DOBLE MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL Y MÁSTER EN ADMINISTRACIÓN DE EMPRESAS (MBA)

TRABAJO FIN DE MÁSTER

ANÁLISIS DE ESCENAS PARA ROBÓTICA Y APLICACIONES DE TELEPRESENCIA INMERSIVA

Autor: Juan Modesto Espinosa Bogas

Directora: D.ª Marta Orduna Cortillas

Madrid

Julio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título ANÁLISIS DE ESCENAS PARA ROBÓTICA Y APLICACIONES DE TELEPRESENCIA INMERSIVA

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2024/2025 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Juan Modesto Espinosa Bogas Fecha: 18 / 07 / 2025

Autorizada la entrega del proyecto

EL DIRECTORA DEL PROYECTO

ORDUNA CORTILLAS MARTA - 73207633J

Digitally signed by ORDUNA CORTILLAS MARTA - 73207633J
Date: 2025.07.18 12:18:11 +02'00'

Fdo.: D.ª Marta Orduna Cortillas Fecha: 18/07/2025

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

DOBLE MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL Y MÁSTER EN ADMINISTRACIÓN DE EMPRESAS (MBA)

TRABAJO FIN DE MÁSTER

ANÁLISIS DE ESCENAS PARA ROBÓTICA Y APLICACIONES DE TELEPRESENCIA INMERSIVA

Autor: Juan Modesto Espinosa Bogas

Directora: D.ª Marta Orduna Cortillas

Madrid

Julio de 2025

A mi familia

Índice

Agradecimientos

En primer lugar, quiero expresar mi máximo agradecimiento a D.ª Marta Orduna, directora de este proyecto, por haber tutorizado el trabajo, así como por su constante apoyo y disposición para ayudarme siempre que lo he necesitado. Hago extensible este agradecimiento a D.ª Amaya Jiménez, encargada de dirigirlo en los primeros meses. Ambas han sido dos personas fundamentales en el desarrollo del TFM, sin cuyo acompañamiento no habría llegado a buen puerto.

Por supuesto, debo agradecer también a mi familia, especialmente a mis padres y a mi hermana, que son el verdadero pilar de mi vida. A ellos les debo todo lo que he conseguido, y en particular, la culminación de este TFM. Sin su apoyo incondicional y su confianza no habría sido capaz de realizarlo.

Por último, agradecer a todos los integrantes del Extended Reality Lab de Nokia Spain, y en particular a su líder, D. Álvaro Villegas, en cuyo seno se ha desarrollado este TFM, por la cálida acogida que me dieron tanto a mí como a mis compañeros, así como por toda la ayuda prestada a lo largo de estos meses siempre que fue necesaria. Ha sido un verdadero honor poder trabajar y aprender con ellos.

Resumen

ANÁLISIS DE ESCENAS PARA ROBÓTICA Y APLICACIONES DE TELEPRESENCIA INMERSIVA

Juan Modesto Espinosa Bogas

juanmodesto.espi@gmail.com

Palabras clave: *Análisis de escenas; robótica; telepresencia inmersiva; LLM.*

1. INTRODUCCIÓN

La inteligencia artificial (IA) ha evolucionado desde sus orígenes en los años 50 hasta convertirse en una tecnología clave en múltiples sectores gracias al aprendizaje profundo y al aumento del poder computacional [1]. En este contexto, una de sus recientes aplicaciones más avanzadas es el **análisis de escenas**, que permite, a diferencia de la simple detección de objetos, no solo identificar objetos en un cuadro o vídeo, sino también comprender sus relaciones espaciales, funcionales y contextuales, de manera similar a como hacen las personas [2]. Las principales funcionalidades explotadas en este proyecto son:

- VQA (*Visual Question Answering*), es decir, respuestas a preguntas basadas en un cuadro.
- Generación de descripciones (*captioning*), ya sea de un cuadro o de un fragmento de vídeo.

Este Trabajo Fin de Máster (TFM) se ha desarrollado en el XRLab de Nokia Spain S.A., un laboratorio especializado en tecnologías de realidad extendida (XR) e integración con sistemas físicos como robots, entornos complejos en los que dicho análisis de escenas puede ser muy valioso. El fin principal del proyecto es investigar, implementar y evaluar algoritmos de análisis de escenas basados en IA, integrados con modelos avanzados de lenguaje (LLM), para dotar a los sistemas de una comprensión visual significativa del entorno, facilitando la autonomía, la interacción hombre-máquina y la toma de decisiones en tiempo real. El trabajo también aborda los retos actuales de estas tecnologías, como la trazabilidad de modelos, el coste computacional o la falta de robustez en escenarios dinámicos.

2. OBJETIVOS

Los principales objetivos del proyecto son:

- Estudiar técnicas avanzadas de análisis de escenas con IA, incluyendo vídeo, imágenes y arquitecturas multimodales.
- Integrar un sistema de análisis de escenas en un robot zoomórfico para interpretar su entorno visual.
- Evaluar y comparar modelos mediante métricas objetivas y herramientas basadas en LLM-as-a-Judge.
- Diseñar una arquitectura cliente-servidor para procesar imágenes de forma remota.
- Establecer un canal de comunicación entre sistemas de telepresencia y servidores de análisis.
- Unificar los resultados visuales con razonamiento en lenguaje natural mediante modelos LLM.

3. SOLUCIÓN PROPUESTA

La solución técnica desarrollada en este TFM se basa en una arquitectura distribuida cliente-servidor que permite la integración de múltiples fuentes de vídeo, el procesamiento paralelo mediante algoritmos avanzados de análisis de escenas, y una capa de razonamiento cognitivo basada LLM. Este enfoque sigue el esquema conceptual sentir-pensar-actuar (*sense-think-act*), ampliamente utilizado en robótica cognitiva por su capacidad para estructurar sistemas autónomos [3], y cuya aplicación en sistemas de telepresencia, si bien no tan extendida, resulta coherente y prometedora [4]. En este caso:

- La capa «sentir» corresponde a la percepción visual del entorno, captada por sensores como cámaras o dispositivos XR y procesada mediante algoritmos de análisis de escenas.
- La capa «pensar» representa la capa cognitiva, donde un LLM interpreta los resultados visuales,

contextualiza la información y genera descripciones, inferencias o acciones.
- La capa «actuar» hace referencia a los módulos que muestran, almacenan o ejecutan decisiones, ya sea a nivel de visualización, interacción o comandos.

La arquitectura distribuida propuesta se compone de estos bloques, cuya representación esquemática puede verse en la Figura 1:

- **Clientes de envío:** Recogen el flujo de diversos dispositivos físicos que capturan vídeo en tiempo real lo envían a los servidores a una resolución de 640 × 480 px, y a una frecuencia de 3 fps, para optimizar el ancho de banda y la latencia en la transmisión, aunque sin sacrificar en exceso su calidad.
- **Servidores de análisis:** Varios servidores operando en paralelo, cada uno ejecutando un algoritmo distinto de análisis de escenas. Estos modelos procesan el vídeo entrante de forma independiente y envían los cuadros procesados y los resultados a los siguientes clientes.
- **Clientes de recepción:** Módulos encargados de recibir los resultados desde los servidores de análisis. Estos clientes permiten visualizar los cuadros procesados e imprimir los resultados, así como registrarlos en un archivo estructurado, que servirá de base para el razonamiento posterior.

Figura 1. Esquema básico de la arquitectura planteada

Además, se incluye una:

- **Capa de razonamiento (LLM):** Un modelo de lenguaje de gran escala lee el archivo estructurado y genera distintas respuestas según el contexto o caso de uso: desde descripciones detalladas hasta resúmenes o decisiones operativas. Esta capa es esencial para transformar datos estrictamente visuales en conocimiento comprensible y útil.

4. MEDIOS DE CAPTACIÓN

La captación eficiente del entorno visual es un componente esencial para cualquier sistema basado en análisis de escenas. En este proyecto se han utilizado dos plataformas complementarias:

Robot Unitree Go2

Es un robot cuadrúpedo bioinspirado (tipo «perro»), desarrollado por la empresa china Unitree Robotics, que por su forma ofrece una excelente capacidad de adaptación a terrenos irregulares. Dotado de 12 grados de libertad, sensores integrados capacidad de ejecución de scripts en tiempo real y conectividad ROS, representa una solución idónea para tareas de inspección, exploración o asistencia en entornos reales [5].

En este proyecto, el Go2 se ha utilizado como plataforma de adquisición de vídeo en movimiento, permitiendo evaluar el rendimiento del análisis de escenas en situaciones dinámicas y semiestructuradas.

La integración con el Go2 requirió superar importantes barreras técnicas. Aunque se exploraron opciones como el SDK ROS 2 oficial, estas resultaron poco viables por incompatibilidades o inestabilidad. La solución fue utilizar GStreamer como herramienta intermedia: el flujo captado por el robot se redirigió mediante UDP a una dirección local, permitiendo su recepción fluida y en tiempo real en el equipo de análisis.

El Búho

El Búho es un sistema portátil de telepresencia inmersiva desarrollado en el XRLab de Nokia, diseñado para capturar vídeo en 360° y permitir la participación remota de usuarios mediante cascos de realidad virtual. Supera las limitaciones de las videoconferencias tradicionales y ofreciendo una experiencia mucho más rica en cuanto a contexto espacial, comunicación no verbal e inmersión sensorial [6]. Está compuesto por dos cámaras ojo de pez montadas sobre una estructura ligera e impresa en 3D, con una unidad de control que codifica el vídeo en tiempo real y lo transmite. El sistema incorpora también audio bidireccional y conectividad multimodal lo que permite su uso tanto en interiores como en exteriores.

En este proyecto, el Búho se ha utilizado como fuente fija de captación visual envolvente, ideal para entornos colaborativos o escenarios estáticos. La conexión con el sistema se realizó mediante una biblioteca propia, que trata el flujo de imagen no accesible como webcam convencional. Para integrarlo con la arquitectura general del sistema, se desarrolló un cliente especializado que recibe las imágenes desde el Búho y las reenvía mediante otra biblioteca propia al servidor correspondiente, igualando así el funcionamiento del resto de fuentes.

Dado que el formato original es *dual fisheye*, fue necesario incorporar una rutina de *stitching* y reproyección esférica a equirectangular, seguida de una transformación adicional que, simulando una cámara virtual, genera una vista rectilínea orientada, sobre la que los modelos de análisis de escenas se pueden aplicar sin problema.

5. ALGORITMOS ESTUDIADOS

En este trabajo se han utilizado varios modelos de análisis de escenas basados en arquitecturas *transformer* y orientados al procesamiento multimodal de cuadros, texto y vídeo:

Algoritmos de VQA

- **CLIP** (OpenAI): Modelo pionero en aprendizaje contrastivo a gran escala. Asocia texto e imagen mediante codificadores independientes y genera *embeddings* que se comparan por similitud. No requiere ajuste supervisado, lo que lo hace versátil para tareas como clasificación, búsqueda o VQA. Aquí se ha usado para VQA, es decir, responder preguntas sobre imágenes.
- **ViLT**: Modelo eficiente que fusiona imagen y texto desde las capas iniciales, sin redes convolucionales previas. Su bajo número de parámetros y coste computacional lo hacen ideal para prototipado rápido. En este proyecto se ha empleado para tareas de VQA, con respuestas breves y directas [7].

Algoritmos de descripción de cuadros

- **Florence** (Microsoft): Plataforma visual generalista entrenada con más de 900 millones de pares imagen-texto. Integra numerosas funcionalidades como clasificación, detección de objetos, segmentación, generación de descripciones o reconocimiento de caracteres [8]. Se ha usado como modelo de referencia para generar descripciones.
- **LAVIS** (Salesforce Research): No es un modelo único, sino un entorno unificado para trabajar con modelos preentrenados. Proporciona una API común y permite generar descripciones o responder preguntas [9]. Aquí se ha utilizado también para la generación de descripciones de imágenes.

Algoritmos de descripción de vídeos

- **OpenBMB**: Plataforma centrada en vídeo y razonamiento visual. Ofrece modelos capaces de describir secuencias dinámicas, lo que resulta útil en contextos donde la información visual cambia con el tiempo [10]. En el trabajo se ha usado para generación de descripciones de fragmentos de vídeo, en lugar de cuadros aislados.
- **LLaVA-Next-Video:** Variante evolucionada de LLaVA, combinando LLM y codificador visual para analizar vídeo. Permite razonamiento sobre secuencias completas, identificando el orden de eventos, acciones o cambios relevantes. Se ha aplicado a flujos de vídeo con el objetivo de obtener descripciones narrativas coherentes y contextualizadas [11].

6. EVALUACIÓN DE LOS ALGORITMOS

Dado el elevado número de modelos disponibles para análisis de escenas, se realizó una evaluación exhaustiva con el fin de seleccionar uno representativo de cada categoría funcional: un modelo para VQA, otro para descripción de cuadros y un tercero para descripción de vídeos. La evaluación combinó dos enfoques complementarios:

- Métricas clásicas, basadas en parámetros estadísticos como BLEU, METEOR o CIDEr [12].
- Evaluación con LLM como juez (*LLM-as-a-Judge*), empleando los modelos Deepseek para evaluar los algoritmos de VQA y descripción de cuadros y Qwen para los de descripción de vídeos [12].

El proceso implicó, en primer lugar, la localización y adaptación de conjuntos de datos adecuados para cada tarea. Posteriormente, fue necesario generar nuevas etiquetas con los modelos en evaluación, aplicando nuestros propios algoritmos sobre las mismas imágenes del dataset para permitir la comparación directa.

