

## Coordinación de agentes de IA para el control de un robot social

Merino-Fidalgo, S.<sup>a,\*</sup>, Zalama, E.<sup>a,b</sup>, Gómez-García-Bermejo, J.<sup>a,b</sup>, Duque-Domingo, J.<sup>a</sup>

<sup>a</sup>ITAP-DISA, Universidad de Valladolid, C/Doctor Mergelina, 47011, Valladolid, España.

<sup>b</sup>CARTIF Centro Tecnológico, 47151, Valladolid, España.

### Resumen

En este trabajo presentamos una arquitectura basada en agentes para la ejecución reactiva de tareas robóticas multi-paso a partir de instrucciones en lenguaje natural. El sistema traduce la instrucción en una secuencia dinámica de acciones gestionadas por un agente director que descompone la tarea, selecciona la siguiente acción en tiempo de ejecución y delega su ejecución a agentes especializados (por ejemplo, navegación o comunicación). A diferencia de enfoques basados en Árboles de Comportamiento, la arquitectura propuesta no genera estructuras de control precompiladas, sino que opera en un bucle cerrado donde el director monitoriza el feedback del robot y replanifica ante fallos o contingencias. La validación se realiza en un entorno real con un robot físico, demostrando la capacidad del sistema para adaptarse a perturbaciones durante la ejecución. Los resultados muestran que la coordinación multi-agente permite una ejecución flexible y robusta con menor esfuerzo de ingeniería manual.

*Palabras clave:* Planificación y ejecución, Redes de robots y sensores inteligentes, Robots móviles, Aspectos cognitivos de los sistemas de automatización y los seres humanos, Modelos de lenguaje de gran tamaño.

### Coordination of AI agents for the control of a social robot

#### Abstract

This paper presents an agent-based architecture for the reactive execution of multi-step robotic tasks from natural language instructions. The system translates a user instruction into a dynamic sequence of actions managed by a director agent, which decomposes the task, selects the next action at runtime, and delegates execution to specialized Action agents (e.g., navigation or speech). Unlike approaches based on Behavior Trees, the proposed architecture does not generate precompiled control structures, but instead operates in a closed-loop manner in which the director continuously monitors robot feedback and replans when failures or contingencies occur. The framework is validated in a real-world environment using a physical robot, demonstrating its ability to adapt to disturbances during execution. Results indicate that multi-agent coordination enables flexible and robust task execution while reducing the manual engineering effort typically associated with structured control representations.

*Keywords:* Planning and Execution, Networks of robots and intelligent sensors, Mobile robots, Cognitive aspects of automation systems and humans, Large language models.

## 1. Introducción

La ejecución de tareas multi-paso a partir de instrucciones en lenguaje natural constituye uno de los objetivos centrales de la robótica de servicio y la robótica social. La posibilidad de interactuar con un robot mediante comandos de alto nivel, expresados en lenguaje natural, reduce la barrera tecnológica para el usuario final y facilita su despliegue en entornos cotidianos como hogares, hospitales, oficinas o espacios públicos (Petrich et al., 2022). Sin embargo, traducir una instrucción lingüística

en una secuencia de acciones robóticas fiables sigue siendo un problema abierto, debido a la ambigüedad del lenguaje, la incertidumbre perceptiva y la necesidad de adaptación durante la ejecución (Liu and Zhang, 2019).

Tradicionalmente, la comunidad ha abordado este problema mediante arquitecturas deliberativas combinadas con representaciones estructuradas de control. En particular, los Árboles de Comportamiento o Behavior Trees (BTs) Colledanchise and Ögren (2017) se han consolidado como una alternativa robusta y modular frente a las máquinas de estados finitos jerárquicas

(Brand and Zafiropulo, 1983). Su adopción en sistemas reales, incluyendo plataformas basadas en ROS (Koubaa et al., 2017), ha demostrado su utilidad para organizar tareas complejas mediante la composición jerárquica de comportamientos Macenski et al. (2020). No obstante, aunque los BTs proporcionan claridad estructural y mecanismos explícitos de recuperación ante fallos, su diseño suele requerir una definición previa de la estructura de control y un esfuerzo considerable de ingeniería. Además, cuando las tareas se derivan dinámicamente de instrucciones abiertas en lenguaje natural, la generación y mantenimiento de estas estructuras puede resultar compleja.

