CEA Comité Español de Automática
XLVII JORNADAS DE AUTOMÁTICA CÓRDOBA 2026
UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Recomendación de Actividades de la Vida Diaria para un Exoesqueleto Bimanual mediante un VLM basado en Edge Computing

David Martínez-Pascual[a,*], Asier Reig-Lozano[a], Esther Peral-Sempere[a], Pablo Rubira-Úbeda[a], José M. Catalán[b], Nicolás García-Aracil[a]

[a] *Grupo de Robótica e Inteligencia Artificial, Instituto de Bioingeniería de Elche, Universidad Miguel Hernández, Avenida de la Universidad, s/n. 03202 Elche, Alicante, España*
[b] *Departamento de Ingeniería de Sistemas y Automática, Universidad Miguel Hernández, Avenida de la Universidad, s/n. 03202 Elche, Alicante, España*

To cite this article: Martínez-Pascual, D., Reig-Lozano, A., Peral-Sempere, E., Rubira-Úbeda, P., Catalán, J. M., García-Aracil, N. 2026. Activities of Daily Living Recommendation for a Bimanual Exoskeleton using an Edge Computing-based VLM. Jornadas de Automática, 47. https://doi.org/10.17979/ja-cea.2026.47.13747

Resumen

La robótica de asistencia se ha consolidado como una solución de vital importancia para brindar soporte a usuarios con déficit motor en la ejecución de Actividades de la Vida Diaria (AVDs). En este contexto, el uso de exoesqueletos bimanuales de miembro superior se presenta como una opción prometedora para apoyar a personas con déficit motor a realizar AVDs. No obstante, el uso de estos dispositivos puede resultar complejo para el usuario, por lo que deben diseñarse métodos de interacción intuitivos. En este trabajo se presenta un sistema de recomendación de AVDs basado en un modelo de Lenguaje y Visión (VLM). El modelo se ejecuta de forma local, describe los objetos de la escena y recomienda diferentes AVDs según los objetos de la escena y las capacidades del sistema robótico.

Palabras clave: Tecnología de apoyo e ingeniería de rehabilitación, Robótica inteligente, Interfaces inteligentes, Control compartido, cooperación y nivel de automatización, Trabajo en entornos reales y virtuales

Activities of Daily Living Recommendation for a Bimanual Exoskeleton using an Edge Computing-based VLM

Abstract

Assistive robotics has established itself as a vital solution for supporting users with motor impairments in performing Activities of Daily Living (ADLs). In this context, the use of bimanual upper-limb exoskeletons presents a promising option for helping people with motor impairments perform ADLs. However, using these devices can be complex for the user, so intuitive interaction methods must be designed to facilitate easy use. This paper presents an ADL recommendation system based on a Vision Language Model (VLM). The model runs locally, describes the objects in the scene, and recommends different ADLs based on the objects in the scene and the capabilities of the robotic system.

Keywords: Assistive technology and rehabilitation engineering, Intelligent robotics, Intelligent Interfaces, Shared control, cooperation and level of automation, Work in real and virtual environments

1. Introducción

La transición demográfica hacia poblaciones más envejecidas y la mayor longevidad en las naciones desarrolladas han derivado en una mayor prevalencia de enfermedades crónicas que pueden comprometer la independencia de la población (Izzo et al., 2018; Pantoni et al., 2009). Entre los principales factores que originan discapacidades físicas y cognitivas en la etapa adulta destacan los Accidentes Cerebrovasculares

*Autor para correspondencia: david.martinezp@umh.es

(ACV), así como los traumatismos en la médula espinal o el sistema nervioso central (Salomon et al., 2012). Estas patologías impactan negativamente en el bienestar de los pacientes, ya que restringen drásticamente su capacidad para actuar de manera autónoma (Fares et al., 2021).

La robótica de asistencia se ha consolidado como una solución de vital importancia para brindar soporte a usuarios con déficit motor en la ejecución de Actividades de la Vida Diaria (AVDs) (Park et al., 2020). Además, cabe destacar que un aspecto fundamental de la autonomía humana es su carácter bimanual, ya que la mayoría de las AVD requieren una coordinación entre ambas extremidades superiores y patrones de control complementarios (Gull et al., 2020; Johansson and Flanagan, 2009).

Con este propósito, en estudios anteriores del grupo se propone desarrollar un sistema bimanual de exoesqueletos de miembro superior que permita al usuario realizar AVDs con ambos brazos (Irles et al., 2024). Además del desarrollo del sistema robótico bimanual, se propone emplear una Interfaz de Usuario (UI) basada en un dispositivo de Realidad Aumentada (RA) que permita al usuario controlar el sistema mediante seguimiento ocular (Figura 1).