Los resultados por categoría fueron los siguientes:

- **VQA**: El modelo ViLT obtuvo los mejores resultados tanto en métricas tradicionales como en la evaluación con LLM, lo que motivó su elección final.
- **Descripción de cuadros**: Aunque Florence obtuvo peores puntuaciones estadísticas que LAVIS, sus descripciones eran más completas y detalladas, y fue claramente superior en la evaluación con LLM. Se escogió Florence como opción definitiva.
- **Descripción de vídeos**: Las métricas clásicas no ofrecieron resultados fiables, por lo que se recurrió únicamente al enfoque con LLM. El modelo LLaVA-Next-Video fue el más valorado, siendo por tanto el seleccionado.

Como conclusión, los modelos integrados en los servidores finales del sistema son:

- **ViLT para VQA.**
- **Florence para descripción de cuadros.**
- **LLaVA para descripción de vídeos.**

7. RESULTADOS Y CASOS DE USO

Como se vio, el sistema desarrollado se basa en una arquitectura modular y distribuida que permite la adquisición, análisis, gestión y uso de información visual en tiempo casi real. La Figura 2 correspondiente ilustra el flujo completo, desde la

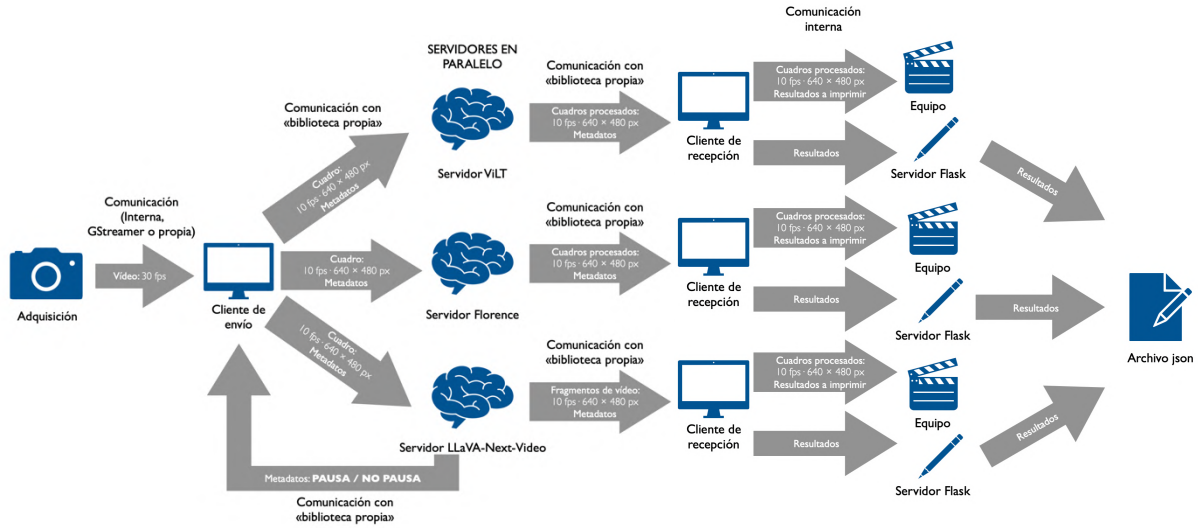


Figura 2. Esquema de la arquitectura cliente-servidor, con los tres modelos elegidos

adquisición del vídeo hasta la actuación final del sistema:

- Un único cliente de envío adquiere vídeo y lo distribuye paralelamente a tres servidores especializados, cada uno ejecutando uno de los modelos seleccionados en el punto anterior:
  - ViLT, que responde preguntas puntuales sobre la escena.
  - Florence, que genera descripciones detalladas cuadro a cuadro
  - LLaVA-Next-Video, que concatena cuadros para construir fragmentos de vídeo coherentes y luego los describe en conjunto
- Cada uno de estos servidores trabaja de forma autónoma, en paralelo, y puede ejecutarse localmente o en máquinas diferentes. Sus resultados se muestran y almacenan por medio de un cliente de recepción asociado, que se encarga también de enviarlos a un servidor Flask, el cual los recoge y escribe en un archivo compartido.
- Tras esto, el sistema de inferencia, basado en un modelo de lenguaje (LLM), en este caso Deepseek (se ha selecciona por la experiencia previa con ella), analiza el contenido del archivo con los resultados. Se distinguen dos modos de lectura:
  - Modo local, para pruebas y depuración.
  - Modo en vivo, que permite leer progresivamente los datos conforme se generan, lo que habilita la ejecución en tiempo casi real.
- Finalmente, el LLM genera una respuesta, instrucción o informe en lenguaje natural en función del contenido recibido. Esa respuesta puede ser leída por un usuario, almacenada o bien ser utilizada como instrucción para un sistema robótico u operativo.

A partir de esta arquitectura funcional se han definido y probado diversos casos de uso que muestran la aplicabilidad del sistema:

Construcción de itinerarios y entornos en sistemas móviles

Un primer caso de uso se centra en la reconstrucción narrativa del recorrido de un robot móvil. Utilizando la secuencia de datos visuales analizados por Florence y LLaVA, el modelo Deepseek genera una descripción continua del entorno recorrido: lugares atravesados, eventos ocurridos y elementos relevantes detectados. Esta capacidad permite construir una traza semántica del desplazamiento del robot, más allá de la mera detección puntual.

Control y navegación autónoma basada en percepción visual

Este caso de uso avanza hacia una integración activa entre visión e inteligencia: el LLM interpreta la descripción generada por los modelos visuales y decide una acción concreta. Esto permite dotar a sistemas móviles o robóticos de una capacidad autónoma de navegación, en la que la acción depende directamente de la comprensión visual del entorno. Se han planteado dos modos:

- Modo asistencial, como ayuda a operadores humanos.
- Modo autónomo, donde el sistema toma decisiones sin intervención externa.

Generación de informes y resúmenes para telepresencia

El sistema puede generar resúmenes automáticos en lenguaje natural sobre una secuencia de vídeo. Esto permite, por ejemplo, documentar lo ocurrido en una reunión, monitorizar entornos remotos, o supervisar actividades en residencias o centros sanitarios. El

LLM interpreta las descripciones visuales y genera un informe compacto y coherente.

Asistente virtual y copiloto para tareas complejas

En entornos donde se siguen procedimientos definidos (como una receta de cocina o un protocolo médico), el sistema puede servir de copiloto inteligente. Tras comparar la ejecución visual de la tarea con un texto estructurado de referencia, el sistema puede detectar si el procedimiento se está siguiendo correctamente, sugerir pasos siguientes o evaluar el desempeño del usuario.

Esta capacidad de comparación estructurada permite una evaluación asincrónica, ampliando la aplicabilidad del sistema a tareas profesionales, educativas e industriales.

8. CONCLUSIONES Y LÍNEAS FUTURAS

Este trabajo ha demostrado la viabilidad técnica y funcional de un sistema distribuido para el análisis avanzado de escenas, basado en inteligencia artificial multimodal y aplicado a entornos dinámicos mediante un robot cuadrúpedo móvil. Se ha logrado integrar distintas técnicas de análisis visual (VQA, descripción de cuadros y vídeos), compararlas rigurosamente mediante métricas objetivas y evaluación con *LLM-as-a-Judge*, y seleccionar las mejores arquitecturas disponibles en cada categoría.

La implementación de una arquitectura cliente-servidor escalable y su posterior combinación con razonamiento en lenguaje natural, facilitando casos de uso diversos: desde la reconstrucción narrativa de recorridos hasta la generación autónoma de informes o instrucciones para navegación. Además, se ha establecido un canal efectivo de comunicación entre dispositivos de adquisición en movimiento y servidores de análisis, consolidando una plataforma robusta para tareas de telepresencia inteligente.

Como líneas futuras, se plantean, entre otras:

- Optimización del rendimiento en entornos reales, especialmente en cuanto a latencia, consumo de recursos y escalabilidad.
- Entrenamiento o ajuste fino de modelos adaptados a contextos específicos, como entornos industriales, agrícolas o sanitarios.
- Integración bidireccional con sistemas robóticos, permitiendo que las decisiones del modelo se traduzcan directamente en acciones del robot.

En conjunto, este proyecto constituye una base sólida para la construcción de sistemas inteligentes de percepción y acción, combinando visión, lenguaje e interacción en tiempo real.

9. REFERENCIAS

1. N. Ahmed, M. Wahed y N. C. Thompson, "The growing influence of industry in AI research", Science, vol. 379, no. 6635, pp. 884-886, mar. 2023, doi: 10.1126/science.ade2420. [Online]. Disponible en: https://www.science.org/doi/10.1126/science.ade2420
2. D. Man y R. Olchawa, "The Possibilities of Using BCI Technology in Biomedical Engineering", en Biomedical Engineering and Neuroscience, W. P. Hunek y S. Paszkiel, Opole, Eds., Polonia: Springer, pp. 30-37. [Online]. Disponible en: https://www.researchgate.net/publication/322958198_The_Possibilities_of_Using_BCI_Technology_in_Biomedical_Engineering
3. R. C. Arkin, Behavior-Based Robotics. Cambridge, MA, EEUU: MIT Press, 1998. [Online]. Disponible en: http://www.coep.ufrj.br/~ramon/COE-841/robotics/book%201998%20-%20Behavior-Based%20Robotics%20-%20Arkin.pdf
4. D. Vernon, Artificial cognitive systems: A primer, Cambrige, MA, EEUU: MIT Press, 2014. [Online]. Disponible en: http://vernon.eu/publications/14_Vernon_Artificial_Cognitive_Systems_ Preamble.pdf
5. "About Go2". Unitree Documentation Center. Visitado: 16 jun. 2025. [Online]. Disponible en: https://support.unitree.com/home/en/developer
6. M. Orduna, P. Pérez, J. Gutiérrez y N. García, "Methodology to Assess Quality, Presence, Empathy, Attitude, and Attention in 360-degree Videos for Immersive Communications", IEEE Transactions on Affective Computing, vol. 14, no. 3, pp. 2375-2388, 1 July-Sept. 2023, doi: 10.1109/TAFFC.2022.3149162. [Online]. Disponible en: 3.2.https://ieeexplore.ieee.org/abstract/document/9713751
7. W. Kim, B. Son y I. Kim, "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision", en Proceedings of the 38th International Conference on Machine Learning, 18-24 jul. 2021, doi: 10.48550/arXiv.2102.03334. [Online]. Disponible en: https://arxiv.org/pdf/2102.03334
8. L. Yuan et al. "Florence: A New Foundation Model for Computer Vision", arXiv:2111.11432. [Online]. Disponible en: https://arxiv.org/pdf/2111.11432
9. D. Li et al., "LAVIS: A One-stop Library for Language-Vision Intelligence", en Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), D. Bollegala, R. Huang y A. Ritter, Eds., Toronto, Canadá, jul. 2023, pp. 31-41, doi: 10.18653/v1/2023.acl-demo.3. [Online]. Disponible en: https://aclanthology.org/ 2023.acl-demo.3.pdf
10. OpenBMB, "OpenBMB: Big Models for Everyone", Medium, 18 nov. 2022. Visitado: 16 jun. 2025. [Online]. Disponible en: https://medium.com/@openbmb/openbmb-big-models-for-everyone-f5a812c4ad43
11. F. Li et al., "LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models", 2024, arXiv.2407.07895.
12. H. Sharma y A. S. Jalal, "A survey of methods, datasets and evaluation metrics for visual question answering", Image and Vision Computing, vol. 116, dec. 2021, Art. no. 104327, doi: 10.1016/j.imavis.2021.104327. [Online]. Disponible en: https://doi.org/10.1016/j.imavis. 2021.104327

Abstract

SCENE ANALYSIS FOR ROBOTICS AND IMMERSIVE TELEPRESENCE APPLICATIONS

Juan Modesto Espinosa Bogas

juanmodesto.espi@gmail.com

Keywords: *Scene analysis; robotics; immersive telepresence; LLM.*

1. INTRODUCTION

Artificial Intelligence (AI) has evolved since its origins in the 1950s to become a key technology across multiple sectors, thanks to deep learning and increased computational power [1]. In this context, one of its most advanced recent applications is scene analysis, which, unlike simple object detection, not only identifies objects in an image or video, but also understands their spatial, functional, and contextual relationships, similar to how humans perceive scenes [2]. The main functionalities explored in this project are:

- VQA (Visual Question Answering): answering questions based on a single frame or image.
- Captioning: generating descriptions, either from a single image or from a video segment.

This Master's Thesis (TFM) has been carried out at Nokia Spain S.A.'s XRLab, a laboratory specialized in extended reality (XR) technologies and their integration with physical systems such as robots—complex environments where scene analysis can be especially valuable. The main goal of the project is to research, implement, and evaluate scene analysis algorithms based on AI, integrated with advanced language models (LLMs), to provide systems with meaningful visual understanding of their surroundings, enhancing autonomy, human-machine interaction, and real-time decision-making. The work also addresses current challenges of these technologies, such as model traceability, computational cost, and lack of robustness in dynamic scenarios.

2. OBJECTIVES

The main objectives of the project are:

- To study advanced AI-based scene analysis techniques, including video, image, and multimodal architectures.
- To integrate a scene analysis system into a zoomorphic robot to interpret its visual environment.
- To evaluate and compare models using objective metrics and LLM-as-a-Judge-based tools.
- To design a client-server architecture for remote image processing.
- To establish a communication channel between telepresence systems and analysis servers.
- To unify visual results with natural language reasoning through LLM models.

3. PROPOSED SOLUTION

The technical solution developed in this Master's Thesis is based on a distributed client-server architecture that enables the integration of multiple video sources, parallel processing through advanced scene analysis algorithms, and a cognitive reasoning layer powered by LLMs. This approach follows the conceptual scheme sense-think-act, widely used in cognitive robotics due to its capacity to structure autonomous systems [3], and although not as commonly applied in telepresence systems, it proves coherent and promising [4]. In this case:

- The “sense” layer corresponds to the visual perception of the environment, captured by sensors such as cameras or XR devices and processed through scene analysis algorithms.
- The “think” layer represents the cognitive component, where an LLM interprets the visual results, contextualizes the information, and generates descriptions, inferences, or actions.