Más recientemente, el auge de los modelos de lenguaje de gran escala (LLMs) (Chang et al., 2024) ha impulsado nuevas estrategias para la descomposición y planificación de tareas a partir de texto libre. Sin embargo, muchos de estos trabajos se centran en entornos simulados o en configuraciones altamente controladas, y no siempre abordan de forma explícita la coordinación continua y reactiva necesaria en entornos físicos reales (Wang et al., 2025).

En este contexto, proponemos una arquitectura basada en coordinación multi-agente en bucle cerrado para la ejecución de tareas multi-paso derivadas de instrucciones en lenguaje natural. A diferencia de los enfoques basados en estructuras de control precompiladas, el sistema no genera un Árbol de Comportamiento explícito, sino que emplea un agente director que descompone dinámicamente la tarea, selecciona la siguiente acción en tiempo de ejecución y delega su realización en agentes especializados (por ejemplo, navegación o comunicación). El director monitoriza de forma continua el feedback proporcionado por el robot y adapta la secuencia de acciones ante fallos o contingencias, permitiendo una replanificación local sin necesidad de reconstruir una estructura global de control.

La propuesta se valida en un entorno real utilizando un robot físico, lo que permite evaluar el comportamiento del sistema bajo condiciones de incertidumbre propias del mundo real. Los resultados muestran que la coordinación multi-agente facilita una ejecución flexible y robusta de tareas multi-paso, reduciendo al mismo tiempo el esfuerzo de ingeniería manual típicamente asociado al diseño y mantenimiento de estructuras de control estáticas. De este modo, el trabajo contribuye a avanzar hacia arquitecturas más adaptativas y escalables para la interacción natural entre humanos y robots.

El resto del artículo se organiza de la siguiente manera: En la Sección 2 se realiza una revisión del marco teórico y de los trabajos relacionados. La Sección 3 detalla la metodología propuesta, incluyendo el proceso de los agentes para realizar las acciones. La Sección 4 presenta los resultados experimentales y un ejemplo de una acción. La Sección 5 discute las contribuciones y ventajas del sistema, así como las áreas de mejora, y la Sección 6 completa con las conclusiones y el trabajo futuro.

## 2. Marco teórico y trabajos relacionados

En Inteligencia Artificial, un agente se define, de forma general, como una entidad que percibe su entorno y actúa sobre él para alcanzar objetivos, seleccionando acciones en función de su historia de percepciones y de un criterio de rendimiento (Kolt, 2025). Desde esta perspectiva, la investigación clásica en

agentes ha abarcado arquitecturas reactivas (respuesta estímulo–acción), deliberativas (planificación explícita) e híbridas. La definición de agentes debe considerar propiedades como autonomía, reactividad, proactividad y sociabilidad para adecuarse a la tarea que va a realizar (Kapoor et al., 2024).

En paralelo, la investigación en sistemas multi-agente estudia cómo múltiples agentes especializados pueden coordinarse para resolver objetivos complejos (p. ej., asignación de tareas, negociación, coordinación y reparto de recursos) (Amirkhani and Barshooi, 2022). Este marco ha sido tradicionalmente valioso cuando una tarea puede descomponerse en subtareas heterogéneas o cuando se requiere modularidad y escalabilidad.

En los últimos años, el término agentes de IA se ha popularizado para describir sistemas que combinan modelos de lenguaje de gran escala (LLMs) con bucles de razonamiento–acción, memoria y uso de herramientas externas (Joshi, 2025). Un patrón representativo es ReAct, que intercala razonamiento en lenguaje natural con acciones (p. ej., consultas a herramientas o entornos) para reducir alucinaciones y mejorar la trazabilidad de decisiones (Yao et al., 2022). No obstante, trasladar estos enfoques a robótica real introduce retos adicionales: (i) seguridad y restricciones físicas, (ii) percepción ruidosa y parcial, (iii) latencias y fallos de ejecución, y (iv) necesidad de mecanismos de recuperación robustos (Barjaková et al., 2023).