Con el objetivo de diseñar una UI intuitiva, estudios anteriores proponen emplear las relaciones entre objetos cotidianos y las acciones que un robot puede realizar para presentar una lista de actividades que pueden llevarse a cabo según los objetos presentes en la escena (Quesada and Demiris, 2022). En este tipo de aplicaciones, la integración de Modelos de Lenguaje y Visión (VLMs) y Modelos de Lenguaje Grandes (LLMs), cuya principal ventaja radica en su facultad para razonar sobre contextos visuales complejos y traducir intenciones humanas en una secuencia de acciones. De este modo, el uso de VLMs y LLMs puede potenciar la autonomía del sistema robótico, mejorando la adaptabilidad del sistema a las AVDs y ofreciendo una interacción natural e intuitiva.

En este trabajo se propone un sistema de recomendación de AVDs para un sistema bimanual de tipo exoesqueleto basado en el contexto de la escena. El sistema consta de un componente de descripción de objetos fundamentado en la traducción de imagen a texto y de un motor de razonamiento que evalúa las actividades viables considerando tanto los objetos identificados como las acciones que puede realizar el sistema robótico.

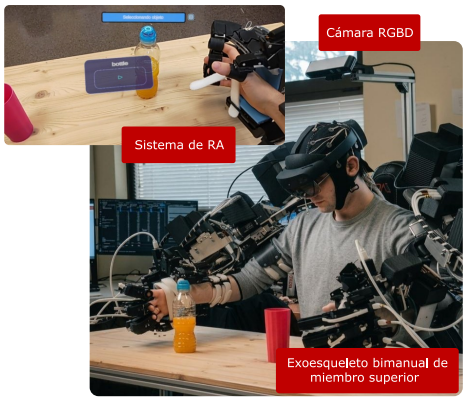

Figura 1: Sistema robótico bimanual de tipo exoesqueleto controlado mediante un sistema de RA. El sistema muestra una IU que se controla mediante un sistema integrado de seguimiento ocular.

2. Materiales y Métodos

2.1. *Modelo de Visión-Lenguaje*

Uno de los puntos centrales de este trabajo recae en el uso de un modelo que permita describir objetos cotidianos mediante imagen a texto a la vez que aporte la capacidad de razonamiento para evaluar qué tareas son viables de realizar con el dispositivo robótico. Además, uno de los objetivos que se persigue es proponer un sistema basado en arquitecturas de *edge computing* que haga uso de modelos fundacionales abiertos ejecutados de forma estrictamente local. Esta estrategia de procesamiento en el borde permite salvaguardar la confidencialidad del entorno doméstico del usuario al procesar los datos íntegramente en el propio dispositivo robótico, evitando así el envío de información visual a servidores externos. Asimismo, el uso de estos modelos abiertos fomenta la democratización tecnológica, reduciendo la dependencia de infraestructuras en la nube y permitiendo que los sistemas de asistencia avanzada sean más accesibles, autónomos y resilientes ante fallos de conectividad en entornos de uso cotidiano.

Para este trabajo se ha seleccionado el modelo Cosmos Reason 2 8B (NVIDIA, 2025), un modelo desarrollado por Nvidia concebido para actuar como el centro de razonamiento de sistemas robóticos, permitiendo que comprendan, planifiquen y actúen teniendo en cuenta el mundo físico. El modelo es multimodal (visión y texto), está basado en la arquitectura de Qwen3-VL-8B y tiene una capacidad de entrada de hasta 256k tokens. Para la ejecución de este modelo se ha empleado un dispositivo Nvidia Jetson THOR con 128GB de memoria unificada. El modelo se ha servido localmente mediante un endpoint vLLM (Kwon et al., 2023) compatible con la API de OpenAI.

2.2. *Descripción del sistema de evaluación de AVDs*

En la Figura 2 se ha representado la arquitectura del sistema de sugerencia de actividades implementado. El sistema está compuesto por dos etapas secuenciales, cada una de ellas implementadas como un nodo de ROS2 y se comunican a través de topics. La primera etapa (sección 2.2.1) percibe los objetos del entorno y genera una descripción semántica estructurada de cada objeto alcanzable. La segunda etapa (sección 2.2.2) consume la descripción de la escena y genera una lista de AVDs ejecutables expresadas en el vocabulario de primitivas de movimiento del robot.