- The "act" layer refers to the modules that display, store, or execute decisions, whether in terms of visualization, interaction, or commands.

The proposed distributed architecture is composed of the following blocks, schematically represented in Figure 1:

- **Sending clients:** These collect video streams from various physical devices capturing real-time footage and send it to the servers at a resolution of 640 × 480 px and a frame rate of 3 fps, optimizing bandwidth and latency without significantly sacrificing quality.
- **Analysis servers:** Several servers operate in parallel, each running a different scene analysis algorithm. These models independently process incoming video and forward the processed frames and results to the next clients.
- **Receiving clients:** Modules responsible for receiving the results from the analysis servers. These clients allow visualization of the processed frames, print the results, and record them in a structured file, which will serve as the basis for subsequent reasoning.

Figure 1. Basic diagram of the proposed architecture

Additionally, the system includes a:

- **Reasoning layer (LLM):** A large language model reads the structured file and generates various responses depending on the context or use case, from detailed descriptions to summaries or operational decisions. This layer is essential to transform purely visual data into understandable and useful knowledge.

4. CAPTURE DEVICES

Efficient capture of the visual environment is an essential component of any system based on scene analysis. This project used two complementary platforms:

Unitree Go2 Robot

The Go2 is a bio-inspired quadruped robot (dog-like) developed by the Chinese company Unitree Robotics. Its form provides excellent adaptability to uneven terrain. Equipped with 12 degrees of freedom, integrated sensors, real-time script execution capability, and ROS connectivity, it is an ideal solution for inspection, exploration, or assistance tasks in real-world environments [5].

In this project, the Go2 was used as a mobile video acquisition platform, enabling performance evaluation of scene analysis in dynamic and semi-structured situations.

Integration with the Go2 required overcoming significant technical barriers. Although options such as the official ROS 2 SDK were explored, they proved unviable due to incompatibilities or instability. The chosen solution was to use GStreamer as an intermediate tool: the video stream captured by the robot was redirected via UDP to a local address, enabling smooth and real-time reception on the analysis system.

The Owl

The Owl is a portable immersive telepresence system developed at Nokia's XRLab, designed to capture 360° video and allow remote user participation via virtual reality headsets. It overcomes the limitations of traditional video conferencing, offering a much richer experience in terms of spatial context, non-verbal communication, and sensory immersion [6]. It consists of two fisheye cameras mounted on a lightweight, 3D-printed structure, with a control unit that encodes and transmits video in real time. The system also includes bidirectional audio and multimodal connectivity, allowing use both indoors and outdoors.

In this project, the Owl was used as a fixed source of immersive visual capture, ideal for collaborative environments or static scenarios. The connection to the system was made via a proprietary library, which handles the video stream not accessible through conventional webcam interfaces. To integrate it with the overall system architecture, a specialized client was developed to receive the images from the Owl and forward them via another proprietary library to the corresponding server, thus aligning its behavior with the rest of the sources.

Since the original format is dual-fisheye, a stitching and spherical reprojection routine to equirectangular format was required, followed by an additional transformation that simulates a virtual camera to generate a rectilinear and oriented view, upon which scene analysis models can be effectively applied.

5. STUDIED ALGORITHMS

This work has used several scene analysis models based on transformer architectures, oriented toward multimodal processing of images, text, and video:

VQA Algorithms

- **CLIP** (OpenAI): A pioneering model in large-scale contrastive learning. It associates text and images using independent encoders and generates

embeddings that are compared based on similarity. It does not require supervised fine-tuning, making it versatile for tasks such as classification, search, or VQA. In this project, it has been used for VQA, i.e., answering questions about images.

- **ViLT**: An efficient model that fuses image and text from the early layers, without relying on convolutional networks. Its low number of parameters and reduced computational cost make it ideal for rapid prototyping. In this project, it was used for VQA tasks, with brief and direct responses [7].

Image Captioning Algorithms

- **Florence** (Microsoft): A general-purpose visual platform trained on over 900 million image-text pairs. It integrates numerous functionalities such as classification, object detection, segmentation, caption generation, or character recognition [8]. It has been used as the reference model for generating image captions.
- **LAVIS** (Salesforce Research): Not a single model but a unified framework to work with pretrained models. It provides a common API and allows for caption generation and question answering [9]. In this project, it has also been used for generating image captions.

Video Captioning Algorithms

- **OpenBMB**: A platform focused on video and visual reasoning. It offers models capable of describing dynamic sequences, which is useful in contexts where visual information changes over time [10]. In this work, it was used for generating captions from video segments instead of isolated frames.
- **LLaVA-Next-Video**: An evolved variant of LLaVA, combining an LLM and a visual encoder to analyze video. It enables reasoning over complete sequences, identifying event order, actions, or relevant changes. It has been applied to video streams to obtain coherent and contextualized narrative descriptions [11].

6. ALGORITHM EVALUATION

Given the large number of models available for scene analysis, a comprehensive evaluation was carried out to select one representative model for each functional category: one model for VQA, another for image captioning, and a third for video captioning. The evaluation combined two complementary approaches:

- Classical metrics, based on statistical parameters such as BLEU, METEOR, or CIDEr [12].
- LLM-as-a-Judge evaluation, using Deepseek models for assessing the VQA and image captioning algorithms, and Qwen for video captioning models [12].

The process involved, first, identifying and adapting suitable datasets for each task. Then, it was necessary to generate new labels with the models under evaluation, applying our own algorithms to the same dataset images to allow for direct comparison.

The results by category were as follows:

- **VQA**: The ViLT model achieved the best results in both traditional metrics and LLM-based evaluation, which led to its final selection.
- **Image captioning**: Although Florence scored lower than LAVIS in statistical metrics, its captions were more complete and detailed, and it clearly outperformed LAVIS in the LLM evaluation. Florence was selected as the final option.
- **Video captioning**: Classical metrics did not provide reliable results, so the evaluation relied solely on the LLM approach. The LLaVA-Next-Video model received the highest scores and was therefore selected.

In conclusion, the models integrated into the system's final servers are:

- **ViLT for VQA**
- **Florence for image captioning**
- **LLaVA for video captioning**

7. RESULTS AND USE CASES

As previously described, the developed system is based on a modular and distributed architecture that enables the acquisition, analysis, management, and utilization of visual information in near real-time. Figure 2 illustrates the complete flow, from video acquisition to the final system response:

- A single sending client acquires video and distributes it in parallel to three specialized servers, each running one of the models selected in the previous section:
  - ViLT, which answers specific questions about the scene.
  - Florence, which generates detailed frame-by-frame descriptions.
  - LLaVA-Next-Video, which concatenates frames to build coherent video segments and then describes them as a whole.

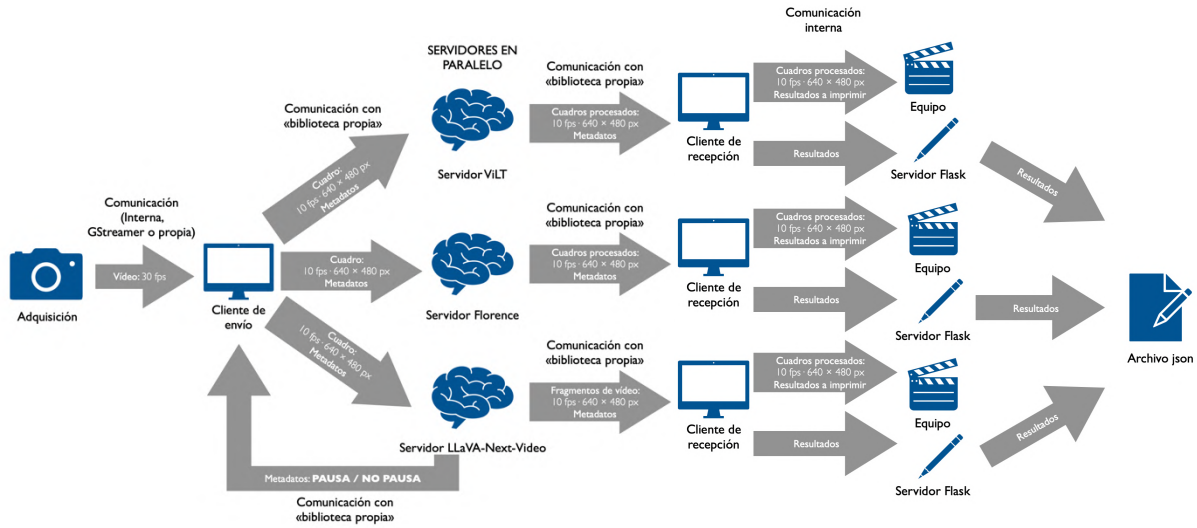


Figure 2. Diagram with the client-server architecture, with the three chosen models

- Each of these servers operates autonomously and in parallel and can run locally or on different machines. Their results are displayed and stored through a dedicated receiving client, which also forwards them to a Flask server that collects and writes them to a shared file.
- After this, the inference system, based on a language model (LLM)—in this case Deepseek (selected due to prior experience with it)—analyzes the content of the results file. Two reading modes are available:
  - Local mode, for testing and debugging.
  - Live mode, which allows progressive reading of data as they are generated, enabling near real-time operation.
- Finally, the LLM generates a response, instruction, or report in natural language based on the content received. This response can be read by a user, stored, or used as an instruction for a robotic or operational system.

Based on this functional architecture, several use cases have been defined and tested, demonstrating the system's applicability:

Route construction and environment mapping in mobile systems

A first use case focuses on the narrative reconstruction of a mobile robot's path. Using the sequence of visual data analyzed by Florence and LLaVA, the Deepseek model generates a continuous description of the traversed environment: places passed through, events that occurred, and relevant elements detected. This capability allows for the construction of a semantic trace of the robot's movement, beyond simple spot detection.

Autonomous navigation and control based on visual perception

This use case moves toward active integration between vision and intelligence: the LLM interprets the description generated by the visual models and decides on a specific action. This enables mobile or robotic systems to gain autonomous navigation capabilities, where actions are directly driven by the visual understanding of the environment. Two operating modes were defined:

Assistance mode, to aid human operators.

Autonomous mode, where the system makes decisions without external input.

Report and summary generation for telepresence

The system can automatically generate natural language summaries from a video sequence. This allows, for example, documenting what occurred during a meeting, monitoring remote environments, or supervising activities in care homes or healthcare centers. The LLM interprets the visual descriptions and produces a compact and coherent report.

Virtual assistant and co-pilot for complex tasks

In environments that follow predefined procedures (such as a cooking recipe or a medical protocol), the system can act as an intelligent co-pilot. By comparing the visual execution of the task with a structured reference text, the system can detect whether the procedure is being followed correctly, suggest next steps, or evaluate the user's performance.

This structured comparison capability enables asynchronous assessment, expanding the applicability of the system to professional, educational, and industrial tasks.

8. CONCLUSIONS AND FUTURE WORK

This work has demonstrated the technical and functional feasibility of a distributed system for advanced scene analysis, based on multimodal artificial intelligence and applied to dynamic environments through a mobile quadruped robot. Various visual analysis techniques (VQA, image and video captioning) have been successfully integrated, rigorously compared using objective metrics and LLM-as-a-Judge evaluation, and the best available architectures in each category have been selected.

The implementation of a scalable client-server architecture, combined with natural language reasoning, has enabled a wide range of use cases—from narrative reconstruction of trajectories to autonomous generation of reports or navigation instructions. Furthermore, an effective communication channel has been established between mobile acquisition devices and analysis servers, consolidating a robust platform for intelligent telepresence tasks.

Future work may include, among other lines:

- Performance optimization in real-world environments, especially regarding latency, resource consumption, and scalability.
- Training or fine-tuning models adapted to specific contexts such as industrial, agricultural, or healthcare settings.
- Bidirectional integration with robotic systems, allowing model decisions to be directly translated into robotic actions.

Overall, this project provides a solid foundation for the construction of intelligent perception and action systems, combining vision, language, and real-time interaction.

9. REFERENCES

1. N. Ahmed, M. Wahed y N. C. Thompson, "The growing influence of industry in AI research", Science, vol. 379, no. 6635, pp. 884-886, mar. 2023, doi: 10.1126/science.ade2420. [Online]. Disponible en: https://www.science.org/doi/10.1126/science.ade2420
2. D. Man y R. Olchawa, "The Possibilities of Using BCI Technology in Biomedical Engineering", en Biomedical Engineering and Neuroscience, W. P. Hunek y S. Paszkiel, Opole, Eds., Polonia: Springer, pp. 30-37. [Online]. Disponible en: https://www.researchgate.net/publication/322958198_The_Possibilities_of_Using_BCI_Technology_in_Biomedical_Engineering
3. R. C. Arkin, Behavior-Based Robotics. Cambridge, MA, EEUU: MIT Press, 1998. [Online]. Disponible en: http://www.coep.ufrj.br/~ramon/COE-841/robotics/book%201998%20-%20Behavior-Based%20Robotics%20-%20Arkin.pdf
4. D. Vernon, Artificial cognitive systems: A primer, Cambrige, MA, EEUU: MIT Press, 2014. [Online]. Disponible en: http://vernon.eu/publications/14_Vernon_Artificial_Cognitive_Systems_ Preamble.pdf
5. "About Go2". Unitree Documentation Center. Visitado: 16 jun. 2025. [Online]. Disponible en: https://support.unitree.com/home/en/developer
6. M. Orduna, P. Pérez, J. Gutiérrez y N. García, "Methodology to Assess Quality, Presence, Empathy, Attitude, and Attention in 360-degree Videos for Immersive Communications", IEEE Transactions on Affective Computing, vol. 14, no. 3, pp. 2375-2388, 1 July-Sept. 2023, doi: 10.1109/TAFFC.2022.3149162. [Online]. Disponible en: 3.2.https://ieeexplore.ieee.org/abstract/document/9713751
7. W. Kim, B. Son y I. Kim, "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision", en Proceedings of the 38th International Conference on Machine Learning, 18-24 jul. 2021, doi: 10.48550/arXiv.2102.03334. [Online]. Disponible en: https://arxiv.org/pdf/2102.03334
8. L. Yuan et al. "Florence: A New Foundation Model for Computer Vision", arXiv:2111.11432. [Online]. Disponible en: https://arxiv.org/pdf/2111.11432
9. D. Li et al., "LAVIS: A One-stop Library for Language-Vision Intelligence", en Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), D. Bollegala, R. Huang y A. Ritter, Eds., Toronto, Canadá, jul. 2023, pp. 31-41, doi: 10.18653/v1/2023.acl-demo.3. [Online]. Disponible en: https://aclanthology.org/ 2023.acl-demo.3.pdf
10. OpenBMB, "OpenBMB: Big Models for Everyone", Medium, 18 nov. 2022. Visitado: 16 jun. 2025. [Online]. Disponible en: https://medium.com/@openbmb/openbmb-big-models-for-everyone-f5a812c4ad43
11. F. Li et al., "LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models", 2024, arXiv.2407.07895.
12. H. Sharma y A. S. Jalal, "A survey of methods, datasets and evaluation metrics for visual question answering", Image and Vision Computing, vol. 116, dec. 2021, Art. no. 104327, doi: 10.1016/j.imavis.2021.104327. [Online]. Disponible en: https://doi.org/10.1016/j.imavis. 2021.104327

Siglas y acrónimos

NOTA: Las siglas marcadas con * se corresponden con nombres específicos de modelos tratados durante el proyecto. En la mayoría de los casos, al considerarlos «nombres propios» no se ha incluido la traducción de la sigla.