En robótica, una línea activa busca convertir instrucciones en lenguaje natural en planes o secuencias de habilidades ejecutables. SayCan combina un LLM con una función de “viabilidad/afordancia” para seleccionar, entre un conjunto de habilidades predefinidas, la siguiente acción más probable y ejecutable, mostrando resultados en tareas de manipulación y ejecución de instrucciones de alto nivel (Yamao et al., 2024). Complementariamente, Inner Monologue propone realimentar al modelo con descripciones y señales del entorno expresadas en lenguaje, mejorando la planificación y la corrección ante fallos mediante un bucle de retroalimentación textual (Huang et al., 2022). Por otro lado, Code as Policies (y el marco de “Language Model Programs”) utiliza modelos de código para sintetizar programas que invocan APIs de percepción y control, permitiendo bucles reactivos (condicionales, iteraciones) y cierta capacidad de razonamiento geométrico (Liang et al., 2022).

Una dirección distinta integra percepción y control de forma más end-to-end. PaLM-E propone un modelo multimodal “embodied” que incorpora señales visuales y de estado para resolver tareas robóticas y de razonamiento con transferencia entre dominios (Driess et al., 2023). RT-2 explora modelos visión–lenguaje–acción (VLA) que transfieren conocimiento preentrenado en datos web hacia control robótico, mostrando mejoras de generalización ante instrucciones y objetos novedosos (Zitkovich et al., 2023). Aunque estas aproximaciones amplían la cobertura semántica, suelen sacrificar parte de la interpretabilidad y requieren consideraciones adicionales para garantizar seguridad, control fino y recuperación ante fallos.

Nuestro enfoque se sitúa dentro de la tendencia del uso de agentes aplicados a robótica, enfatizando la coordinación en bucle cerrado y la delegación a agentes especializados para ejecutar tareas multi-paso desde lenguaje natural, sin imponer una estructura de control precompilada. En comparación con estrategias centradas en seleccionar habilidades por puntuación o en sintetizar programas completos, proponemos un ciclo explícito

de selección de la siguiente acción con monitorización y replanificación local validado en un robot real.

### 3. Metodología propuesta

La metodología propuesta se basa en una arquitectura multi-agente en bucle cerrado para la ejecución de tareas robóticas multi-paso a partir de instrucciones en lenguaje natural. El sistema recibe una instrucción verbal, la transforma en texto, la procesa mediante un agente central denominado Director, y ejecuta las acciones correspondientes a través de agentes especializados.

A diferencia de enfoques anteriores, el sistema no está basado en estructuras de control predefinidas (Merino-Fidalgo et al., 2025), sino que selecciona dinámicamente la siguiente acción en tiempo de ejecución, adaptándose al estado actual del robot y al feedback recibido.

El proceso completo puede dividirse en tres fases: (i) adquisición y transcripción de la instrucción, (ii) descomposición y planificación dinámica por parte del Director, y (iii) delegación y ejecución reactiva de acciones mediante agentes especializados.

#### 3.1. Arquitectura multi-agente

La arquitectura general del sistema se muestra en la Figura 1. En ella se representa el flujo completo desde la emisión de la instrucción por parte del usuario hasta la ejecución física en el robot.

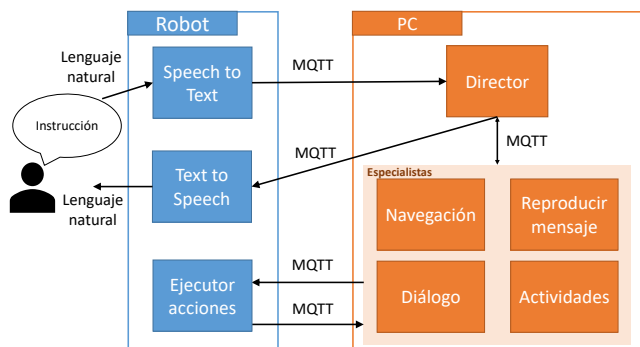


Figura 1: Representación gráfica del método propuesto. El marco azul muestra la parte del sistema ejecutada por el robot social, mientras que el naranja representa los agentes del ordenador.

El proceso comienza cuando el usuario proporciona una instrucción verbal al robot. Esta señal es enviada a un módulo de Speech-to-Text ejecutado en un ordenador, donde se convierte en texto en lenguaje natural. La instrucción transcrita se transmite entonces al agente Director a través de un sistema de mensajería basado en MQTT (Message Queuing Telemetry Transport) (Mishra and Kertesz, 2020), que actúa como plataforma de comunicación entre los distintos módulos.

