La comprensión del entorno se basa en un esquema multimodal formado por: dos escáneres láser Hokuyo 2D para una cobertura perimetral de seguridad y para localización; una cámara frontal RGB-D destinada a la detección y localización tridimensional de usuarios durante la navegación interactiva; y una segunda cámara RGB de gran angular que, montada sobre una unidad *pan-tilt* en la zona superior (cuello), se emplea en el reconocimiento facial. Además, el prototipo incorpora una matriz de micrófonos capacitada para estimar la dirección de procedencia acústica (*Direction of Arrival*, DoA).

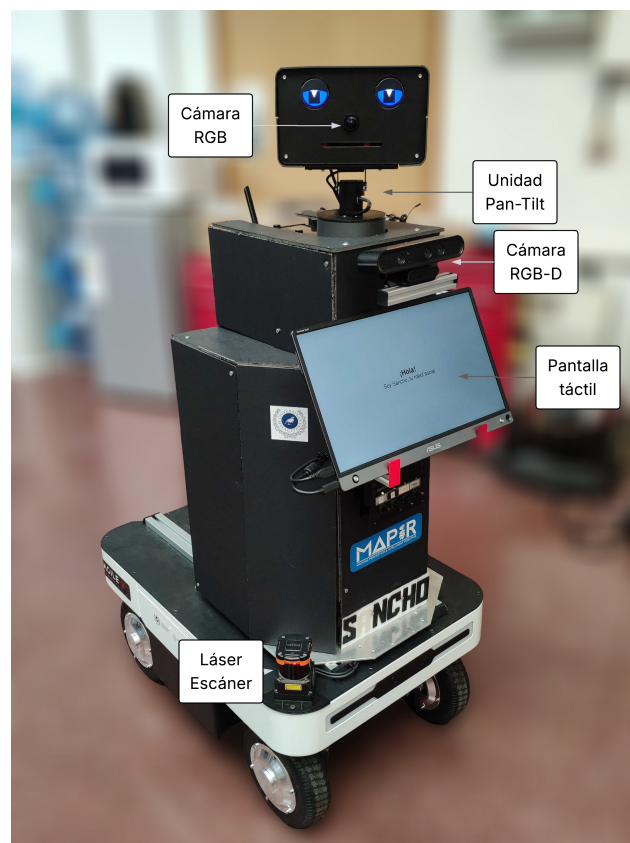


Figura 1: Robot de servicios Sancho. El robot está equipado con dos escáneres LiDAR para navegación, una cámara RGB-D para captura de color y profundidad, y una matriz de micrófonos. La estructura incluye una unidad *pan-tilt* a modo de cuello, una cámara RGB frontal y una interfaz de pantalla táctil para la interacción y retroalimentación del usuario.

A nivel de interacción, el robot integra un altavoz para la comunicación verbal, una pantalla táctil frontal y pantallas LED ubicadas en la cabeza, a modo de ojos, para dotar a la plataforma de expresividad y favorecer una interacción natural con el usuario.

El procesamiento se ejecuta localmente mediante un ordenador (AMD Ryzen 7, 32 GB RAM) sobre el sistema operativo Ubuntu 22.04 LTS y el *framework* ROS 2 Humble. Aunque este dispositivo es suficiente, la fluidez de la interacción con humanos es sensible a la latencia de los módulos con alto coste computacional. Por ello, la arquitectura permite descargar tareas de mayor demanda a servidores externos de computación en el borde (*edge computing*) Ambrosio-Cestero et al. (2026), lo que reduce significativamente los tiempos de respuesta.

2.2. Capacidades integradas

Para operar de forma autónoma y segura en un entorno compartido, el sistema integra diversas capacidades de percepción y navegación social previamente validadas en Cañete Baena et al. (2025); Cañete et al. (2024). En concreto, el robot utiliza la red neuronal *MoveNet* TensorFlow (2024) para la estimación de la pose y la inferencia de formaciones de grupos de usuarios en tiempo real. La integración de esta percepción habilita una navegación socialmente consciente (*socially-aware navigation*) Möller et al. (2021) que ajusta las trayectorias del robot para respetar los espacios próximos y evitar interrumpir las zonas de interacción de los grupos de personas en conversación. Asimismo, se incorporan rutinas de *gaze-shifting* que coordinan el movimiento de la cabeza y la base para reforzar la naturalidad y la atención compartida durante el contacto visual inicial.