	AR	*Augmented Reality* | Realidad Aumentada
*	**BMB**	[*Open*] *Big Model Base*
	BLEU	*BiLingual Evaluation Understudy* | Estudio de Evaluación Bilingüe
*	**BLIP**	*Bootstrapping Language-Image Pre-training*
	CIDEr	Consensus-based Image Description Evaluation | *Evaluación de la descripción de imágenes basada en el consenso*
*	**CLIP**	*Contrastive Language-Image Pretraining*
	CNN	*Convolutional Neural Network* | Redes Neuronales Convolucionales
*	**DeiT**	*Data efficient image Transformers*
	IA	Inteligencia Artificial
	IDE	*Integrated Development Environment* | Entorno de Desarrollo Integrado
	IMU	*Inertial Measurement Unit* | Unidad de Medición Inercial
	FPS	Fotogramas por segundo
	GOFAI	*Good Old-Fashioned Artificial Intelligence* | Inteligencia Artificial Buena a la Antigua
*	**GPT**	*Generative Pretrained Transformer*
	GPU	*Graphics Processing Unit* | Unidad de Procesamiento Gráfico
*	**LAVIS**	*Language and Vision Interface for Systems*
*	**LLaMA**	*Large Language Model Architecture*
*	**LLaVA**	*Large Language and Vision Assistant*
	LLM	*Large Language Model* | Modelo de Lenguaje Avanzado
*	**MAE**	*Masked Autoencoder*
	METEOR	*Metric for Evaluation of Translation with Explicit ORdering* | Métrica para evaluar la traducción con orden explícito

	MLP	*Multilayer Perceptron* | Perceptrón Multicapa
	MR	*Mixed Reality* | Realidad Mixta
	NLP	*Natural Language Processing* | Procesamiento de Lenguaje Natural
	ODS	Objetivos de Desarrollo Sostenible
*	**OpenCV**	*Open Source Computer Vision Library*
*	**PaLM**	*Pathways Language Model*
	RGPD	Reglamento General de Protección de Datos
	ROUGE	*Recall-Oriented Understudy for Gisting Evaluation* | Estudio para la evaluación de la esencia orientado al recall
	SoC	*System on a Chip* | Sistema en chip
	TFM	Trabajo Fin de Máster
	TPU	*Tensor Processing Unit* | Unidad de Procesamiento Tensorial
	VQA	*Visual Question Answering* | Respuesta a preguntas visuales
*	**ViT**	*Vision Transformer*
*	**ViLT**	*Vision-and-Language Transformer*
	VPN	*Virtual Private Network* | Red Privada Virtual
	VR	*Virtual Reality* | Realidad virtual
	WUPS	*Wu-Palmer Similarity* | Semejanza de Wu-Palmer
	XR	*eXtended Reality* | Realidad Extendida
*	**YOLO**	*You Only Look Once*

Índice de figuras

Índice de tablas

ANÁLISIS DE ESCENAS PARA ROBÓTICA Y APLICACIONES DE TELEPRESENCIA INMERSIVA

Capítulo 1

INTRODUCCIÓN

1.1. Interés del trabajo

La inteligencia artificial (IA) es una disciplina que ha tenido un «largo» recorrido en la historia de la ciencia y la ingeniería. A pesar de que su expansión se ha acelerado en la última década, sus fundamentos teóricos se remontan a mediados del siglo XX. Ya en 1950, Alan Turing planteaba en su célebre artículo *Computing Machinery and Intelligence* [1] la posibilidad de que una máquina pudiera pensar, y apenas seis años después, se acuñaba el término «inteligencia artificial» durante la Conferencia de Dartmouth de 1956, considerada por muchos el nacimiento oficial del campo [2].

Desde sus inicios, la IA ha estado ligada a una aspiración ambiciosa: dotar a las máquinas de capacidades que tradicionalmente consideramos humanas, como razonar, aprender o percibir. Las primeras décadas de investigación se centraron en modelos simbólicos y reglas lógicas, lo que hoy se denomina IA simbólica o *good old-fashioned artificial intelligence* (buena inteligencia artificial a la antigua, más conocida por la sigla GOFAI) [3]. Surgieron programas como ELIZA (1966) [4], un pionero *chatbot*, o sistemas expertos que aplicaban reglas lógicas para diagnosticar enfermedades. Sin embargo, estas aproximaciones se mostraron frágiles ante la complejidad y variabilidad del mundo real.

En paralelo, aparecieron enfoques estadísticos e inspirados en el cerebro humano, como las redes neuronales, cuyo origen se remonta a mediados del siglo XX. No obstante, la limitada capacidad computacional de la época y la falta de datos dificultaron su desarrollo, y a partir de los años 70 y 80, la IA entró en lo que se conoce como sus primeros «inviernos», periodos marcados por el desencanto, el estancamiento de los avances y la reducción drástica de la financiación [5]. Este ciclo se repitió en los 90, con expectativas no cumplidas y una creciente desconfianza hacia las promesas de la IA.

La situación comenzó a cambiar radicalmente a partir de la década de 2010, con la aparición del *deep learning* (en inglés, «aprendizaje profundo») y el desarrollo de la mejora de la capacidad de procesamiento, dos hitos de los que se hablará más adelante en este trabajo. Hechos como la victoria del programa AlphaGo frente al campeón mundial de go (un juego asiático similar a las damas) en 2016 [6] o el desarrollo de modelos generativos multimodales en los últimos años reflejan esta transformación.

La IA actual no es una disciplina homogénea, sino un conjunto de técnicas que incluyen métodos estadísticos, modelos probabilísticos, lógica difusa, aprendizaje automático (también conocido por el inglés *machine learning*) y aprendizaje profundo, muchas de las cuales se basan en fundamentos matemáticos y estadísticos desarrollados a lo largo del siglo XX. Esta diversidad metodológica, unida a los avances recientes, ha permitido que en los últimos años la IA se haya consolidado como uno de los

pilares tecnológicos de mayor impacto en el desarrollo industrial y científico. Su aplicación está transformando progresivamente la forma en que las organizaciones producen, interactúan, gestionan información y toman decisiones. Este proceso no solo afecta a sectores tradicionalmente tecnológicos, sino que se extiende transversalmente a campos como la salud, la educación, la logística o la automatización industrial, redefiniendo los límites de la ingeniería tal y como la hemos concebido hasta ahora [7], [8].

En este contexto, la principal aplicación de los algoritmos basados en IA es el desarrollo de máquinas o equipos que sean capaces de desarrollar una cierta «inteligencia», de manera que puedan funcionar de forma autónoma, o, al menos, con cierta capacidad de resolución de problemas por sí mismos, haciendo que la interacción con los operadores sea cada vez menor. Dentro de este marco, uno de los elementos más relevantes en la evolución de la IA es la capacidad de dotar a los sistemas de una percepción visual avanzada, similar a la que poseen los seres humanos. Esta necesidad se justifica no solo por el hecho de que la mayoría de la información que procesamos como individuos es visual (más del 80% según estudios neurológicos [9]), sino porque muchas de las tareas que requieren automatización en entornos reales dependen precisamente de una comprensión eficiente del entorno visual.

El análisis de escenas representa, en este sentido, un avance sustancial respecto a las técnicas tradicionales de visión por ordenador. No se limita a la detección de objetos o su clasificación, sino que busca identificar relaciones espaciales, funciones contextuales y crear estructuras dentro de un cuadro o una secuencia de vídeo [10]. El presente Trabajo Fin de Máster (TFM) se sitúa, por tanto, en el marco de una línea tecnológica de especial relevancia, alineada con las tendencias más actuales en inteligencia artificial, visión por ordenador y automatización avanzada. Su propósito es aportar una contribución práctica y aplicada al desarrollo y evaluación de sistemas de análisis de escenas, con una aproximación integradora que combina teoría, implementación técnica y análisis crítico en un entorno realista.

1.2. Motivación del TFM

Determinar qué es realmente la ingeniería no es una tarea sencilla. Existen innumerables definiciones en las que aparecen, de forma recurrente, conceptos como la creatividad, el diseño o la invención [11], [12]. Sin embargo, hay una idea que, aunque se intuye, no suele figurar con la frecuencia que merece: la búsqueda de un mundo más sencillo. Poner al servicio de la sociedad la creatividad y la inquietud por el conocimiento que caracterizan a los ingenieros es algo que rara vez se menciona, pero sin lo cual todo el esfuerzo técnico pierde su sentido.

Esa, y no otra, puede considerarse la principal motivación de este Trabajo Fin de Máster: que las aportaciones realizadas al campo de la inteligencia artificial, y, más concretamente, al análisis de escenas, contribuyan al avance de estas disciplinas en su permanente búsqueda de la excelencia. Que fomenten la interacción y colaboración con los usuarios, y que, en última instancia, puedan suponer una mejora tangible en la vida de las personas.

Cuesta no imaginar la infinidad de aplicaciones que podría tener un robot dotado de capacidades próximas al sentido de la vista, o un sistema de telepresencia que permita al usuario experimentar una verdadera sensación de «teletransporte», como la mejora de la interacción entre máquinas y operarios, asistencia a personas con movilidad reducida, colaboración remota desde cualquier parte del mundo, tecnologías adaptadas a necesidades específicas... En definitiva, un campo de enorme potencial que ya está en desarrollo y que sin duda vivirá una gran evolución en los próximos años, dando lugar a nuevas aplicaciones aún por descubrir.

1.3. Contexto del proyecto

Este Trabajo de Fin de Máster se ha desarrollado en paralelo a la realización de prácticas en el XRLab de Nokia Spain S.A., un departamento de innovación especializado en el desarrollo e integración de tecnologías inmersivas. Este laboratorio forma parte del ecosistema de investigación aplicada de Nokia y constituye una de sus unidades de experimentación más activas en el ámbito de la realidad extendida, así como en la aplicación de estas tecnologías en escenarios de comunicación avanzada, visualización remota e interacción háptica.

El término XRLab no es casual: hace referencia directa a su especialización en tecnologías XR (del inglés *eXtended Reality*, realidad extendida), un paraguas conceptual que engloba las realidades virtual (VR, por el inglés *Virtual Reality*), aumentada (AR, por *Augmented Reality*) y mixta (MR, por *Mixed Reality*) [13]. El laboratorio no solo investiga estas tecnologías de forma aislada, sino que trabaja en su integración con sistemas físicos, como plataformas robóticas o entornos de red 5G/6G, para explorar soluciones aplicables a contextos reales de trabajo, salud, formación o exploración remota. Entre los retos planteados, uno de los más recurrentes es el de dotar a estas plataformas de una percepción más rica y significativa del entorno, lo que plantea una pregunta central: ¿cómo puede la inteligencia artificial ayudar a que estos sistemas no solo plasmen el entorno, sino que lo entiendan?

Para comprender adecuadamente el marco donde se inscribe este TFM, conviene detenerse en una definición precisa de los conceptos de realidad virtual aumentada y mixta. La realidad virtual consiste en la generación de un entorno digital completamente simulado, donde el usuario se introduce mediante visores o sistemas envolventes, y que le permite interactuar con objetos virtuales como si estuviera presente físicamente. Este entorno es independiente del mundo real y requiere una inmersión total del usuario. En contraste, la AR superpone elementos digitales sobre el entorno físico percibido, normalmente a través de cámaras o dispositivos móviles, enriqueciendo la percepción sin sustituirla. La realidad mixta, a su vez, permite una interacción fluida entre elementos virtuales y reales, integrándolos en un espacio compartido donde ambos influyen entre sí en tiempo real [14].

Como se ha mencionado, la realidad extendida es el término que engloba a todas estas tecnologías y sus combinaciones posibles. En contextos como el del XRLab, el uso de XR no se limita a experiencias de entretenimiento o simulación, sino que se orienta hacia soluciones industriales, médicas y robóticas. De hecho, la evolución reciente de estas tecnologías ha propiciado su aplicación a procesos altamente complejos, como la asistencia remota en intervenciones quirúrgicas, la formación avanzada en entornos de riesgo controlado o la gestión y control de sistemas robóticos distribuidos, como se verá más adelante [15], [16]. Estas aplicaciones, sin embargo, requieren no solo una transmisión visual fiel, sino una comprensión más profunda del entorno que facilite la toma de decisiones a distancia o el comportamiento autónomo del sistema.