El Director constituye el núcleo de la arquitectura. Recibe la instrucción en lenguaje natural, la analiza y la descompone en una secuencia dinámica de acciones. En lugar de generar una estructura de control estática, el Director decide en tiempo de ejecución qué acción debe ejecutarse a continuación, en función del progreso de la tarea y del feedback recibido. Además, si la

ejecución falla o sucede algún problema, el Director emite un mensaje en lenguaje natural a través del módulo Text-to-Speech para avisar al usuario de lo sucedido.

Para llevar a cabo la ejecución, el Director delega las acciones en distintos Agentes Especialistas, también comunicados mediante MQTT. Entre ellos se incluyen:

- Navegación, encargado del desplazamiento físico del robot.
- Reproducir mensaje, orientado a la emisión de mensajes.
- Diálogo, responsable de la interacción verbal, gestionando las respuestas del usuario.
- Otros módulos de actividades especializadas.

Cada agente ejecutor informa al Director del resultado de la acción (éxito, fallo u otro estado relevante). Esta retroalimentación permite mantener un bucle cerrado de control. Esta arquitectura distribuida desacopla claramente la toma de decisiones (Director) de la ejecución física (Agentes Especialistas), favoreciendo la modularidad, la escalabilidad y la reutilización de habilidades.

#### 3.2. Funcionamiento del Director

El Director constituye el núcleo decisional del sistema y es responsable de la coordinación dinámica de la ejecución. Su funcionamiento se basa en un modelo de control iterativo en bucle cerrado, en el que cada decisión se toma en función del estado actualizado de la tarea y del feedback proporcionado por los agentes ejecutores.

Tras recibir la instrucción en lenguaje natural, el Director realiza un análisis inicial orientado a identificar la intención global y a generar una descomposición preliminar en acciones atómicas o subtareas. Esta descomposición no se materializa en una estructura de control estática, sino que actúa como una guía inicial susceptible de modificación durante la ejecución.

A continuación, el Director selecciona la siguiente acción más adecuada en función del estado actual del sistema y la delega al agente especialista correspondiente. Una vez ejecutada la acción, el agente devuelve un mensaje de retroalimentación indicando el resultado obtenido. El Director actualiza entonces su estado interno y determina el siguiente paso a ejecutar. Este proceso se repite de manera iterativa hasta completar la tarea o hasta detectar una condición que impida su continuación.

En caso de que durante la ejecución se determine que la instrucción no puede completarse —ya sea por fallos reiterados, imposibilidad de alcanzar un objetivo o condiciones externas no previstas— el Director finaliza la secuencia de manera controlada y activa el módulo de comunicación para reproducir un mensaje informando al usuario de la situación. De este modo, el sistema mantiene coherencia comunicativa y evita comportamientos indefinidos o silencios prolongados ante errores irreversibles.

La replanificación se realiza de forma incremental y localizada, evitando la reconstrucción completa de una estructura global de control. Este enfoque favorece la adaptabilidad sin introducir la complejidad asociada a representaciones jerárquicas rígidas.

El Algoritmo 1 resume el ciclo completo de decisión del Director, incluyendo la descomposición inicial, la ejecución iterativa y la gestión de contingencias.

---

**Algoritmo 1:** Bucle de ejecución del Director

---

**Entrada:** Instrucción en lenguaje natural  $I$

**Salida:** Ejecución de la tarea o cancelación controlada

```

1  $plan \leftarrow$  DESCOMPONER( $I$ );
2  $estado \leftarrow$  INICIALIZARESTADO();
3 while no FINALIZADO( $plan, estado$ ) do
4    $a \leftarrow$  SELECCIONARSIGUIENTEACCION( $plan, estado$ );
5    $agente \leftarrow$  ASIGNARAGENTE( $a$ );
6    $r \leftarrow$  EJECUTAR( $agente, a$ );
7   ACTUALIZARESTADO( $estado, r$ );
8   if FALLO( $r$ ) then
9      $plan \leftarrow$ 
10      RECUPERAROREPLANIFICAR( $plan, estado$ );
11     if NORECUPERABLE( $estado$ ) then
12       NOTIFICARUSUARIO(‘‘No puedo
        completar la instrucción’’);
        break;
```

---

### 3.3. Gestión de fallos y adaptación

La robustez del sistema se fundamenta en un mecanismo explícito de gestión de contingencias integrado en el ciclo de decisión del Director. Durante la ejecución pueden producirse distintos tipos de eventos inesperados, tales como fallos de navegación, errores en la ejecución de actividades específicas, problemas de comunicación entre módulos o situaciones de timeout derivadas de retrasos en la respuesta de algún agente.