Con respecto a la navegación segura en entornos complejos, el sistema cuenta con un método de generación automatizada de puntos de paso (*waypoints*) Moreno et al. (2020) que establece una topología de navegación que permite al robot cruzar puertas y espacios estrechos de manera segura. Este método está adaptado e integrado dentro del paquete Nav2 Macenski et al. (2023)

En el ámbito de la interacción humano-robot (HRI), el sistema combina modalidades visuales y acústicas. Por la parte visual, es capaz de realizar tareas de reconocimiento facial para la identificación de usuarios a largo plazo. Por la parte acústica, incorpora módulos de procesamiento y gestión conversacional detallados en trabajos previos Quemada-Torres (2025). La reactividad social del robot se fundamenta en un modelo de escucha continua implementado mediante *OpenWakeWord* Scripka (2026), configurado para la detección de la palabra clave (*wakeword*) “Sancho”. De forma complementaria a esta alerta, el robot está dotado de un subsistema de capacidades conversacionales que abarca desde la síntesis y reconocimiento de voz (ASR y TTS) hasta la integración con Modelos Grandes de Lenguaje (LLM). Para dotar a este modelo de capacidad de control sobre la plataforma, el sistema incorpora un servidor basado en el protocolo MCP (*Model Context Protocol*), el cual permite traducir las intenciones extraídas del lenguaje natural en comandos directamente ejecutables por la plataforma (como la asignación de misiones de alto nivel). Este conjunto de herramientas conforma la interfaz de comunicación utilizada por el robot.

Las capacidades descritas se exponen en ROS 2 mediante interfaces estándar de acciones y servicios. Este diseño modular permite una coordinación efectiva a través del árbol de comportamiento detallado en la Sección 3.

3. Núcleo de decisión basado en Prioridades

Si bien el robot dispone de múltiples capacidades, la ejecución aislada de estos módulos no resulta en un comportamiento inteligente. Para operar de forma autónoma en un entorno humano, es imprescindible coordinar de forma efectiva estas habilidades. A tal fin, el núcleo de control se estructura como un árbol de comportamiento (BT) que articula el flujo de ejecución y permite la interrupción ordenada de las tareas. Su implementación se apoya en las bibliotecas *py_trees* Stonier (2026a) y *py_trees_ros* Stonier (2026b).

El intercambio de datos entre los distintos módulos de ROS 2 y los nodos del árbol se gestiona a través de una pizarra de datos o *Blackboard*. A nivel conceptual, este componente opera como un almacenamiento centralizado en formato clave-valor donde se actualiza el contexto de ejecución (p. ej., el nivel de batería, las misiones activas o las detecciones de usuarios). A nivel de funcionamiento, los nodos del árbol registran explícitamente sus permisos de acceso (lectura o escritura) sobre variables específicas del diccionario. Este mecanismo de intermediación evita el paso directo de parámetros entre comportamientos, permitiendo que las distintas ramas evalúen sus condiciones de activación de forma segura y de forma independiente.

Como se muestra en la figura 2, el árbol se apoya en la semántica de los nodos de control tipo *Selector*. Estos nodos evalúan a sus hijos de izquierda a derecha, deteniéndose y devolviendo un estado de *Éxito* o *En Ejecución* en cuanto una de las ramas responde afirmativamente. Aprovechando esta característica dentro de la propagación periódica de la señal de control (*tick*), el árbol principal se ha estructurado a partir de un nodo *Selector* raíz del que cuelgan las cuatro ramas principales ordenadas por prioridad, que se describen a continuación.

3.1. Gestión de energía

La rama situada más a la izquierda del *Selector* raíz, y por tanto la de máxima prioridad, tiene como objetivo preservar la operatividad de la plataforma. En la práctica, este subárbol monitoriza continuamente el estado de carga de la batería y, si el valor desciende por debajo de un umbral predefinido, interrumpe de inmediato cualquier actividad en curso. Al activarse esta condición, el robot emite avisos mediante síntesis de voz para informar a los usuarios de su estado y ejecuta la rutina de retorno a la base de carga (*return to base*). De esta forma, se asegura la recarga del sistema antes de atender a cualquier otra interacción o tarea.