En paralelo al avance de las tecnologías XR, el concepto de telepresencia inmersiva ha adquirido una relevancia creciente en numerosos ámbitos técnico-operativos. Esta noción no se limita a permitir la observación de un entorno remoto, sino que persigue recrear la experiencia sensorial completa de estar físicamente presente en otro lugar, incorporando tanto la percepción visual y espacial como la posibilidad de interactuar con el entorno en tiempo real.

Así, mediante dispositivos como cámaras estereoscópicas, micrófonos espaciales, sensores de profundidad, interfaces hápticas o cascos de realidad virtual, se busca generar una representación continua y coherente del espacio remoto, ofreciendo al usuario una capacidad de interpretación y acción que supera con creces la de una simple transmisión de vídeo [17]. Estas tecnologías no actúan de forma

aislada, sino que se integran en sistemas distribuidos donde la baja latencia, la alta fidelidad visual y la capacidad de respuesta en tiempo real resultan esenciales.

La telepresencia inmersiva se está consolidando como solución estratégica en entornos donde el acceso directo resulta complejo, peligroso o inviable: operaciones en centrales industriales, supervisión de instalaciones críticas, inspección de infraestructuras, asistencia médica remota o exploración subacuática, entre otros. En todos estos casos, la capacidad de observar, interpretar y operar a distancia, con la misma precisión que si se estuviera físicamente presente, representa un cambio de paradigma en la relación entre ser humano y entorno operativo.

Un caso representativo lo constituye la conducción autónoma teleasistida, también trabajada en el XRLab. En ella se emplean sistemas de visión avanzada para que un operador pueda asumir el control de un vehículo en situaciones de emergencia o decisión crítica, todo ello desde una ubicación remota. Aunque vinculado habitualmente a la automatización, este enfoque responde en esencia a una lógica de telepresencia inmersiva, donde el factor humano conserva un rol decisivo a través de interfaces sensoriales enriquecidas.

Para que estos entornos sean verdaderamente operativos, es imprescindible estructurar semánticamente la información visual ofrecida al usuario, es decir, que no se limite a mostrar imágenes, sino que estas reflejen una interpretación significativa del espacio: reconocer objetos, entender relaciones espaciales, detectar anomalías o anticipar movimientos. Esto conduce directamente al análisis de escenas, eje fundamental del presente trabajo.

Antes de abordar ese importante concepto, conviene introducir el dispositivo físico a través del cual se materializa la experiencia de telepresencia inmersiva que aquí se plantea: el sistema Búho, que representa una solución innovadora que permite a los usuarios remotos experimentar una sensación de presencia en un entorno físico distante. Este sistema combina tecnologías de captura de video en 360 grados y audio espacial para ofrecer una experiencia inmersiva y realista.

El Búho está diseñado para capturar el entorno físico mediante cámaras de alta resolución que generan una visión panorámica completa. Simultáneamente, un conjunto de micrófonos direccionales recoge el sonido ambiente, preservando la direccionalidad y profundidad auditiva. Esta combinación permite a los usuarios remotos percibir tanto el vídeo como el sonido del entorno de manera coherente y envolvente [18].

La interacción con el sistema se realiza a través de dispositivos de realidad virtual, como las gafas Oculus Quest, o mediante teléfonos y tabletas. Los usuarios pueden controlar su punto de vista en el entorno remoto utilizando gestos táctiles. Además, el sistema permite la representación de los usuarios remotos mediante avatares o ventanas de video superpuestas, facilitando la comunicación bidireccional con los participantes presentes en el entorno físico.

El Búho ha sido implementado en diversos contextos, incluyendo conferencias híbridas y entornos colaborativos, demostrando su capacidad para integrar a participantes remotos en actividades presenciales de manera efectiva [19]. Su diseño modular y su compatibilidad puede convertirlo en una solución accesible para aplicaciones que requieren una experiencia de telepresencia inmersiva.

Como se introdujo antes, también es necesario definir brevemente el análisis de escenas, una rama avanzada de la visión por ordenador que busca dotar a los sistemas artificiales de una comprensión semántica del entorno visual.

La visión por ordenador, tal y como se ha desarrollado desde los años 80, se ha centrado tradicionalmente en arquitecturas diseñadas para resolver tareas específicas y acotadas. Estas tareas,

como la detección de bordes, la clasificación supervisada o la detección de objetos, se limitan a identificar qué aparece en una imagen o dónde se encuentra, y aunque son componentes esenciales en cualquier algoritmo, no permiten que un sistema interprete una escena de manera completa [20]. Así, los modelos más básicos pueden detectar con gran precisión la presencia de un coche, una persona o una señal de tráfico, pero no disponen de mecanismos integrados para inferir si esa persona está cruzando una calle, si el coche está frenando ante un obstáculo o si ambos interactúan en una situación potencialmente crítica. Este nivel de comprensión requiere incorporar información relacional, espacial, temporal y contextual, que excede las capacidades de los clasificadores tradicionales.

El análisis de escenas se posiciona, por tanto, en una capa superior del razonamiento visual, extrayendo no solo objetos, sino también las relaciones espaciales, funcionales y contextuales que existen entre ellos. Esta distinción es crítica en entornos donde la capacidad de interpretación debe ser robusta, adaptativa y rápida, como ocurre en robótica móvil, vehículos autónomos, sistemas de inspección automática o plataformas de asistencia remota [21].

Este tipo de análisis permite pasar de una percepción fragmentada a una representación estructurada de la escena, en la que cada objeto, acción o disposición adquiere significado dentro de un marco interpretativo. Por ejemplo, no se trata solo de reconocer que hay una persona y una silla, sino de inferir que la persona está sentada, que la silla forma parte de una zona de descanso, o que existe una interacción entre ambos elementos que puede ser relevante para una determinada tarea.

Desde el punto de vista técnico, el análisis de escenas combina capacidades de detección, segmentación, razonamiento visual y generación de lenguaje natural, lo que lo convierte en un componente esencial para sistemas que requieren comprender su entorno de forma autónoma o asistir a un usuario humano en la interpretación de este.

En el marco del presente trabajo, el análisis de escenas constituye la base sobre la que se articula la interacción entre los sistemas físicos (como el dispositivo Búho o los robots zoomórficos) y el entorno visual que los rodea. Su correcta implementación permite estructurar la información captada por los sensores en términos comprensibles y útiles, tanto para un operador humano como para los propios algoritmos que gobiernan el comportamiento del sistema.

Por ello, este proyecto tiene como finalidad principal investigar, seleccionar y aplicar algoritmos avanzados de análisis de imágenes basados en inteligencia artificial para su integración en sistemas reales de telepresencia inmersiva y plataformas robóticas. A través de este enfoque, se busca no solo demostrar la viabilidad técnica de dicha integración, sino también evaluar su impacto en la mejora de la comprensión visual del entorno, la autonomía de los dispositivos, y la calidad de la interacción hombre-máquina. En definitiva, el propósito del trabajo es contribuir al desarrollo de soluciones inteligentes que permitan a estos sistemas interpretar su entorno de manera significativa, actuar con mayor criterio y adaptarse con eficacia a contextos complejos y cambiantes. Esta línea de investigación se apoya en tecnologías de vanguardia [22], y responde a una necesidad cada vez más evidente en múltiples sectores: que las máquinas no solo vean, sino que entiendan.

Es por ello que, además, se hace necesaria la integración de una capa de razonamiento sobre las escenas, basada en modelos avanzados de lenguaje (más conocidos por la sigla LLM, del inglés *Large Language Model*). Este componente actúa como un módulo cognitivo que, a partir de la información generada por el análisis de escenas, es capaz de procesarla y generar respuestas a partir de ella de un modo similar a como lo haría una persona.

De este modo, el análisis de escenas aporta a estos sistemas una capacidad análoga al sentido de la vista mientras que el LLM representa un equivalente funcional del cerebro: permiten razonar sobre lo

que se percibe, establecer conexiones entre los elementos de la escena y contextualizar la información en función de la situación y las necesidades del usuario.

Este planteamiento permite ir más allá de la mera transmisión visual o de la descripción básica del entorno, abriendo la puerta a formas de interacción y comprensión más ricas que resultan clave en aplicaciones de telepresencia inmersiva y robótica.

No obstante, estas aspiraciones se enfrentan a una serie de desafíos tecnológicos que todavía condicionan su aplicación práctica:

- La madurez de estas tecnologías sigue siendo limitada, y su despliegue en escenarios complejos requiere superar barreras significativas en rendimiento y estabilidad.
- La gran cantidad de recursos que demanda el procesamiento de grandes volúmenes de datos visuales puede comprometer la experiencia de inmersión y la operatividad de sistemas autónomos o asistidos en tiempo real [23].
- La mayoría de los modelos actuales ofrecen buen rendimiento en contextos cerrados, pero carecen aún de la robustez necesaria para operar con fiabilidad en situaciones imprevisibles o altamente cambiantes [24].
- La dificultad de interpretar los resultados de modelos de aprendizaje profundo, cajas negras difíciles de auditar, lo que plantea retos de trazabilidad y confianza [24].
- El coste económico de implementación a gran escala representa una barrera importante, especialmente en aplicaciones industriales o institucionales donde la escalabilidad es un factor decisivo.

En este contexto de grandes posibilidades, pero también de retos concretos, se enmarca el presente Trabajo de Fin de Máster, que busca contribuir con soluciones técnicas reales al avance de esta frontera tecnológica.

1.4. Objetivos del proyecto

Los objetivos definidos responden tanto a necesidades tecnológicas identificadas en el estado del arte como a los desafíos prácticos observados durante el desarrollo del trabajo. A continuación, se presentan los objetivos concretos del proyecto:

- **Estudiar en profundidad distintas técnicas y modelos de análisis de escenas mediante inteligencia artificial**, abarcando métodos aplicables a cuadros, secuencias de vídeo y arquitecturas multimodales. El estudio incluirá su adaptación a diferentes formatos de entrada, así como el análisis de su coherencia temporal.
- **Integrar un sistema de análisis de escenas en un robot zoomórfico,** con el objetivo de recibir y procesar el vídeo capturado por el propio dispositivo, permitiendo identificar elementos relevantes del entorno y facilitar futuras tareas de interacción o navegación básica.
- **Evaluar y comparar el rendimiento de distintos modelos de análisis de escenas mediante el uso de métricas objetivas y herramientas de evaluación con lenguaje natural**, utilizando bases de datos etiquetados y aplicando técnicas de LLM como juez, más conocido por el inglés *LLM-as-a-Judge*.

- **Diseñar e implementar una arquitectura distribuida cliente-servidor para el procesamiento remoto de imágenes**, que permita enviar información visual desde el robot u otros dispositivos hacia un servidor de procesamiento, y retornar los resultados analizados para su visualización en distintos terminales.
- **Establecer un canal de comunicación funcional entre el dispositivo de *hardware* «El Búho»** y los servidores de análisis, ampliando las posibilidades de captura y envío de datos desde plataformas físicas heterogéneas.
- **Integrar los resultados obtenidos de los distintos modelos en un sistema único y coherente**, que combine las capacidades de análisis de escenas con un módulo de razonamiento basado en LLM, que permita generar respuestas en lenguaje natural y facilite su aplicación conjunta en escenarios reales.

1.5. Metodologías y fases de desarrollo

El desarrollo del proyecto se ha estructurado a lo largo de distintas fases, marcadas por una evolución tanto en los objetivos como en la complejidad de los elementos que lo componen. Desde el inicio, se adoptó una metodología flexible, basada en una combinación de planificación anticipada y capacidad de adaptación a medida que aparecían nuevas necesidades o limitaciones. En lugar de seguir una ruta completamente fija desde el principio, se optó por construir una primera versión esencial, sobre la que fue posible iterar, ajustar y construir progresivamente nuevas funcionalidades.

Esta forma de trabajar permitió que, a medida que el proyecto avanzaba, se fueran definiendo con mayor precisión los roles de cada componente, así como las relaciones entre ellos, buscando siempre que el conjunto fuera fácil de mantener, escalar o modificar. En este sentido, cada decisión se tomó con la mirada puesta no solo en su efectividad inmediata, sino también en su impacto a medio plazo en la organización general del sistema.

Las fases de trabajo no fueron completamente lineales, sino que en muchos casos fue necesario volver sobre pasos anteriores para reconfigurar ciertas partes, incorporar aprendizajes obtenidos o adaptar el diseño a nuevas condiciones. Esta lógica de revisión continua no fue percibida como un problema, sino como una consecuencia natural del tipo de solución que se buscaba: una arquitectura versátil, pensada para evolucionar con el tiempo. Entre las fases del trabajo definidas, destacan:

- Fase 1: Estudio del estado del arte.
- Fase 2: Familiarización con las plataformas y herramientas.
- Fase 3: Diseño inicial del sistema.
- Fase 4: Desarrollo de prototipos funcionales y modelo cliente servidor.
- Fase 5: Adquisición de datos, tratamiento de los mismo y evaluación de los modelos.
- Fase 6: Identificación y desarrollo de casos de uso concretos.
- Fase 7: Iteración y refinamiento.
- Fase 8: Documentación y redacción de la memoria.

En la Tabla 1.1 se muestra una distribución temporal aproximada del tiempo dedicado a cada una de las anteriores fases durante el curso académico en el que se ha realizado este TFM.

Tabla 1.1. Planificación temporal del TFM

Mes	Fases de desarrollo							
	1	2	3	4	5	6	7	8
Octubre			—	—	—	—	—	—
Noviembre				—	—	—	—	—
Diciembre	—	—		—	—	—	—	—
Enero	—	—			—	—	—	—
Febrero	—	—	—			—	—	—
Marzo	—	—	—	—		—	—	—
Abril	—	—	—		—			—
Mayo	—	—	—		—			
Junio	—	—	—	—	—			

El desarrollo se apoyó además en una constante reflexión sobre cómo mejorar y reutilizar cada uno de los programas y módulos realizados, buscando siempre una utilidad global del mismo. Este principio guio muchas de las decisiones clave, y permitió mantener una estructura lo bastante abierta como para integrar casos de uso diversos.