Ante cualquiera de estas circunstancias, el Director evalúa el contexto y aplica una estrategia de recuperación adecuada. Dependiendo del tipo de incidente y del estado de la tarea, el sistema puede optar por reintentar la acción, modificar el orden previsto de ejecución, seleccionar una alternativa disponible o, en caso necesario, solicitar información adicional al usuario. Cuando ninguna estrategia permite restablecer una ejecución viable, el sistema interrumpe la tarea de forma controlada e informa explícitamente al usuario mediante un mensaje reproducido por el robot.

Este mecanismo de adaptación continua permite mantener un comportamiento robusto en entornos reales caracterizados por incertidumbre y variabilidad. A diferencia de enfoques basados en estructuras jerárquicas predefinidas, donde todos los caminos posibles deben anticiparse durante el diseño, la arquitectura propuesta permite gestionar situaciones no previstas mediante decisiones locales tomadas en tiempo de ejecución, reduciendo así el esfuerzo de ingeniería manual y aumentando la flexibilidad del sistema.

### 3.4. Agentes especializados

La arquitectura incorpora un conjunto de agentes especializados encargados de ejecutar las acciones delegadas por el Director. Estos agentes encapsulan las capacidades funcionales del robot —como navegación, interacción verbal o ejecución de actividades específicas— y actúan como una capa de abstracción entre la lógica de coordinación y la plataforma robótica.

Todos los agentes comparten una interfaz común de comunicación, lo que permite desacoplar la toma de decisiones de los detalles de implementación. Esta separación favorece la modularidad y facilita tanto la incorporación de nuevas habilidades como la sustitución de módulos existentes sin modificar la lógica del Director.

El diseño es independiente de la plataforma concreta: los agentes pueden adaptarse a distintos robots siempre que exista un módulo que traduzca las órdenes abstractas en comandos específicos del sistema de control. De este modo, la arquitectura resulta escalable y reutilizable, permitiendo ampliar progresivamente el repertorio de acciones disponibles sin alterar la estructura global del sistema.

## 4. Experimentos y resultados

La validación del sistema se llevó a cabo en un entorno real utilizando un robot físico Temi (Temi, 2024), equipado con capacidades de navegación autónoma e interacción verbal. La arquitectura descrita en la sección anterior fue desplegada en un entorno distribuido, donde los distintos módulos (Director, agentes especialistas y componentes de voz) se comunicaban mediante el protocolo MQTT. Los agentes utilizan GPT 4o mini como LLM de razonamiento.

El entorno de pruebas consistió en un espacio interior estructurado con varias localizaciones diferenciadas (por ejemplo, salón, pasillo o cocina), permitiendo evaluar tareas que implican desplazamiento, comunicación y ejecución secuencial de acciones. Las pruebas se realizaron bajo condiciones reales de operación, incluyendo posibles imprecisiones en la localización y variaciones temporales en la ejecución de acciones.

Se llevó a cabo un estudio con 25 usuarios sin conocimientos previos del software ni de la arquitectura interna del robot. Antes de comenzar, se les proporcionó una breve explicación sobre las capacidades disponibles del robot, incluyendo navegación entre ubicaciones predefinidas e interacción verbal básica. A cada participante se le pidió que formulara, libremente y con sus propias palabras, una instrucción de cada uno de los cuatro tipos establecidos. Podían expresarlas como quisieran en lenguaje natural, incluyendo las acciones, el vocabulario y la forma de expresarlas que consideraran oportunos, siempre que respetaran las características de cada tipo. En primer lugar, debían proponer una tarea simple compuesta por dos acciones encadenadas, por ejemplo: ‘‘Di hola y luego ve al baño’’. En segundo lugar, una tarea más compleja, formada por más de dos acciones e incluyendo al menos una condición lógica, por ejemplo: ‘‘Ve al salón y pregunta a Laura si está viendo la televisión; si responde que sí, ven al dormitorio y dímelo; de lo contrario, dile que venga’’. En tercer lugar, debían formular una instrucción errónea que incluyera argumentos no válidos dentro del dominio definido, como ubicaciones inexistentes o contactos no registrados. Finalmente, tenían que proponer una tarea imposible, incorporando acciones que el robot no es capaz de realizar, como coger objetos físicos o grabar un vídeo.