3.2. Interrupción humana

El segundo nivel de prioridad gestiona la reactividad del robot ante la llamada verbal directa de un usuario. Esta rama se activa de forma asíncrona al registrarse el evento de detección de la palabra clave (*wakeword*) emitido por el módulo acústico descrito en la Sección 2. Al validarse este estímulo, la rama toma el control de la ejecución: suspende la misión en curso (si la hubiera), coordina la base y la cabeza en una rutina de *gaze-shifting* con seguimiento facial para establecer contacto visual, e inicia la sesión de interacción conversacional Valencia-Rojas et al. (2026). Cabe destacar que, debido a

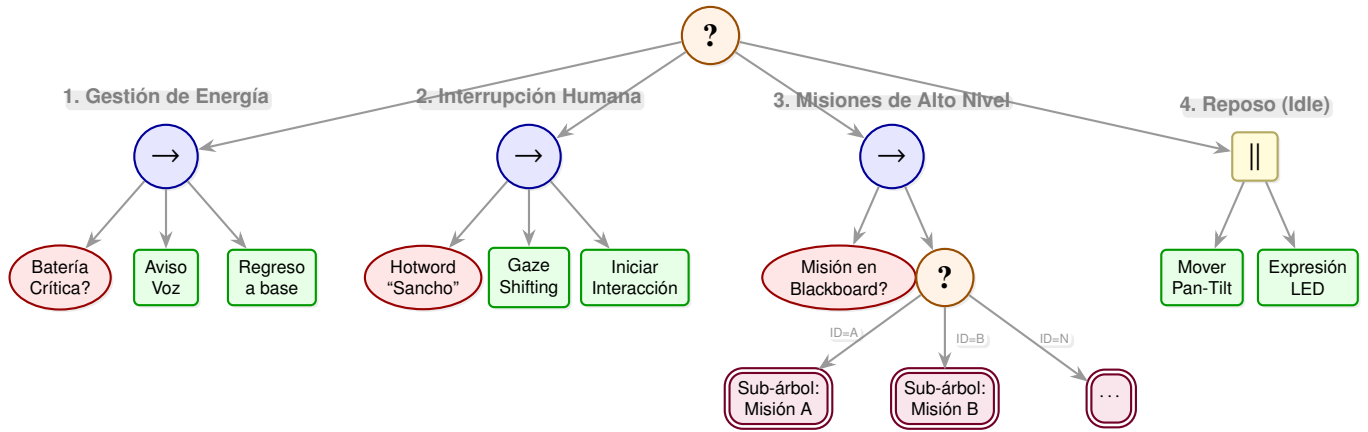


Figura 2: Esquema conceptual del árbol de comportamiento principal. El nodo raíz (Selector) establece una jerarquía de cuatro prioridades evaluadas de izquierda a derecha. La rama de mayor prioridad garantiza la operatividad continua mediante el retorno a la base de carga si el nivel de batería es bajo. El segundo nivel procesa interrupciones reactivas para la interacción humano-robot al detectar la palabra clave de activación. El tercer nivel despliega las *Misiones de Alto Nivel*, delegando la ejecución a subárboles específicos según el contexto almacenado en el *Blackboard*. Finalmente, el estado de *Reposo* actúa como comportamiento por defecto, ejecutando rutinas pasivas concurrentes.

la estructura jerárquica del árbol, este nodo solo puede ser interrumpido por una alerta crítica de batería (Nivel 1). Si esto ocurre, la sesión de interacción se aborta de forma segura y no se reanuda tras el proceso de recarga, asumiendo que el contexto de la conversación habría perdido su validez temporal.

3.3. Misiones de alto nivel

La tercera rama de prioridad gestiona las tareas operativas del robot, como la navegación entre puntos de interés o el guiado de usuarios. Estas misiones se implementan mediante nodos vinculados a servidores de acción (*Action Servers*) de ROS 2. Un aspecto relevante de esta capa dentro del BT es su capacidad para preservar el contexto de ejecución. Si la rama superior (Nivel 2) se activa por una interpelación humana, este subárbol deja de recibir la señal de control (*tick*), lo que detiene temporalmente la navegación o la tarea en curso sin destruir su estado lógico en la secuencia. Una vez finalizada la interacción social, la señal vuelve a propagarse hasta este nivel, lo que permite que el planificador reanude la misión desde el punto en que se pausó.