1.6. Competencias empleadas

El Máster en Ingeniería Industrial destaca por la transversalidad y multidisplinaridad de los contenidos que en él se imparten, otorgando la formación necesaria para, en el futuro, resolver satisfactoriamente los problemas industriales desde diversos ámbitos del conocimiento. El impacto formativo de este TFM se ve reflejado en las siguientes competencias, conocimientos y habilidades [25].

1.6.1. Competencias

- **Saber aplicar e integrar sus conocimientos, la comprensión de estos, su fundamentación científica y sus capacidades de resolución de problemas en entornos nuevos y definidos de forma imprecisa, incluyendo contextos de carácter multidisciplinar tanto investigadores como profesionales altamente especializados:** Se ha requerido tener unos conocimientos previos y comprenderlos adecuadamente para poder realizar correctamente este TFM. Además, al haber realizado el mismo de manera simultánea a las prácticas en un laboratorio con perfiles tanto técnicos como investigadores, la gestión de entornos nuevos e interdisciplinares ha sido total.
- **Ser capaces de predecir y controlar la evolución de situaciones complejas mediante el desarrollo de nuevas e innovadoras metodologías de trabajo adaptadas al ámbito científico/investigador, tecnológico o profesional concreto, en general multidisciplinar, en el que se desarrolle su actividad:** Una constante durante el desarrollo de un trabajo como este.

- **Proyectar, calcular y diseñar productos, procesos, instalaciones y plantas:** El proyecto se basa, casi íntegramente, en el diseño de nuevos procesos.
- **Realizar investigación, desarrollo e innovación en productos, procesos y métodos:** La explicación es prácticamente análoga al punto anterior, al ser necesaria esta investigación previa para poder realizar un proyecto que se precie.
- **Conocimiento y capacidades para el proyectar y diseñar instalaciones eléctricas y de fluidos, iluminación, climatización y ventilación, ahorro y eficiencia energética, acústica, comunicaciones, domótica y edificios inteligentes e instalaciones de seguridad:** Ya que este proyecto se ha realizado durante unas prácticas en Nokia Spain, el enfoque del mismo hacia las comunicaciones es claro. Además, dado su carácter innovativo, los temas que se tratan tienen sin duda relación con instalaciones domóticas y de seguridad.
- **Realización, presentación y defensa, una vez obtenidos todos los créditos del plan de estudios, de un ejercicio original realizado individualmente ante un tribunal universitario, consistente en un proyecto integral de Ingeniería Industrial de naturaleza profesional en el que se sinteticen las competencias adquiridas en las enseñanzas:** Este TFM constituye un ejercicio completo de integración de conocimientos adquiridos durante el máster, con aplicación directa en un entorno profesional real. Supone la culminación del proceso formativo y refleja de forma estructurada y aplicada las competencias propias del perfil del ingeniero industrial.
- **Capacidad para diseñar y proyectar sistemas de producción automatizados y control avanzado de procesos:** El proyecto aborda directamente el diseño de sistemas inteligentes con capacidad para interpretar escenas en tiempo real y automatizar decisiones. Estas soluciones se integran en sistemas robóticos y de telepresencia, con aplicación directa en entornos de producción y control avanzado.
- **Capacidad para diseñar sistemas electrónicos y de instrumentación industrial:** Aunque no se ha abordado el diseño electrónico desde un punto de vista *hardw*are, el trabajo implica la selección, configuración e integración de sensores y sistemas de percepción visual avanzados, formando parte esencial de la instrumentación empleada en las plataformas robóticas y de telepresencia.

1.6.2. Conocimientos, habilidades y destrezas

Los resultados de aprendizaje también incluyen una serie de conocimientos, habilidades y destrezas, que constituyen los saberes teóricos adquiridos que permiten realizar una tarea concreta con eficacia. De entre los trabajados, destacan:

- **Haber adquirido conocimientos avanzados y demostrado, en un contexto de investigación científica y tecnológica o altamente especializado, una comprensión detallada y fundamentada de los aspectos teóricos y prácticos y de la metodología de trabajo en uno o más campos de estudio:** El desarrollo del proyecto ha exigido una comprensión profunda tanto de los fundamentos teóricos de la visión por ordenador como de su aplicación práctica, en un entorno tecnológico especializado.
- **Tener conocimientos adecuados de los aspectos científicos y tecnológicos de métodos matemáticos, analíticos y numéricos en la ingeniería, automática, electrónica industrial, informática industrial, fabricación, etc.:** Se han aplicado

conceptos y herramientas propias de estas áreas para el análisis de algoritmos, el manejo de *software* técnico y la integración con sistemas robóticos.

- **Conocimientos y capacidades para organizar y dirigir empresas:** Aunque no es el foco principal del proyecto, se ha trabajado con criterios organizativos y de gestión durante las fases de planificación y evaluación del desarrollo.
- **Conocimientos y capacidades de estrategia y planificación aplicadas a distintas estructuras organizativas:** La organización de las fases del proyecto y su ejecución temporal se han estructurado con criterios propios de planificación estratégica y gestión de proyectos.
- **Conocimientos de sistemas de información a la dirección, organización industrial, sistemas productivos y logística y sistemas de gestión de calidad:** El enfoque estructurado del proyecto y el uso de herramientas digitales avanzadas se alinea parcialmente con esta competencia, aunque sin un tratamiento específico de sistemas de gestión.
- **Saber transmitir de un modo claro y sin ambigüedades a un público especializado o no, resultados procedentes de la investigación científica y tecnológica o del ámbito de la innovación más avanzada:** Esta competencia se trabaja tanto en la elaboración del documento como en su futura presentación y defensa ante tribunal, adaptando el lenguaje técnico a distintos perfiles.
- **Haber desarrollado la autonomía suficiente para participar en proyectos de investigación y colaboraciones científicas o tecnológicas dentro su ámbito temático:** El proyecto se ha llevado a cabo de forma autónoma y en paralelo a prácticas profesionales, integrándose en un entorno de investigación aplicada.
- **Ser capaces de asumir la responsabilidad de su propio desarrollo profesional y de su especialización en uno o más campos de estudio:** El trabajo ha reforzado el desarrollo independiente y la profundización técnica en áreas avanzadas como el análisis semántico y la robótica.
- **Poder ejercer funciones de dirección general, dirección técnica y dirección de proyectos I+D+i en plantas, empresas y centros tecnológicos:** Se ha trabajado con criterios propios de proyectos de innovación tecnológica, aplicando metodología estructurada y herramientas propias del ámbito industrial.
- **Aplicar los conocimientos adquiridos y resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares:** El trabajo ha exigido resolver problemas técnicos en un entorno novedoso con alta componente interdisciplinar.
- **Ser capaz de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios:** El proyecto invita a reflexionar sobre el impacto del uso de sistemas inteligentes en distintos ámbitos, especialmente en cuanto a privacidad y uso ético de los datos visuales.
- **Saber comunicar las conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin**

ambigüedades: Esta competencia se refleja en la redacción del documento y en la necesidad de presentar los resultados de forma accesible para perfiles técnicos y no técnicos.

- **Poseer las habilidades de aprendizaje que permitan continuar estudiando de un modo autodirigido o autónomo:** El proyecto ha requerido buscar soluciones de forma independiente, aprender nuevas herramientas y adaptarse a tecnologías no previamente conocidas.

1.7. Estructura del documento

Este documento se encuentra estructurado en cinco capítulos principales, además de una serie de anexos complementarios, con el objetivo de proporcionar una visión clara, progresiva y ordenada del trabajo realizado. La organización del contenido responde a una lógica técnico-formal que permite abordar, de forma sistemática, tanto el contexto teórico como el desarrollo práctico del proyecto.

En el Capítulo 1, se introducen los elementos fundamentales del Trabajo Fin de Máster. Se presentan el interés del tema abordado, su motivación y los objetivos específicos que se persiguen. Además, se describen la metodología seguida, las competencias desarrolladas a lo largo del proyecto y la estructura general del documento, con el fin de facilitar al lector una guía del contenido que encontrará en las siguientes secciones.

El Capítulo 2 desarrolla el estudio del estado del arte y los fundamentos técnicos sobre los que se apoya el trabajo. En él se realiza un análisis detallado de los principales conceptos relacionados con el análisis semántico de escenas, los fundamentos de la inteligencia artificial aplicada a visión por ordenador, los modelos más relevantes existentes en la literatura, así como los criterios y herramientas de evaluación empleados. Asimismo, se presentan los aspectos técnicos relacionados con los sistemas distribuidos, el soporte *hardw*are necesario y las aplicaciones actuales en el ámbito de la robótica y la telepresencia inmersiva. Este capítulo sirve como base teórica que sustenta las decisiones tecnológicas adoptadas en las fases posteriores del proyecto.

En el Capítulo 3 se describe la metodología concreta seguida y los recursos empleados. Se detallan el enfoque metodológico, las herramientas *software* utilizadas, y los sistemas *hardware* con los que se ha trabajado, incluyendo los distintos dispositivos de captura, procesamiento y visualización empleados durante el desarrollo del proyecto.

El Capítulo 4 constituye el núcleo principal del trabajo, donde se presenta el desarrollo técnico del proyecto, los resultados obtenidos y su validación. En él se abordan aspectos como la integración del sistema robótico, el desarrollo e implementación de los distintos algoritmos de análisis de escenas, la arquitectura cliente-servidor empleada para el procesamiento distribuido y la evaluación comparativa de los modelos. Asimismo, se incluye el trabajo realizado en la integración de dispositivos adicionales, como el *hardware* específico Búho, y se propone una integración de resultados en un sistema coherente y modular. El capítulo finaliza con una síntesis técnica de los resultados, así como un análisis crítico de las dificultades encontradas, las decisiones adoptadas y las posibles mejoras identificadas.

Finalmente, el Capítulo 5 presenta las conclusiones generales del proyecto, las propuestas de mejora y líneas de trabajo futuras, y una valoración personal sobre el trabajo desarrollado y el aprendizaje alcanzado durante su ejecución.

Además de los capítulos principales, el documento incorpora un apéndice dedicado a la alineación del proyecto con los Objetivos de Desarrollo Sostenible (ODS).

Capítulo 2

ESTUDIO DEL ESTADO DEL ARTE Y FUNDAMENTOS TÉCNICOS

Este capítulo expone los fundamentos conceptuales y técnicos que sustentan el desarrollo del sistema propuesto. A través de una revisión estructurada de las principales técnicas de análisis de escenas mediante inteligencia artificial se identifican los avances más relevantes en el ámbito de la visión por ordenador y se justifica la selección de los modelos y arquitecturas empleados en este trabajo.

El enfoque adoptado combina un estudio del estado del arte con la exposición detallada de los principios técnicos implicados, abarcando desde los métodos clásicos hasta los modelos multimodales de última generación. Se analizan tanto los aspectos teóricos, por ejemplo, el funcionamiento de las redes neuronales aplicadas al reconocimiento visual, como las herramientas prácticas necesarias para su implementación en un entorno distribuido, con especial atención a las limitaciones actuales y a los retos derivados de su aplicación en contextos reales.

Asimismo, se revisan conceptos clave relacionados con el procesamiento de imágenes en arquitecturas cliente-servidor, la percepción visual en robótica y las posibilidades que ofrece la telepresencia inmersiva como caso de uso avanzado. Todo ello con el objetivo de establecer un marco sólido sobre el que se construyen las decisiones de diseño adoptadas en el desarrollo del sistema descrito en los capítulos posteriores.

2.1. Fundamentos de IA para visión por ordenador y razonamiento

2.1.1. Principios del aprendizaje profundo

El aprendizaje profundo (*deep learning*) constituye una subárea del aprendizaje automático (*machine learning*) que se basa en el uso de modelos jerárquicos compuestos por múltiples capas de procesamiento no lineales. Estos modelos, inspirados en la arquitectura del cerebro humano, permiten representar datos con altos niveles de abstracción, facilitando la extracción automática de características relevantes a partir de información en bruto [23]. A diferencia de los enfoques tradicionales, donde el diseño de las variables requería una intervención manual intensiva, el aprendizaje profundo permite que las propias redes aprendan estas representaciones directamente desde los datos.

Una de las características esenciales del aprendizaje profundo es su capacidad para aproximar funciones complejas mediante la composición de transformaciones sucesivas. Cada capa de una red profunda actúa como una función que opera con una gran cantidad de datos: las primeras detectan patrones simples mientras que las capas posteriores combinan esos patrones para identificar estructuras más complejas [26], tal y como se muestra en la Figura 2.1. Esta propiedad de abstracción jerárquica es la base de su éxito en tareas como el reconocimiento de imágenes, la comprensión del lenguaje natural o el análisis de señales temporales.