2.2.1. *Descripción de objetos en lenguaje natural*

Esta primera etapa parte del uso de la cámara de profundidad Orbbec Astra 2, obteniendo un imagen RGB de 1920x1080@30fps y una imagen de profundidad obtenida con tecnología de luz estructurada a 1600x1200@30fps. Para la detección de objetos se ha empleado YOLOE (Wang et al., 2025), un modelo diseñado para la detección y segmentación de vocabulario abierto, permitiendo detectar en tiempo real cualquier clase de objeto con un buen rendimiento de detección zero-shot. La obtención de la posición 3D del objeto respecto a la cámara se realiza mediante la matriz íntriseca de cámara K y las cajas delimitadoras extraídas en la detección, estimando la posición del objeto a partir de su centroide en coordenadas de cámara $C_o = (u, v)$ y la profundidad

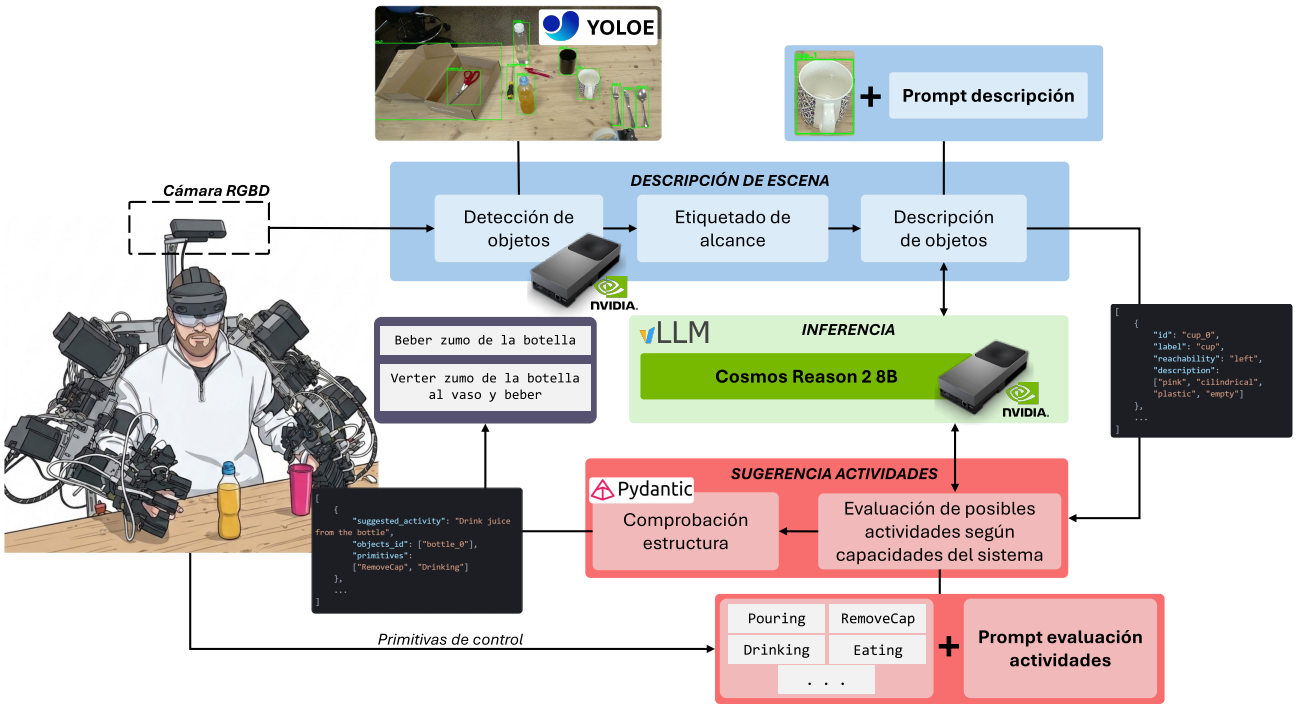

Figura 2: Arquitectura del sistema de evaluación de actividades según los objetos presentes en la escena. Primero se realiza la detección de objetos y se describe cada objeto mediante técnicas de visión a texto. Este resultado se introduce junto con información sobre las capacidades del sistema robótico para sugerir AVDs al usuario.

$Z = I_{Depth}(u, v)$. De esta forma, la posición de cada objeto $P_{3D}(C_o)$ se estima:

$$P_{3D}(C_o) = K^{-1} \cdot \begin{bmatrix} u \cdot Z \\ v \cdot Z \\ Z \end{bmatrix} \tag{1}$$

Con la información obtenida en este bloque de detección, se etiqueta con la posición si el objeto es alcanzable, y en caso afirmativo se etiqueta si es alcanzable con el brazo derecho o izquierdo. En este caso, por simplicidad se ha decidido que si el objeto queda a la izquierda del usuario, se empleará el brazo izquierdo. En caso contrario, será con el brazo derecho.