Por su parte las misiones son gestionadas por un nodo independiente externo al Árbol de Comportamiento. Este nodo organiza las tareas en una cola ordenada por un valor numérico (a menor valor, mayor prioridad). En caso de igualdad, la misión más reciente prevalece. La inserción de nuevas misiones se realiza mediante acciones de ROS 2, mientras que la modificación de la cola (limpiar, extraer o consultar) se gestiona a través de servicios. Finalmente, este nodo es el encargado de publicar la misión seleccionada en el *Blackboard*, lo que permite mantener el Árbol de Comportamiento desacoplado, limitándose únicamente a consultar y ejecutar la tarea resultante junto con los parámetros necesarios (p. ej. coordenadas métricas para una navegación)

3.4. Reposo

La última rama del árbol define el estado de reposo del sistema. Se activa exclusivamente cuando el robot cuenta con

energía suficiente y no existen misiones de alto nivel en cola ni usuarios interactuando. Dado que el robot opera en un espacio de trabajo compartido, se ha optado por evitar la exploración aleatoria (*wandering*) constante para no interferir con la actividad humana ni consumir recursos innecesariamente. En su lugar, esta capa implementa rutinas de presencia social mínima. El robot mantiene la base móvil estática, pero ejecuta movimientos periódicos del cuello (*pan-tilt*) y actualiza sutilmente su expresión. De este modo, comunica visualmente que se encuentra operativo y a la espera de nuevas interacciones, evitando que los usuarios lo perciban como un sistema apagado o fuera de servicio.

4. Validación Experimental

Para validar las toma de decisiones y la capacidad del sistema para gestionar conflictos lógicos sin bloqueos ni pérdida de estado, se ha evaluado cada módulo de forma aislada en un entorno simulado y se ha diseñado un experimento global, que se ha ejecutado repetidas veces, en un entorno físico de laboratorio. La prueba básica consiste en una secuencia de interrupciones que fuerzan al sistema a transitar por todas las capas de prioridad del Árbol de Comportamiento.

En caso del desarrollo de este experimento ha sido documentado en vídeo (ver figura 3) y se encuentra disponible públicamente en el repositorio del proyecto¹. Cabe señalar que, dada la naturaleza dinámica y modular de esta plataforma de investigación, la disposición física del robot mostrada en la grabación corresponde a una configuración de desarrollo (sin el carenado final y con periféricos adicionales, como un brazo manipulador). No obstante, la arquitectura de control lógico y los sensores implicados en la prueba son estrictamente los descritos en la Sección 2.

4.1. Fase 1: Interrupción social y asignación de misión

El experimento comienza con el robot ejecutando una rutina de *roaming* consistente en la navegación autónoma y aleatoria por los distintos nodos del grafo topológico del mapa.

¹Repositorio del proyecto y vídeo demostrativo: https://github.com/MAPIRlab/sancho_robot



Figura 3: Secuencia de eventos del experimento global. (a) El robot opera en la misión base de *roaming*. (b) Un usuario interrumpe la navegación mediante el *wakeword* para asignar una tarea verbalmente. (c) El robot navega hacia el laboratorio indicado en la primera misión. (d) Tras completar la solicitud, un segundo usuario lo interpela y le asigna una nueva misión. (e) Durante este segundo trayecto, se inyecta una alerta de batería crítica, forzando la ejecución de la rama de gestión de energía. (f) El robot aborta la tarea en cuestión y se aproxima a su estación para la recarga asistida. En todos los fotogramas se incluye en la esquina superior izquierda el indicador del nodo activo en el Árbol de Comportamiento.

Nótese que la rutina de *roaming* es una misión de alto nivel (Nivel 3) y no el comportamiento base del robot, el cual es estático.