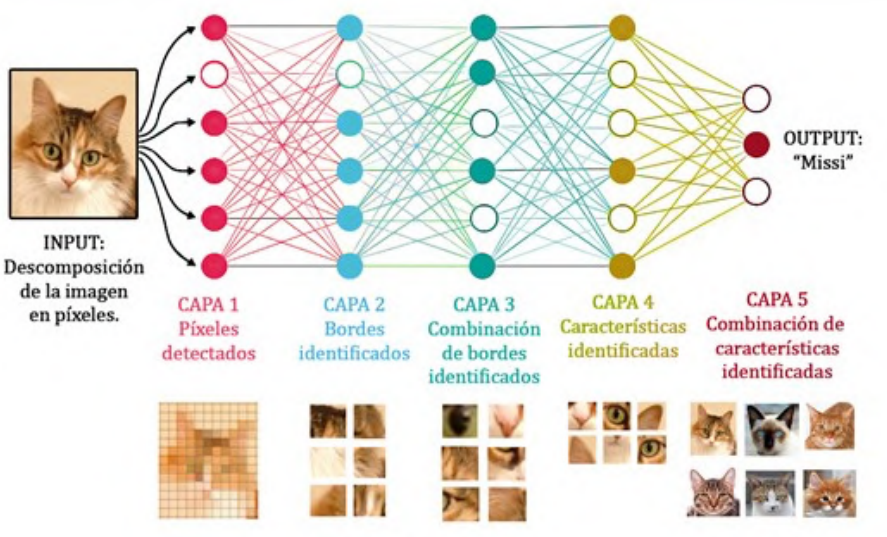


Figura 2.1. Esquema mostrando el funcionamiento del *deep learning* [27]

Desde el punto de vista computacional, el aprendizaje profundo ha avanzado gracias al incremento en la disponibilidad de datos a gran escala, la mejora de las capacidades de cómputo, especialmente mediante el uso de GPU (por las siglas en inglés de Unidad de Procesamiento Gráfico, *Graphics Processing Unit*) y TPU (Unidad de Procesamiento Tensorial, *Tensor Processing Unit*), y el desarrollo de técnicas de entrenamiento más eficientes, como la normalización por lotes (*batch normalization*), funciones de activación avanzadas y estrategias de regularización (*dropout, early stopping*) [28]. Estos avances han permitido escalar las redes a miles de millones de parámetros, aumentando su capacidad predictiva, aunque también introduciendo nuevos desafíos relacionados con el coste computacional, la interpretabilidad y el riesgo de sobreajuste.

En el ámbito de la visión por ordenador, una de las áreas más consolidadas del aprendizaje profundo, estas redes han revolucionado el análisis de cuadros y vídeo, al superar con creces los métodos clásicos basados en reglas heurísticas o descriptores manuales [29]. Gracias a arquitecturas adecuadas y a grandes conjuntos de datos etiquetados, se ha logrado un desempeño eficiente en tareas como clasificación de imágenes, segmentación semántica y detección de objetos. Este progreso ha sido igualmente determinante en el desarrollo de modelos más recientes que combinan entrada visual y textual, un enfoque multimodal que, como se introdujo en el Capítulo 1, resulta esencial en los sistemas de análisis de escenas integrados con módulos de razonamiento basados en LLM empleados en este trabajo.

No obstante, el aprendizaje profundo no está exento de limitaciones. Su rendimiento depende en gran medida de la cantidad y calidad de los datos de entrenamiento, así como del ajuste fino de un gran número de hiperparámetros. Además, su naturaleza opaca ha suscitado un creciente interés por técnicas

de interpretabilidad y explicabilidad, especialmente en contextos donde la fiabilidad y la trazabilidad son requisitos críticos [30].

En resumen, el aprendizaje profundo representa una referencia metodológica, y un punto de partida técnico para el desarrollo y selección de los modelos analizados en los siguientes apartados. Su capacidad para representar de forma jerárquica la información visual es clave para abordar el análisis estructurado de escenas, que se detalla en los modelos específicos a partir del apartado 2.2.

2.1.2. Redes neuronales convolucionales

El tipo más básico de red neuronal recibe el nombre de «perceptrón multicapa» (o MLP, del inglés *multilayer perceptron*) aunque últimamente se las conoce también como redes completamente conectadas, y su importancia radica en que puede utilizarse como clasificador. Su funcionamiento no es tan complejo como cabría esperar: cada neurona (también llamadas «nodos» en algunas fuentes) representa una variable de estado, y además lleva asociado un valor llamado umbral o sesgo; a cada conexión entre neuronas se le asigna otro valor, en este caso llamado peso; así, para cada nodo queda definida una función que depende de los valores anteriores. De esta manera, los valores de entrada que recibe una neurona se multiplican por los pesos propios de cada conexión, se suman entre sí y se suman al umbral o sesgo: esa es la respuesta de la neurona. La ecuación 2.1 expresa matemáticamente la salida de una neurona, de la que además se muestra una representación gráfica en la Figura 2.2.

$$s = \sum (e_i \cdot p_i) + u \tag{2.1}$$

Las redes neuronales convolucionales (CNN, por sus siglas en inglés, *Convolutional Neural Network*) constituyen uno de los pilares fundamentales del aprendizaje profundo aplicado a visión por ordenador. A diferencia de las redes ordinarias, las CNN están diseñadas para explotar la estructura espacial de los datos visuales mediante el uso de filtros o núcleos de convolución que se deslizan sobre la imagen de entrada, extrayendo características locales como bordes, texturas o patrones geométricos [31].

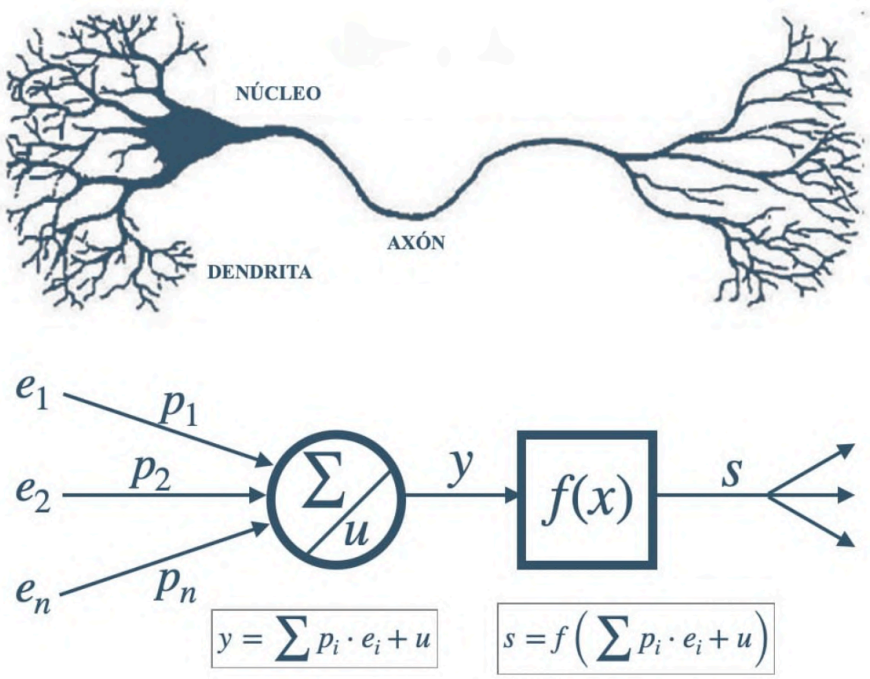


Figura 2.2. Neurona real y artificial

El funcionamiento básico de una CNN se basa en la aplicación sucesiva de tres tipos de capas principales: capas convolucionales, capas de activación no lineal y capas de reducción de dimensionalidad, como se mostró en la Figura 2.1, y se presenta aquí de manera más teórica en la Figura 2.3. Las primeras detectan patrones locales, las segundas introducen no linealidad para aumentar la capacidad de modelado, y las terceras ayudan a reducir la complejidad computacional y mejorar la robustez frente a traslaciones o pequeñas deformaciones en la imagen [32].

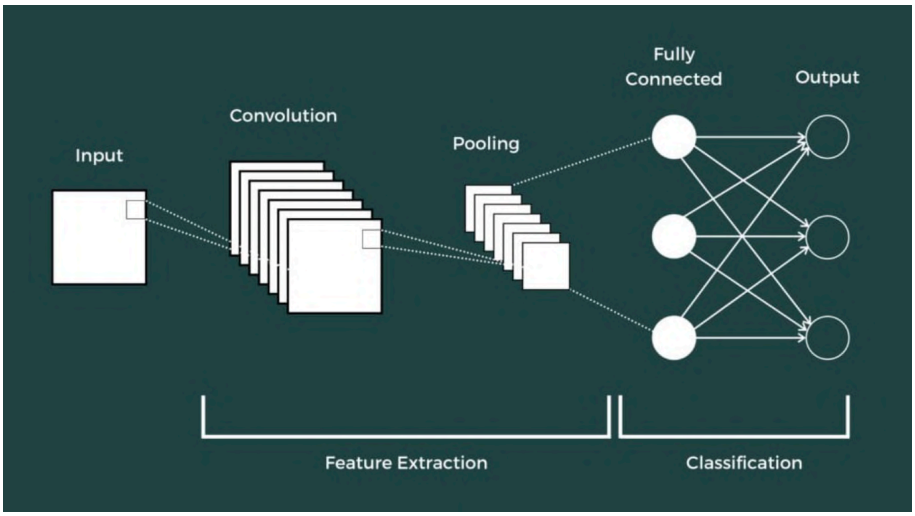


Figura 2.3. Red neuronal convolucional [33]

Uno de los aspectos clave de las CNN es que los pesos permanecen fijos a lo largo del procesado, lo que permite que un mismo filtro se aplique a toda la imagen, reduciendo el número de parámetros respecto a una red densa tradicional. Esto no solo mejora la eficiencia del entrenamiento, sino que también facilita la generalización del modelo, al centrarse en patrones repetidos que aparecen en distintas regiones de la imagen. Además, las CNN aprovechan la estructura tridimensional de los datos visuales (alto, ancho, número de canales) para mantener la coherencia espacial entre las capas, permitiendo así una interpretación más natural de las activaciones intermedias [34].

Desde su aparición, las CNN han experimentado un desarrollo vertiginoso. Modelos como LeNet-5 [39], orientado al reconocimiento de dígitos manuscritos, sentaron las bases de esta arquitectura. Sin embargo, el punto de inflexión se produjo con AlexNet en 2012, que demostró el potencial del aprendizaje profundo en el desafío ImageNet, superando ampliamente a los métodos clásicos en tareas de clasificación [29]. A partir de ahí, surgieron arquitecturas más sofisticadas como VGGNet, GoogLeNet o ResNet, esta última introduciendo el concepto de conexiones residuales para facilitar el entrenamiento de redes muy profundas [28].

En el contexto de este trabajo, las CNN han sido empleadas como base para modelos de análisis de escenas en cuadros y vídeos, especialmente en tareas de detección de objetos, segmentación semántica o extracción de descripciones visuales. Su capacidad para identificar regiones relevantes y representar de forma jerárquica la información visual las convierte en herramientas particularmente eficaces en entornos robóticos, donde la eficiencia y la robustez son factores críticos.

No obstante, las CNN presentan también ciertas limitaciones. Su capacidad para capturar relaciones espaciales a larga distancia es limitada, lo que ha motivado la aparición de arquitecturas más recientes basadas en mecanismos de atención, como los *transformers* visuales, que se tratarán en el siguiente apartado 2.1.3. Además, su entrenamiento requiere grandes volúmenes de datos etiquetados y recursos

computacionales significativos, lo que ha impulsado el desarrollo de técnicas de transferencia de conocimiento y aprendizaje auto-supervisado. Sin embargo, pese a estas limitaciones, las CNN siguen siendo una tecnología de referencia en tareas de visión por ordenador aplicadas al análisis de escenas, y muchos modelos actuales híbridos [32], integran convoluciones como componentes iniciales o auxiliares, aprovechando su eficiencia espacial y su capacidad de extraer características robustas.

En conclusión, las redes convolucionales han supuesto un avance fundamental para el tratamiento de imágenes en inteligencia artificial, y siguen siendo una tecnología central en numerosas aplicaciones industriales, médicas y robóticas. Su integración en sistemas distribuidos, como los desarrollados en este proyecto, permite dotar de capacidades perceptivas avanzadas a dispositivos con acceso remoto y conectividad en tiempo real.

2.1.3. *Transformers* en visión por ordenador

El desarrollo de arquitecturas basadas en *transformers* ha implicado una gran evolución en el campo de la inteligencia artificial, especialmente en el procesamiento de lenguaje natural (NLP, *Natural Language Processing*), donde estos modelos han superado con creces a los empleados con anterioridad. El mecanismo clave que sustenta su eficacia es la «autoatención», una operación que permite modelar relaciones arbitrarias entre elementos de una secuencia de entrada, sin asumir ninguna estructura local previa [35]. Esta propiedad ha resultado ser especialmente valiosa al ser trasladada al dominio visual, dando lugar a una nueva generación de modelos capaces de procesar imágenes como secuencias de elementos, con una capacidad de representación global y contextualizada que trasciende las limitaciones espaciales de las CNN.

La aplicación de *transformers* en visión por ordenador comenzó con el *Vision Transformer* (ViT), introducido por [36]. En ViT, una imagen se divide en parches de tamaño fijo (por ejemplo, 16x16 píxeles), que se aplanan y proyectan al espacio como vectores que son tratados como *tokens* y procesados mediante un *transformer* estándar, idéntico al utilizado en NLP. La clave de este modelo es que no impone ninguna restricción de vecindad, permitiendo que cada parche atienda a todos los demás, lo que habilita una captación temprana de relaciones espaciales a larga distancia.

Aunque ViT logró resultados competitivos en clasificación de imágenes, su rendimiento dependía fuertemente del tamaño de la base de datos y del poder computacional disponible. Posteriormente, se introdujeron variantes más eficientes y entrenables, como DeiT (*Data-efficient Image Transformers*) [37], que incorporó técnicas para optimizar entrenamiento supervisado, permitiendo obtener buenos resultados incluso con conjuntos de datos más pequeños.

A nivel funcional, los *transformers* visuales ofrecen ventajas claras frente a las CNN en tareas que implican razonamiento estructural, inferencia contextual o interpretación de relaciones espaciales complejas. En lugar de extraer representaciones puramente locales, los modelos basados en atención pueden integrar información de toda la escena desde las primeras capas del procesamiento, lo que resulta especialmente ventajoso en análisis de escenas, identificación de interacciones entre objetos o formulación de interpretaciones de alto nivel.