Para evaluar el comportamiento del sistema, se definieron métricas orientadas tanto a la robustez como a la capacidad de adaptación ante diferentes tipos de instrucciones. La métrica principal fue la tasa de éxito, definida como el porcentaje de instrucciones completadas correctamente en relación con el total

Tabla 1: Resultados experimentales obtenidos en el estudio con 25 usuarios.

Tipo de tarea	Tasa de éxito (%)	Detección correcta (%)	Replanificaciones (media)
Simple	94	–	0.3
Compleja	86	–	1.2
Errónea	-	90	0.8
Imposible	-	93	0.2

de instrucciones válidas recibidas. Se consideró que una tarea era exitosa cuando todas las acciones previstas se ejecutaban en el orden adecuado y el sistema finalizaba sin intervención externa. En el caso de instrucciones erróneas o imposibles, se evaluó la tasa de detección correcta, es decir, el porcentaje de casos en los que el Director identificó la imposibilidad de completar la instrucción y notificó adecuadamente al usuario mediante un mensaje verbal. Adicionalmente, se midió el número medio de replanificaciones locales por tarea, entendido como el número de veces que el Director modificó la secuencia de ejecución tras recibir feedback de los agentes especialistas. Para añadir realismo a los experimentos, por cada tipo de instrucción se introdujeron 5 situaciones inesperadas durante la ejecución, como cerrar una puerta o bloquear el camino con un obstáculo o no responder a una pregunta realizada por el robot.

Los resultados obtenidos, recogidos en la Tabla 1, muestran un comportamiento robusto del sistema en los distintos tipos de tareas evaluadas. En las tareas simples, compuestas por dos acciones encadenadas, se alcanzó una tasa de éxito del 94 %, con un número medio de replanificaciones bajo (0.3), lo que indica que la arquitectura responde de forma estable cuando la estructura de la instrucción es directa y poco ambigua.

En las tareas complejas, que incorporaban múltiples acciones y condiciones, la tasa de éxito descendió al 86 %, incrementándose el número medio de replanificaciones a 1.2 por tarea. Este aumento refleja la mayor dificultad asociada a la gestión de dependencias condicionales y a la coordinación dinámica de secuencias más largas.

En el caso de las tareas erróneas, que incluían argumentos no válidos dentro del dominio del sistema, se obtuvo una detección correcta del 90 %. Estos resultados indican que el Director fue capaz, en la mayoría de los casos, de identificar inconsistencias y notificar adecuadamente al usuario. De forma similar, en las tareas imposibles —aquellas que requerían capacidades no disponibles en el robot— la tasa de detección correcta alcanzó el 93 %, con un número reducido de replanificaciones.

Los errores observados pueden agruparse en tres categorías principales. En primer lugar, se detectaron fallos asociados a instrucciones excesivamente ambiguas, cuya formulación dificultaba la correcta interpretación por parte del Director. En segundo lugar, se identificaron problemas derivados de múltiples intentos fallidos durante la replanificación, especialmente en escenarios donde una acción no alcanzaba el estado esperado tras varios reintentos consecutivos. Finalmente, y de manera más relevante, algunos errores estuvieron relacionados con la validación implícita de plausibilidad de las acciones: en ausencia de un mecanismo formal de verificación previa, el Director intentó ejecutar instrucciones que el sistema no podía realizar, lo que condujo a fallos posteriores en la cadena de ejecución.

Estos resultados evidencian que la arquitectura basada en coordinación dinámica ofrece un comportamiento sólido en la

mayoría de los escenarios evaluados. La elevada tasa de éxito en tareas simples y el rendimiento consistente en tareas complejas indican que el enfoque permite gestionar secuencias de acciones de forma estable incluso en presencia de incertidumbre. La principal diferencia con el estado del arte reside en que, en caso de que se produzca un error, el Director se encarga de buscar una solución en el momento, mientras que en otros sistemas requieren una replanificación completa o parcial a partir del error, lo que se traduce en más tiempo de espera hasta retomar la ejecución.

Además, desde el punto de vista de la interacción humano-robot, los participantes del estudio manifestaron que el uso del sistema resultaba sencillo e intuitivo. Tras una breve explicación inicial de las capacidades disponibles, los usuarios pudieron formular instrucciones de manera natural, sin necesidad de aprender una sintaxis específica ni estructuras rígidas de comando. Esta percepción positiva sugiere que la arquitectura no solo aporta robustez técnica en la ejecución, sino también un valor añadido en términos de usabilidad y accesibilidad, aspectos clave para la adopción de sistemas robóticos en entornos reales.