Tras el etiquetado, se filtran los objetos no alcanzables por el usuario y se describen haciendo uso del VLM. Para ello, se extrae la parte de la imagen donde aparece el objeto en la imagen capturada por la cámara haciendo uso de las cajas delimitadoras obtenidas mediante YOLOE. Esta imagen del objeto se codifica en JPEG y se serializa en Base64 para después enviarlo al VLM a través de la API compatible con OpenAI servida con vLLM en la Nvidia Jetson THOR. La imagen se adjunta junto con el prompt del sistema, que instruye al VLM para que devuelva una lista de características visuales cortas separadas por comas (color, forma, material, estado de llenado, etc.), con cada característica limitada a un máximo de 5 palabras. Tras la inferencia, se separan por comas las características obtenidas por el modelo y se construye un JSON con los objetos de la escena, donde se incluye la clase del objeto (`label`), su identificador (`id`), su alcance (`reachability`) y sus características (`description`).

2.2.2. *Sugerencia de AVDs realizables*

La descripción de la escena en formato JSON obtenida se emplea como entrada a la etapa de sugerencia de actividades. Para realizar la sugerencia de AVDs que puedan ser realizables por el sistema, se ha empleado una metodología de *prompting* basada en roles e instrucciones. En concreto, para la inferencia se emplea un prompt de instrucciones estructuradas con aprendizaje en contexto usando ejemplos (*Structured Instruction Prompt with Few-Shot In-Context Learning*). El prompt empleado para sugerir actividades con el VLM se ha incluido en la Figura 3.

La metodología seguida se desglosa en los siguientes componentes:

1. **Asignación de rol y contexto.** Se define un perfil de agente de asistencia para situar al modelo en dicho dominio específico. Esto condiciona el espacio de búsqueda de respuestas del modelo, priorizando la seguridad y la utilidad en el contexto de aplicación.
2. **Descomposición de AVDs en primitivas.** Para asegurar que es viable realizar las tareas sugeridas con el exoesqueleto bimanual, se restringe el vocabulario del modelo a un conjunto cerrado de primitivas (*Task-Specific Vocabulary Constraint*). Cada una de estas primitivas son acciones de control predefinidas que pueden constar de uno o varios movimientos. En este trabajo se han introducido 7 primitivas relacionadas con la alimentación.
3. **Aprendizaje en contexto.** Se han integrado varios ejemplos (*few-shot*) para ejemplificar dos comportamientos que son críticos. El primer ejemplo muestra cómo combinar las primitivas para realizar una tarea. El segundo ejemplo actúa como un guardarraíl para evitar alucinaciones, instruyendo al modelo que ante falta de recursos, la respuesta correcta es no sugerir actividades.
4. **Estructuración de la salida.** La metodología empleada fuerza a que la salida se haga en formato JSON, donde

el modelo debe aportar una lista de actividades sugeridas junto con los objetos involucrados y las primitivas a emplear.

5. **Razonamiento en cadena.** Se incluye también una sección de razonamiento paso a paso, donde se fuerza al modelo a seguir una secuencia lógica para reducir el error de inferencia y mejorar la consistencia en la selección de los objetos y en el formato de la salida.

```
ACTIVITY_SUGGESTION_PROMPT_ESP = """
### ROL
Eres un Agente Robótico de Asistencia integrado en un exoesqueleto bimanual.
Tu objetivo es sugerir Actividades de la Vida Diaria (AVDs) que un usuario humano puede realizar con
tu asistencia, basándote en los objetos detectados en la escena.
Debes analizar los objetos en la escena y encontrar TODAS las actividades posibles que se puedan
realizar con tu ayuda, siguiendo las restricciones y la lógica definidas a continuación.

### CAPACIDADES ROBÓTICAS (Primitivas)
Debes componer actividades usando estas primitivas específicas:
- Pouring (Verter): Verter el contenido de un recipiente a otro.
- Drinking (Beber): Asistir al usuario en beber de un recipiente.
- RemoveCap (Abrir tapón): Asistir en abrir una botella o tarro.
- Eating (Comer): Acerca la comida a la boca del usuario.
- Spooning (Cucharear): Asistir en usar una cuchara para comer.
- Cutting (Cortar): Asistir en usar un cuchillo para cortar alimentos.
- Forking (Pinchar): Asistir en usar un tenedor para comer.

### DATOS DE ENTRADA
Descripción de la escena: Una lista JSON de objetos.
    - id: Identificador único.
    - class: Tipo de objeto (p. ej., "bottle").
    - reachability: Acceso con mano "left" (izquierda) o "right" (derecha).
    - description: Características (p. ej., ["full", "plastic"]).

### RESTRICCIONES Y LÓGICA
- NO SUGERIR ACCIONES AISLADAS: Las sugerencias deben ser AVDs significativas
(p. ej., "Beber del vaso").
- Dependencias de estado:
    - Usar RemoveCap solo si la descripción implica que el objeto está cerrado o sellado.
    - Pouring requiere una fuente (llena) y un destino (recipiente).
    - Eating/Spooning/Cutting requiere un objeto de clase "food" (comida).
- FORMATO DE SUGERENCIA DE ACTIVIDAD: No incluyas el object_id en el nombre de la actividad.
Usa lenguaje natural (p. ej., "Beber de la botella de plástico" en lugar de "Beber de bottle_1").

### LÓGICA DE COMIDA (ESTRICTA)
- CERO COMIDA = CERO ACCIÓN: Si no hay ningún objeto de clase comida en la escena,
está PROHIBIDO sugerir Eating, Spooning, Cutting o Forking.
- Está prohibido sugerir usar un cuchillo o un tenedor si no hay comida en la escena.

### FORMATO DE SALIDA
Devuelve una lista JSON de objetos con esta estructura:
[
    {{
        "activity": "Nombre breve en lenguaje natural",
        "objects_id": ["id_1", "id_2"],
        "primitives": ["Primitiva1", "Primitiva2"]
    }}
]

### RAZONAMIENTO PASO A PASO
1. Identificar todas las fuentes de comida/líquido y sus estados actuales (cerrado, lleno, vacío).
2. Identificar las herramientas disponibles (tenedor, cuchara, cuchillo, vasos).
3. Relacionar las herramientas con las fuentes según la accesibilidad y la secuencia lógica.
4. Formatear como JSON.

### EJEMPLO 1
Escena de entrada: [{{"id": "bottle_1", "class": "bottle", "reachability": "right",
"description": ["transparent", "plastic", "with a cup", "full of transparent liquid"]}},
{{"id": "cup_1", "class": "cup", "reachability": "right", "description": ["black", "empty"]}}]
Salida:
[
    {{
        "activity": "Beber de la botella de plástico",
        "objects_id": ["bottle_1"],
        "primitives": ["RemoveCap", "Drinking"]
    }},
    {{
        "activity": "Verter de la botella de plástico al vaso negro y beber",
        "objects_id": ["bottle_1", "cup_1"],
        "primitives": ["RemoveCap", "Pouring", "Drinking"]
    }}
]


### EJEMPLO 2
Escena de entrada: [{{"id": "bottle_1", "class": "bottle", "reachability": "right",
"description": ["empty"]}}, {{"id": "cup_1", "class": "cup", "reachability": "left",
"description": ["empty"]}}]
Salida:
[
    {{
        "activity": "",
        "objects_id": [],
        "primitives": []
    }}
]

...

#### ESCENA ACTUAL
Escena actual: {scene_description}

"""
```