Durante este trayecto, un usuario interpela al robot pronunciando la palabra clave (“Sancho”). Este estímulo activa la capa de *Interrupción Humana* (Nivel 2), la cual tiene mayor prioridad. Inmediatamente, el BT pausa la meta de navegación del *roaming* sin destruir su estado, detiene la base móvil e inicia el módulo conversacional. El usuario ordena verbalmente: “Ve al laboratorio 2.3.7 y di que vamos a desayunar”. El sistema de diálogo procesa la intención, genera los parámetros (destino y mensaje) y encola una nueva misión de alto nivel (*move&speak*) mediante una acción de ROS 2, insertando los datos en el *Blackboard*. Una vez que el módulo conversacional determina el fin de la interacción, la rama finaliza con estado de *Éxito*.

4.2. Fase 2: Ejecución de la nueva misión

Al finalizar la interrupción, la señal de control (*tick*) vuelve a propagarse por el árbol. La nueva misión encolada (*move&speak*), al compartir el mismo nivel de prioridad numérica que la tarea previa de *roaming*, asume el control por haber sido insertada más recientemente. El robot reanuda la marcha, navega hacia el lugar indicado y, mediante la síntesis de voz, entrega el mensaje, completando la misión de forma satisfactoria.

4.3. Fase 3: Preservación de energía y recarga

En este momento se produce una segunda interrupción social. Otro usuario detiene al robot con el *wakeword* y le ordena una nueva misión similar, pero dirigida a una sala diferente. El robot atiende la petición y comienza la navegación hacia el nuevo destino.

A mitad de camino, se inyecta por *software* una señal simulada de batería crítica (caída drástica por debajo del umbral de seguridad predefinido). Esta señal activa de manera inmediata la rama de *Gestión de Energía* (Nivel 1), que representa la prioridad máxima del árbol. El BT resuelve la colisión priorizando la navegación hasta la base de carga y emitiendo un aviso verbal informando de la situación.

El experimento concluye satisfactoriamente cuando el robot alcanza su estación. La ejecución de este guion, ejecutado en repetidas ocasiones con algunas variantes, demuestra que la arquitectura jerárquica propuesta no solo coordina eficientemente un amplio volumen de habilidades subyacentes, sino que toma decisiones coherentes y seguras ante la concurrencia de eventos en un entorno dinámico real.

5. Conclusiones y Trabajo Futuro

Este trabajo presenta un sistema de toma de decisiones jerárquico para la plataforma robótica *Sancho*, basado en un Árbol de Comportamiento (BT) estructurado en cuatro capas de prioridad. Mediante su implementación en ROS 2 y el uso de una memoria compartida (*Blackboard*), la arquitectura propuesta logra independizar de manera efectiva los módulos de navegación, percepción e interacción del núcleo de control lógico. La evaluación experimental confirma que este diseño resuelve la concurrencia de eventos sin bloqueos, el robot es capaz de interrumpir la navegación base para atender comandos verbales, integrar de forma fluida nuevas misiones encoladas, y abortar cualquier tarea ante una alerta de energía para asegurar su recarga.

Como trabajo futuro, la investigación plantea tres líneas principales de desarrollo. En primer lugar, a nivel de validación, se contempla el despliegue del robot en modo de opera-

ción continua en entornos cotidianos y desestructurados (*in-the-wild*). Este paso es fundamental para evaluar la robustez del árbol de comportamiento frente a situaciones impredecibles que no surgen en el laboratorio, tales como interacciones simultáneas de múltiples usuarios, ruido ambiental que afecte al módulo acústico, o la gestión prolongada de la batería durante jornadas completas.

En segundo lugar, desde el plano del rendimiento técnico, los esfuerzos se centrarán en la optimización de la latencia global del sistema. Dado que la fluidez en la Interacción Humano-Robot (HRI) es crítica para la aceptación por parte del usuario, se buscará mitigar los cuellos de botella inherentes al ciclo de comunicación multimodal (reconocimiento de voz, procesamiento del LLM y síntesis acústica). Esto requerirá refinar la gestión asíncrona entre los servidores de acción locales y los nodos externos de computación en el borde (*edge computing*).

Finalmente, a nivel de comportamiento, se investigará la evolución del control lógico hacia un modelo de autonomía proactiva. Actualmente, el comportamiento de *Sancho* es estrictamente reactivo ante las llamadas directas (*wakeword*) o la inserción externa de tareas. El objetivo a medio plazo es dotar al sistema de la capacidad de interpretar de forma continua su contexto espacial y social para generar misiones por iniciativa propia (p. ej., acercarse a un usuario que parece buscar información), pasando de ser una herramienta de ejecución de comandos a un agente social proactivo.