En conjunto, estos resultados refuerzan la idea de que la coordinación basada en agentes en bucle cerrado constituye una aproximación prometedora tanto desde el punto de vista técnico como desde la experiencia de usuario.

## 5. Discusión

Los resultados obtenidos ponen de manifiesto que la arquitectura basada en un Director con coordinación dinámica en bucle cerrado constituye una alternativa viable a enfoques basados en estructuras de control predefinidas para la ejecución de tareas multi-paso a partir de lenguaje natural. La elevada tasa de éxito en tareas simples y el rendimiento sólido en tareas complejas indican que la descomposición dinámica y la delegación a agentes especializados permiten mantener coherencia en la ejecución incluso cuando intervienen múltiples acciones encadenadas.

El incremento en el número medio de replanificaciones en tareas complejas refleja el carácter inherentemente incierto de los entornos reales y la necesidad de mecanismos de adaptación continua. Lejos de interpretarse como una debilidad, este comportamiento evidencia que el sistema es capaz de detectar desviaciones respecto al estado esperado y ajustar la secuencia de acciones sin reiniciar la tarea completa, lo que supone una ventaja frente a estructuras rígidas donde todos los caminos alternativos deben estar predefinidos.

No obstante, el análisis cualitativo de los errores revela áreas de mejora. En primer lugar, la ambigüedad lingüística continúa siendo un factor crítico: cuando la instrucción no es específica con claridad el objetivo o introduce referencias implícitas,

el Director puede generar una descomposición inadecuada. En segundo lugar, la estrategia de replanificación basada en intentos sucesivos puede resultar insuficiente en situaciones donde la causa del fallo no es transitoria, sino estructural. Finalmente, la ausencia de un mecanismo explícito de validación de plausibilidad antes de la ejecución provoca que, en determinados casos, el Director intente realizar acciones que el sistema no puede llevar a cabo, detectándose la imposibilidad únicamente durante la fase de ejecución.

En conjunto, la discusión confirma que la arquitectura propuesta ofrece un equilibrio adecuado entre flexibilidad y robustez para escenarios reales, al tiempo que identifica líneas claras de mejora orientadas a reforzar la fiabilidad ante instrucciones ambiguas o físicamente no ejecutables.

## 6. Conclusiones y trabajo futuro

En este trabajo se ha presentado una arquitectura basada en coordinación multi-agente en bucle cerrado para la ejecución de tareas robóticas multi-paso a partir de instrucciones en lenguaje natural. El sistema se articula en torno a un agente central, denominado Director, responsable de la descomposición dinámica de la tarea, la delegación a agentes especializados y la gestión de contingencias durante la ejecución. A diferencia de enfoques basados en estructuras de control predefinidas, la propuesta evita la generación de representaciones estáticas, apostando por una toma de decisiones incremental en tiempo de ejecución.

La validación experimental con 25 usuarios en un entorno real ha demostrado que la arquitectura permite ejecutar correctamente la mayoría de las tareas simples y complejas, así como detectar en un alto porcentaje las instrucciones erróneas o imposibles. Los resultados evidencian que la coordinación dinámica proporciona un nivel adecuado de flexibilidad y robustez ante perturbaciones, reduciendo el esfuerzo de ingeniería manual asociado a la definición explícita de todos los posibles caminos de ejecución.

No obstante, el análisis de los errores identificados revela limitaciones que abren líneas claras de mejora. En particular, la ambigüedad lingüística y la ausencia de mecanismos formales de validación previa pueden conducir a intentos de ejecución de acciones no plausibles o no soportadas por el sistema. Asimismo, la estrategia actual de replanificación podría enriquecerse mediante mecanismos más sofisticados de razonamiento sobre fallos persistentes.

Como trabajo futuro, se plantea la incorporación de modelos explícitos de capacidades y restricciones que permitan validar la factibilidad de una instrucción antes de su ejecución, así como la integración de mecanismos avanzados de desambiguación interactiva con el usuario. Estas extensiones permitirían reforzar la fiabilidad del sistema y avanzar hacia arquitecturas más completas para la ejecución segura y transparente de tareas robóticas derivadas de lenguaje natural en entornos reales.