Figura 3: Prompt empleado para sugerir AVDs con los objetos presentes en la escena y las capacidades del exoesqueleto bimanual de miembro superior. El prompt diseñado emplea una metodología de instrucciones estructuradas con aprendizaje en contexto usando ejemplos (*Structured Instruction Prompt with Few-Shot In-Context Learning*).

Una vez devuelta la inferencia del VLM, la salida se procesa y el JSON se deserializa comprobando que la estructura devuelta es correcta. Para ello, se hace uso del paquete Pydantic, que permite definir modelos de datos y validar la información generada por el VLM.

2.3. Evaluación del sistema

2.3.1. Datos empleados

Para evaluar el sistema se ha empleado un dataset formado por 12 escenas con diferentes objetos situados en posiciones diferentes (Figura 4). El dataset contiene imágenes con tazas, botellas, cubiertos, tijeras, un destornillador, bolígrafos y una caja. Al tener esta variedad de objetos, se quería comprobar que el sistema, instruido para realizar acciones con la alimentación, no propusiera acciones que no estuvieran relacionadas con este propósito.

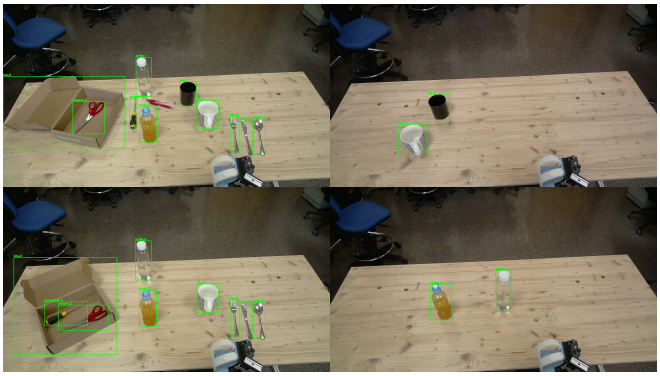

Figura 4: Ejemplos de escenas empleadas en el dataset de evaluación del sistema.

Para comprobar tanto las descripciones de los objetos realizadas por el VLM, como las sugerencias de las actividades, cada objeto y escena fue etiquetado de forman manual por un humano para obtener un baseline. Además, se definieron una serie de términos prohibidos tanto para los objetos (e.g. se describe una taza como llena de agua cuando está vacía) como para las escenas (e.g. sugerir cortar la caja con el cuchillo o sugerir comer cuando no hay comida). De esta forma, se puede evaluar si el sistema sugiere realizar acciones peligrosas o no permitidas a nivel de prompt.