Agradecimientos

Este trabajo ha sido desarrollado en el contexto de los proyectos MINDMAPS (PID2023-148191NB-I00), financiado por el Ministerio de Ciencia e Innovación, y por el Plan Andaluz de Investigación, Desarrollo e Innovación (PPRO-TEP960-G-2023).

Referencias

- Ambrosio-Cestero, Gregorio, Andrades, G., Cipriano, Gonzalez-Jimenez, Javier, Ruiz-Sarmiento, Jose-Raul, 2026. Csar: Containerized system architecture for robotics.
URL: <https://arxiv.org/abs/2606.30293>
- Cañete, A., Quemada-Torres, E., Ruiz-Sarmiento, J.-R., Moreno, F. Á., Gonzalez-Jimenez, J., 2024. Sistema multimodal para la orientación de robots móviles hacia su interlocutor. Jornadas de Automática 45.
URL: <https://doi.org/10.17979/ja-cea.2024.45.10939>
DOI: 10.17979/ja-cea.2024.45.10939
- Cañete, A., Jun. 2025. Sistema de detección de usuarios y comunicación proactiva para un robot autónomo de servicios. Trabajo fin de grado, Universidad de Málaga.
URL: <http://hdl.handle.net/10630/40821>
- Cañete Baena, A., Galindo Andrades, C., Moreno Dueñas, F. A., Gonzalez Jimenez, A. J., 2025. Sistema de detección de usuarios y comunicación proactiva para un robot autónomo de servicio 1 (1), Robótica.
URL: <http://138.100.76.10/index.php/SimposiosCEA/article/view/96>
DOI: 10.64117/simposioscea.v1i1.46
- Macenski, S., Moore, T., Lu, D., Merzlyakov, A., Ferguson, M., 2023. From the desks of ros maintainers: A survey of modern & capable mobile robotics algorithms in the robot operating system 2. Robotics and Autonomous Systems.
- Moreno, F.-A., Monroy, J., Ruiz-Sarmiento, J.-R., Galindo, C., Gonzalez-Jimenez, J., 2020. Automatic waypoint generation to improve robot navigation through narrow spaces. Sensors 20 (1).
URL: <https://www.mdpi.com/1424-8220/20/1/240>
DOI: 10.3390/s20010240
- Möller, R., Furnari, A., Battiato, S., Härmä, A., Farinella, G. M., 2021. A survey on human-aware robot navigation.
URL: <https://arxiv.org/abs/2106.11650>
- Quemada-Torres, E., Jun. 2025. Sistema multimodal de interacción humano robot basado en técnicas de ia. Trabajo fin de grado, Universidad de Málaga.
URL: <https://hdl.handle.net/10630/40843>
- Scripka, D., 2026. openwakeword: An open-source audio wake word detection framework. <https://github.com/dscripka/openWakeWord>, repositorio de GitHub.
- Stonier, D., 2026a. py_trees: a python implementation of behaviour trees. https://github.com/splintered-reality/py_trees.
- Stonier, D., 2026b. py_trees_ros: Behaviours, trees, and utilities that extend pytrees for use with ros. https://github.com/splintered-reality/py_trees_ros.
- TensorFlow, 2024. Movenet: Ultra fast and accurate pose detection model. Último acceso: 2025-05-02.
URL: <https://www.tensorflow.org/hub/tutorials/movenet>
- Valencia-Rojas, A., Hormigo-Jiménez, P., Quemada-Torres, E., Moreno, F. A., Gonzalez-Jiménez, J., 2026. Multimodal attention and robust interlocutor tracking in natural hri. In: Jornadas de Automática. Vol. 47.
URL: <https://doi.org/>
- Ögren, P., Sprague, C. I., May 2022. Behavior trees in robot control systems. Annual Review of Control, Robotics, and Autonomous Systems 5 (1), 81–107.
URL: <http://dx.doi.org/10.1146/annurev-control-042920-095314>
DOI: 10.1146/annurev-control-042920-095314