## Agradecimientos

La investigación que se presenta en este trabajo ha recibido financiación del proyecto SOROCARE PID2024-157671OB-I00 financiado por MICIU/AEI/10.13039/501100011033/ FEDER, UE.

## Referencias

- Amirkhani, A., Barshooi, A. H., 2022. Consensus in multi-agent systems: a review. *Artificial Intelligence Review* 55 (5), 3897–3935.
- Barjaková, M., Garneró, A., d'Hombres, B., 2023. Risk factors for loneliness: A literature review. *Social Science & Medicine* 334, 116163.  
URL: <https://www.sciencedirect.com/science/article/pii/S0277953623005208>  
DOI: <https://doi.org/10.1016/j.socscimed.2023.116163>
- Brand, D., Zafiropulo, P., 1983. On communicating finite-state machines. *Journal of the ACM (JACM)* 30 (2), 323–342.
- Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., Xie, X., 3 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 39.  
DOI: 10.1145/3641289
- Colledanchise, M., Ögren, P., 8 2017. Behavior trees in robotics and ai: An introduction. *Behavior Trees in Robotics and AI*.  
URL: <http://arxiv.org/abs/1709.00084><http://dx.doi.org/10.1201/9780429489105>  
DOI: 10.1201/9780429489105
- Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al., 2023. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.
- Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al., 2022. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*.
- Joshi, J., 2025. The evolution of agentic ai: From rule-based systems to autonomous agents. *IJSAT-International Journal on Science and Technology* 16 (4).
- Kapoor, S., Stroebel, B., Siegel, Z. S., Nadgir, N., Narayanan, A., 2024. Ai agents that matter. *arXiv preprint arXiv:2407.01502*.
- Kolt, N., 2025. Governing ai agents. *arXiv preprint arXiv:2501.07913*.
- Koubaa, A., et al., 2017. *Robot Operating System (ROS)*. Vol. 1. Springer.
- Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., Zeng, A., 2022. Code as policies: Language model programs for embodied control. *arXiv preprint arXiv:2209.07753*.
- Liu, R., Zhang, X., 2019. A review of methodologies for natural-language-facilitated human-robot cooperation. *International Journal of Advanced Robotic Systems* 16 (3), 1729881419851402.
- Macenski, S., Martin, F., White, R., Clavero, J. G., 2020. The marathon 2: A navigation system. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.  
DOI: 10.1109/IROS45743.2020.9341231
- Merino-Fidalgo, S., Sánchez-Girón, C., Zalama, E., Gómez-García-Bermejo, J., Duque-Domingo, J., 12 2025. Behavior tree generation and adaptation for a social robot control with llms. *Robotics and Autonomous Systems* 194, 105165.  
URL: <https://www.sciencedirect.com/science/article/pii/S0921889025002623?via%3Dihub>  
DOI: 10.1016/J.ROBS.2025.105165
- Mishra, B., Kertesz, A., 2020. The use of mqtt in m2m and iot systems: A survey. *IEEE Access* 8, 201071–201086.  
DOI: 10.1109/ACCESS.2020.3035849
- Petrich, L., Jin, J., Dehghan, M., Jagersand, M., 2022. A quantitative analysis of activities of daily living: Insights into improving functional independence with assistive robotics. In: *2022 International Conference on Robotics and Automation (ICRA)*. pp. 6999–7006.  
DOI: 10.1109/ICRA46639.2022.9811960
- Temí, 2024. Introducing temí robot v3. Accessed: 03-02-2026.  
URL: <https://www.robotemí.com/product/temí-sales-contact/>
- Wang, Z., Kodner, J., Rambow, O., 2025. Exploring limitations of llm capabilities with multi-problem evaluation. In: *The Sixth Workshop on Insights from Negative Results in NLP*. pp. 121–140.
- Yamao, K., Kanaoka, D., Isomoto, K., Mizutani, A., Tanaka, Y., Tamukoh, H., 2024. Development of a saycan-based task planning system capable of handling abstract nouns. In: *Proceedings of International Conference on Artificial Life & Robotics (ICAROB2024)*. ALife Robotics, pp. OS15–4.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., Cao, Y., 2022. React: Synergizing reasoning and acting in language models. In: *The eleventh international conference on learning representations*.
- Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., et al., 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In: *Conference on Robot Learning*. PMLR, pp. 2165–2183.