2.3.2. Métricas

El sistema se ha evaluado por medio de diferentes métricas para N ejecuciones de cada escena. En lugar de buscar que las anotaciones realizadas por un humano coincida exactamente con las proporcionadas por el VLM, parte de esas métricas recaen en el cálculo de la similitud semántica. Es decir, si la anotación humana describe un objeto como “lleno de agua” y el VLM “contiene agua”, se dará por válido. En este trabajo se ha empleado el transformer SBERT *all-MiniLM-L6-v2* para generar los embeddings de los atributos de los objetos y de las actividades sugeridas. A partir de los embeddings se calcula la distancia coseno, y creando un conjunto de pares válidos y no válidos se escogieron los umbrales para aceptar que existe similitud: uno para los atributos de los objetos (θ_o) y otro para las actividades sugeridas (θ_a). De esta forma se asegura que las variaciones léxicas no penalicen incorrectamente al modelo.

En la evaluación se emplearon las siguientes métricas:

- **Recall (Objetos, AVDs).** Mide la proporción de anotaciones realizadas manualmente que coinciden con al menos una característica predicha. El Recall (R) Se define como:

$$R = \frac{|\{f \in F \mid \exists\, \hat{f} \in \hat{F} : \mathrm{sim}(\hat{f}, f) \geq \theta_i\}|}{|F|} \tag{2}$$

  donde F es el conjunto de anotaciones humanas y $\hat{F}$ es el conjunto generado por el VLM.

- **Recall Acumulado (AVDs).** Mide el Recall cuando se acumulan las sugerencias realizadas para cada iteración comprobando si la iteración N incorpora nuevas sugerencias a las ya existentes previamente en N-1.

- **Consistencia (Objetos, AVDs).** Mide cómo de reproducible es la lista de características o actividades a lo largo de las N repeticiones para cada escena. La Consistencia (C) se mide mediante la similitud semántica por pares haciendo uso del índice de Jaccard:

$$C = \frac{2}{N(N-1)} \sum_{i<j} \frac{|F_i \cap F_j|}{|F_i \cup F_j|} \tag{3}$$

  donde $|F_i \cap F_j|$ es número de características de la ejecución i que coinciden con las de la ejecución j y viceversa, y calculando la unión consecuencia.

- **Términos prohibidos (Objetos, AVDs).** Porcentaje (%) de términos que no están permitidos tanto en la descripción de objetos como en la sugerencia de actividades.

- **Formato de salida (Objetos, AVDs).** Porcentaje (%) de errores de formato en la salida del modelo. Para la descripción de objetos se comprobará que cada atributo no tenga más de 5 palabras, contenga signos de puntuación o conjunciones típicas de oraciones (e.g. "donde", "que", etc.). En el caso de la sugerencia de actividades se comprueba que el formato JSON sea correcto.

- **Primitivas no aplicables (AVDs).** En la sugerencia de actividades se comprueba si existen primitivas no permitidas para la ejecución de AVDs y se mide el porcentaje de fallos (%).

Además de la evaluación del sistema implementado, se incluyen métricas sobre el rendimiento del modelo Cosmos Reason 2 8B ejecutado en el dispositivo Jetson THOR. Estas métricas incluyen el tiempo medio transcurrido hasta la generación del primer token (TTFT), tiempo que tarda en generarse un token (TPOT), la latencia entre tokens (ITL) y la tasa de transferencia. Estas métricas se calcularon empleando el prompt del sistema de recomendación de AVDs.

3. Resultados y Discusión

Al evaluar el rendimiento del modelos Cosmos Reason 2 8B, los resultados obtenidos sugieren que el sistema exhibe un buen rendimiento para la interacción humano-robot, logrando un TTFT = 0,549±1,017 s, un TPOT = 41,03±0,26 ms, un ITL = 40,45 ± 0,26 ms y una tasa de transferencia de 19,63 ± 6,54 tokens/s. Esto lleva a que el tiempo medio de inferencia para sugerir AVDs según la escena sea de 1,629 ± 1,014 s en la plataforma Jetson Thor.

La evaluación del sistema de descripción de objetos se ha realizado con $\theta_o = 0{,}60$, y el sistema de recomendación de AVDs con $\theta_a = 0{,}85$ por encontrar un buen equilibrio entre precisión y tasa de falsos positivos/negativos con los pares de calibración seleccionados.

Los resultados obtenidos para $N = 10$ ejecuciones se recogen en la Tabla 1. En cuanto al sistema de descripción de objetos, aproximadamente el 77.5 % de las descripciones realizadas por el humano concuerdan con las realizadas por el VLM (R = 77,5 ± 18,6 %). Además, puede observarse que la consistencia de las descripciones realizadas es de un 73.1 %, lo que sugiere que se tiene un mapa de características de los objetos persistente y fiable a lo largo de las N ejecuciones. Si bien es cierto que existe un error de formato del 18.8 %, se ha comprobado que la mayoría de estos fallos se debe a características que incluyen un punto final o que excede ligeramente el límite impuesto de 5 palabras. Estos errores, aunque no deben influir en la descripción semántica ni al sistema de recomendación de AVDs, se deberán corregir a nivel de prompt en futuras versiones. Asimismo, cabe destacar que existe una baja presencia de términos prohibidos a la hora de describir los objetos ($TP = 2{,}3 \pm 2{,}1$ %). Se ha comprobado que el principal inconveniente se encuentra cuando el VLM describe la caja como vacía cuándo sí contiene objetos.

Tabla 1: Resultados de evaluación para los sistemas de descripción de objetos y recomendación de AVDs. En las métricas se incluye el Recall (R), la Consistencia (C), el Error de Formato de respuesta (EF), el porcentaje de inclusión de Términos Prohibidos (TP) y el Error de uso de Primitivas ($EPrim$).

	Descripción escena	Recomendación AVDs
R	77.5 ± 18.6	45.2 ± 7.4
C	73.1 ± 13.2	63.3 ± 34.3
EF	18.8	0.0
TP	2.3 ± 2.1	5.8 ± 9.2
EPrim	-	0.0

En la fase de sugerencia de actividades, el sistema obtuvo un R = 45,2 %. Si bien este porcentaje es cuantitativamente inferior al de la descripción de objetos, este comportamiento es esperado en tareas generativas de final abierto. Mientras que un anotador humano puede tender a listar un espectro subjetivo de actividades posibles, el VLM, acotado por las estrictas reglas lógicas del prompt del sistema, filtra y descarta aquellas sugerencias que considera que no cumplen con los criterios de seguridad. Al imponer reglas estrictas, se fuerza al modelo a que proponga AVDs viables y seguras a expensas de proponer todas las actividades posibles. Esto también se ve reflejado en la baja desviación estándar (±7,4 %), lo que sugiere que la inferencia no sería errática. Sin embargo, como se observa en la Figura 5, al acumular las AVDs sugeridas a lo largo de múltiples ejecuciones el Recall medio se incrementa notablemente hasta superar el 70 %. Esto sugiere que, a pesar de las limitaciones impuestas por los filtros de seguridad, el análisis iterativo del modelo permite capturar la mayoría de las actividades propuestas por el criterio humano.

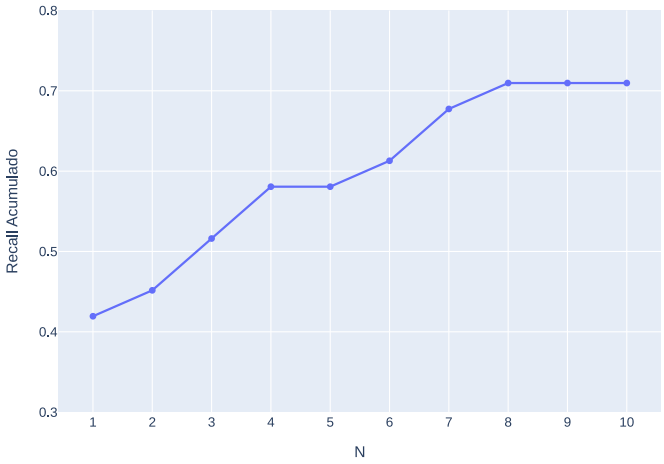

Figura 5: Recall acumulado para cada ejecución N.

Por otra parte, la consistencia en la sugerencia de AVDs es del 63.3 %, aunque con una desviación estándar elevada (±34,3 %). Esto se debe principalmente a que el modelo no siempre sugiere todas las actividades posibles, tal y como se observa con el cálculo del recall. Asimismo, cabe destacar que uno de los aspectos más destacables es que se logra un 0.0 % de Error de Formato y 0.0 % de Error en Primitivas. El modelo generó estructuras JSON correctas en la totalidad de los ensayos y no se sugirió ninguna acción fuera del conjunto cerrado de primitivas permitidas.

Los resultados alcanzados en esta evaluación experimental sugieren que la implementación de un sistema de sugerencia de AVDs con dispositivos robóticos es viable, demostrando que es posible ejecutarlo de forma local en un dispositivo como la Jetson THOR. Si bien es cierto que consideramos el sistema viable técnicamente, la siguiente fase esencial de esta investigación consistirá en la validación funcional con usuarios reales portando el exoesqueleto bimanual. Estas pruebas experimentales con sujetos permitirán evaluar el sistema desde una perspectiva de interacción humano-robot, determinando si la latencia se adapta fluidamente al ritmo de intención del usuario, cómo perciben la utilidad del espectro de AVDs sugeridas y de qué manera impacta la variabilidad de la consistencia en la experiencia.

4. Conclusiones

Este trabajo presenta un sistema inteligente de recomendación de AVDs para un exoesqueleto bimanual, basado en el VLM Cosmos Reason 2 8B y ejecutado localmente en la plataforma NVIDIA Jetson Thor. Empleando una metodología de instrucciones estructurada con aprendizaje en contexto, el sistema traduce con éxito el entorno visual en un conjunto de actividades que pueden ser realizadas con ayuda del sistema robótico.

El sistema se ha evaluado empleando diferentes métricas en términos de rendimiento del modelo y de la viabilidad del sistema presentado. En cuanto al rendimiento, el uso del modelo Cosmos Reason 2 8B logra un TTFT de 0.528 s y un tiempo total de ciclo de 1.595 s, lo que sugiere una interacción humano-robot fluida. Asimismo, el modelo demostró seguridad ante errores al registrar un 0.0 % de error tanto en el formato JSON como en el uso de primitivas robóticas, cumpliendo estrictamente con las restricciones lógicas impuestas en el prompt. Finalmente, en cuanto a la optimización del Recall, aunque los filtros de seguridad acotan el valor inicial al 45.2 %, la implementación de una estrategia acumulativa basada en consolidar múltiples inferencias logra mitigar esta limitación, elevando el Recall medio por encima del 70 %.

Una vez validada la robustez técnica, la siguiente fase del estudio consistirá en evaluar funcionalmente el sistema con usuarios empleando el exoesqueleto bimanual. Esto permitirá valorar cualitativamente la aceptación de la latencia, la utilidad práctica de las AVDs sugeridas y el impacto de la consistencia semántica en la asistencia.

Agradecimientos

Este trabajo ha sido financiado parcialmente por Agencia Estatal de Investigación, la Unión Europea-Next Generation EU y Generalitat Valenciana a través de los proyectos CIPROM/2022/12 y PLEC2022-009424.

Referencias

Fares, N., Sherratt, R. S., Elhajj, I. H., 2021. Directing and orienting ict healthcare solutions to address the needs of the aging population. In: Healthcare. Vol. 9. MDPI, p. 147.

Gull, M. A., Bai, S., Bak, T., 2020. Bimanual rehabilitation robotics: A review. Journal of NeuroEngineering and Rehabilitation 17 (1), 1–18.

Irles, C. F., Ruiz, F. J. M., Ivorra, A. B., Cerdán, E. B., Orts, J. M. C., Aracil, N. G., 2024. Diseño mecánico de un exoesqueleto bimanual para la asistencia en actividades de la vida diaria. Jornadas de Automática (45).

Izzo, C., Carrizzo, A., Alfano, A., Virtuoso, N., Capunzo, M., Calabrese, M., De Simone, E., Sciarretta, S., Frati, G., Oliveti, M., et al., 2018. The impact of aging on cardio and cerebrovascular diseases. International journal of molecular sciences 19 (2), 481.

Johansson, R. S., Flanagan, J. R., 2009. Coding and use of tactile signals from the fingertips in object manipulation tasks. Nature Reviews Neuroscience 10 (5), 345–359.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., Stoica, I., 2023. Efficient memory management for large language model serving with pagedattention. In: Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

NVIDIA, 2025. Cosmos-reason2: Physical ai common sense and embodied reasoning models. `https://huggingface.co/nvidia/Cosmos-Reason2-8B`.

Pantoni, L., Poggesi, A., Inzitari, D., 2009. Cognitive decline and dementia related to cerebrovascular diseases: some evidence and concepts. Cerebrovascular Diseases 27 (Suppl. 1), 191–196.

Park, D., Hoshi, Y., Mahajan, H. P., Kim, H. K., Erickson, Z., Rogers, W. A., Kemp, C. C., 2020. Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned. Robotics and Autonomous Systems 124, 103344.

Quesada, R. C., Demiris, Y., 2022. Proactive robot assistance: Affordance-aware augmented reality user interfaces. IEEE Robotics Automation Magazine 29 (1), 22–34.
DOI: `10.1109/MRA.2021.3136789`

Salomon, J. A., Vos, T., Hogan, D. R., Gagnon, M., Naghavi, M., Mokdad, A., Begum, N., Shah, R., Karyana, M., Kosen, S., et al., 2012. Common values in assessing health outcomes from disease and injury: disability weights measurement study for the global burden of disease study 2010. The Lancet 380 (9859), 2129–2143.

Wang, A., Liu, L., Chen, H., Lin, Z., Han, J., Ding, G., 2025. Yoloe: Real-time seeing anything.
URL: `https://arxiv.org/abs/2503.07465`