Esta capacidad de modelar relaciones globales sin recurrir a estructuras convolucionales también ha favorecido el desarrollo de modelos híbridos, que combinan convoluciones en las primeras etapas (para una extracción eficiente de características de bajo nivel) con bloques de atención en niveles superiores, intentando explotar lo mejor de ambos mundos.

Fruto de su capacidad para integrar múltiples fuentes de información, los *transformers* son especialmente adecuados para modelos multimodales, que procesan simultáneamente información visual y textual. Esta propiedad ha sido aprovechada en modelos como CLIP (*Contrastive Language-Image Pretraining*) [38], Florence, BLIP (*Bootstrapping Language-Image Pre-training*), Flamingo y LLaVA (*Large Language and Vision Assistant*), que alinean representaciones visuales con lenguaje natural mediante técnicas de entrenamiento contrastivo o generativo. Estos modelos no solo comprenden el contenido visual, sino que también pueden razonar sobre él, responder preguntas, generar descripciones o clasificar imágenes en función de criterios textuales, lo que los convierte en herramientas idóneas para escenarios donde es necesario interpretar y comunicar el contenido visual de forma precisa.

Desde el punto de vista técnico, el principal desafío de los *transformers* radica en su coste computacional: el mecanismo de atención escala cuadráticamente con el número de *tokens*, lo que implica una carga computacional elevada en imágenes de alta resolución. Para paliar este problema, se han desarrollado variantes que aproximan la atención completa mediante técnicas de reducción dimensional o atención localizada.

En este trabajo, los modelos basados en *transformers* se emplean como núcleo para tareas avanzadas de análisis de escenas, incluyendo interpretación visual en tiempo real, procesamiento multimodal y evaluación asistida por lenguaje. Estas capacidades resultan determinantes para alcanzar los objetivos del sistema propuesto, que requiere comprender y contextualizar el entorno visual con un alto nivel de precisión, adaptándose dinámicamente a escenarios cambiantes y permitiendo una interacción inteligente en entornos distribuidos, robóticos o de telepresencia inmersiva.

2.1.4. Modalidades de aprendizaje

El rendimiento y la aplicabilidad de los modelos de visión por ordenador no dependen únicamente de la arquitectura utilizada, sino también, de forma crítica, del enfoque de aprendizaje adoptado. La estrategia elegida para entrenar el modelo condiciona directamente la calidad de las representaciones generadas, la cantidad de datos necesarios, su escalabilidad y su capacidad para generalizar a contextos nuevos. En este apartado se describen las principales modalidades de aprendizaje utilizadas en visión artificial, con especial atención a su impacto en el análisis de escenas:

- **Aprendizaje supervisado**

  Es la modalidad más tradicional y extendida, en la que los modelos aprenden a partir de ejemplos etiquetados manualmente. En el ámbito visual, esto se traduce en conjuntos de datos donde cada imagen, región o píxel está asociado a una etiqueta semántica, clase, máscara de segmentación o descripción textual. La red ajusta sus parámetros minimizando una función de error entre las predicciones generadas y los valores esperados.

  Este enfoque ha sido determinante en el auge de las redes convolucionales profundas, favorecido por grandes bases de datos como ImageNet [39] o COCO [40]. Sin embargo, su efectividad conlleva una limitación crítica: la necesidad de disponer de grandes volúmenes de anotaciones precisas, lo que puede resultar inviable en dominios donde el etiquetado es costoso o escaso, como sucede en sectores industriales, clínicos o científicos.

- **Aprendizaje auto-supervisado**

  Para paliar esta dependencia del etiquetado, se ha consolidado el aprendizaje auto-supervisado, donde los modelos se entrenan a partir de tareas internas o pretextos que no requieren intervención humana. Estas tareas, como reconstruir regiones ocultas, predecir relaciones

espaciales o contrastar diferentes vistas de una misma escena [41], permiten que el modelo descubra regularidades y estructuras latentes por sí mismo.

Modelos como SimCLR o MAE (*Masked Autoencoders*) han puesto de manifiesto que estas estrategias pueden generar representaciones visuales de gran utilidad, reutilizables en fases posteriores de ajuste fino con tareas específicas. Esta aproximación resulta especialmente adecuada en contextos con grandes volúmenes de datos no estructurados y escasa anotación, como ocurre habitualmente en el análisis de escenas reales.

Además, este enfoque facilita la creación de modelos transferibles y polivalentes, lo que ha sido clave en el surgimiento de los llamados *foundation models* [42], entrenados para capturar representaciones generales antes de ser adaptados a usos concretos. Su aplicación al análisis visual permite abordar problemas diversos desde una base común y robusta.

- **Aprendizaje multimodal**

  En situaciones reales, el análisis de escenas rara vez se limita a una única fuente de información. Los sistemas deben ser capaces de integrar diferentes tipos de datos, como lenguaje, señales acústicas o estímulos físicos. Por ello, el aprendizaje multimodal se ha consolidado como una estrategia esencial, en la que los modelos se entrenan para alinear o fusionar representaciones procedentes de dominios distintos.

  Modelos como CLIP, ViLT (*Vision-and-Language Transformer*), Florence o LLaVA han sido entrenados para combinar visión y lenguaje, permitiendo realizar tareas como clasificación condicionada al texto, generación de descripciones automáticas o respuesta a preguntas visuales [43]. Esta capacidad es especialmente relevante para el presente trabajo, ya que se plantea el uso de modelos multimodales tanto para el análisis visual como para la evaluación verbal del contenido interpretado (tal y como se introdujo en el Capítulo 1 al hablar sobre la integración de análisis de escenas y razonamiento basado en LLM), una funcionalidad que requiere una compresión profunda e integrada del entorno observado.

  En este ámbito, las arquitecturas tipo *transformer* han demostrado ser especialmente eficaces, ya que tratan las secuencias visuales y lingüísticas de forma homogénea. Esto facilita el entrenamiento multitarea y la transferencia entre dominios, permitiendo construir modelos flexibles y reutilizables.

- **Aprendizaje por refuerzo visual**

  Aunque menos habitual en tareas de percepción, el aprendizaje por refuerzo adquiere un papel destacado cuando el sistema debe tomar decisiones en un entorno y aprender de sus consecuencias. En robótica y navegación autónoma, por ejemplo, el análisis visual se entrelaza con políticas de acción que se optimizan en función de las recompensas obtenidas por el agente.

  El reto principal en estos casos es mantener la coherencia entre los módulos de percepción y los de decisión, garantizando que las representaciones visuales sean útiles para actuar. Entornos simulados como Habitat [44] han demostrado la viabilidad de entrenar agentes capaces de interpretar escenas y comportarse de forma adaptativa en entornos complejos.

Cada una de estas modalidades, supervisada, auto-supervisada, multimodal y por refuerzo, ofrece ventajas complementarias y responde a necesidades diferentes dentro del espectro del aprendizaje en visión artificial. En el contexto de este trabajo, donde se busca construir un sistema capaz de analizar escenas de forma eficiente, flexible y contextual, la combinación de un preentrenamiento auto-

supervisado con una posterior fase supervisada se perfila como la estrategia más adecuada. Este enfoque mixto permite aprovechar grandes volúmenes de datos no etiquetados, reducir la dependencia de anotaciones costosas y adaptar el modelo a tareas específicas de forma eficaz. En definitiva, constituye uno de los pilares técnicos fundamentales sobre los que se construye la propuesta desarrollada en este proyecto.

2.1.5. *Large Language Models* (LLM)

Los *Large Language Models* constituyen uno de los mayores avances en el campo de la inteligencia artificial en la última década. Entrenados sobre enormes cantidades de texto y datos no estructurados, estos modelos han demostrado capacidades extraordinarias en generación de lenguaje natural, razonamiento, inferencia semántica y resolución generalista de tareas. En los últimos años, han ampliado su alcance hacia la interacción multimodal con datos visuales, auditivos y sensoriales, convirtiéndose en componentes versátiles dentro de sistemas complejos de inteligencia artificial.

Desde el punto de vista técnico, un LLM se define como un modelo basado en la arquitectura *transformer*, con un número de parámetros que puede oscilar entre cientos de millones y cientos de miles de millones, entrenándose mediante tareas de predicción de *tokens*. Entre los más influyentes se encuentran GPT (*Generative Pre-trained Transformer*, desarrollado por OpenAI) [45], PaLM (*Pathways Language Model*, de Google) [46], LLaMA (*Large Language Model Architecture*, desarrollado por Meta) [47], Claude (de Anthropic) y DeepSeek, cada uno con diferencias significativas en tamaño, estructura y comportamiento.

Aunque inicialmente concebidos como modelos puramente lingüísticos, su capacidad para realizar razonamientos complejos y manejar instrucciones abiertas ha propiciado su integración en tareas visuales. En la actualidad, su papel resulta clave tanto en la evaluación de salidas generadas por modelos de visión como en la generación, interpretación y razonamiento sobre contenido visual.

Como ya se ha mencionado en algunas ocasiones, una de las aplicaciones más relevantes de los LLM en este trabajo es su uso como capa de razonamiento sobre las salidas generadas por sistemas de análisis de escenas, que ya producen descripciones lingüísticamente estructuradas o representaciones semánticas del entorno. Este uso, que todavía se está consolidando, permite a los LLM llevar a cabo tareas como el resumen, la clasificación, la extracción de información relevante o la generación de órdenes de control a partir de dicha información [48], dotando a los sistemas de una capacidad análoga a la del razonamiento humano sobre la percepción visual [49].

Arquitectura y funcionamiento

La base funcional de los LLM es el *transformer* autoregresivo, introducido en [35], en el que cada *token* es predicho condicionado a todos los anteriores en una secuencia. Esta estructura permite modelar dependencias de largo alcance y construir representaciones semánticas profundas. El entrenamiento inicial se realiza con bases de datos masivas (por ejemplo, Wikipedia, GitHub), seguido por procesos de ajuste fino.

Uno de los elementos críticos en el desarrollo de LLM es su escalabilidad. Investigaciones como las *Scaling Laws* (Leyes de escalabilidad) [50] han demostrado que el rendimiento mejora de forma logarítmica pero constante con el aumento del número de parámetros, lo que ha impulsado una carrera hacia modelos cada vez más grandes y complejos, con capacidades emergentes en tareas no previstas originalmente.

Aplicaciones en visión por ordenador

En el ámbito de la visión por ordenador, los LLM han comenzado a desempeñar un papel destacado en tres áreas principales:

- **Modelos multimodales con entrada visual**

  Modelos como CLIP [50], Flamingo, BLIP-2, Florence [51] y LLaVA integran visión e idioma en arquitecturas conjuntas que permiten tareas como describir cuadros, VQA (del inglés *Visual Question Answering*, es decir, respuesta a preguntas visuales), razonamiento visual y conversación. En estos sistemas, la entrada visual se codifica mediante un modelo especializado (por ejemplo, ViT), cuya salida se proyecta al espacio de *tokens* e interactúa con el modelo lingüístico.

  Esta estrategia, también empleada en el presente proyecto, habilita la generación de descripciones, inferencias y respuestas visualmente contextualizadas en función de preguntas o instrucciones.

- **Razonamiento avanzado sobre análisis de escenas**

  El uso de LLM como razonadores sobre las salidas de los modelos de análisis de escenas es una línea emergente. En este enfoque, el LLM opera a partir de información lingüística o estructurada ya generada, y realiza tareas de razonamiento adicionales como: resumen de la escena, extracción de conceptos clave, clasificación de situaciones, inferencia de acciones recomendadas, o generación de comandos para actuadores autónomos [48]. Esta capacidad permite transformar los sistemas de telepresencia inmersiva y robótica en plataformas cognitivas que no solo describen su entorno, sino que pueden actuar o asistir de forma adaptativa [49]. En el presente trabajo, esta funcionalidad se implementa como un componente clave dentro de la arquitectura distribuida cliente-servidor.

- **Evaluación basada en lenguaje natural (*LLM-as-a-judge*)**

  Una de las aplicaciones más recientes y prometedoras de los LLM es su uso como evaluadores semánticos. Frente a métricas tradicionales, los LLM permiten comparar salidas de modelos visuales mediante criterios semánticos, estilísticos o de adecuación contextual.

  Este método, conocido como *LLM-as-a-judge*, es empleado en este trabajo para comparar de forma cualitativa y contextualizada las descripciones generadas por distintos modelos de análisis de escenas. El procedimiento concreto se detalla en el capítulo correspondiente.

- **Generación y refinamiento de datos anotados**

  Los LLM también se utilizan para enriquecer bases de datos mediante «pseudoetiquetas», descripciones sintéticas o explicaciones automáticas. Este uso tiene implicaciones relevantes en contextos donde el etiquetado manual es costoso o limitado, y contribuye al desarrollo de modelos más explicables y auditables, especialmente en sectores como la medicina, la ingeniería industrial o el derecho.

Limitaciones y desafíos

A pesar de sus múltiples capacidades, los LLM presentan limitaciones importantes. El coste computacional de entrenamiento y uso es elevado; su razonamiento interno es opaco y difícil de auditar; y su fiabilidad puede verse afectada por sesgos implícitos en los datos de entrenamiento. Además, su

integración en flujos de visión plantea retos técnicos como la alineación semántica entre modalidades, la representación eficiente de imágenes y la generación controlada de salidas estructuradas.

Con todo, su capacidad para integrarse con entradas visuales, lingüísticas y estructuradas, los LLM constituyen una pieza clave en la arquitectura híbrida desarrollada en este proyecto.

2.1.6. Soporte *hardware* para el desarrollo de IA visual

La evolución de los modelos de inteligencia artificial en tareas de visión por ordenador ha estado estrechamente ligada al desarrollo de infraestructuras *hardware* especializadas. La capacidad de entrenar y desplegar redes profundas, con millones o incluso miles de millones de parámetros, depende críticamente del soporte computacional disponible, no solo en términos de potencia bruta, sino también de eficiencia energética, latencia, paralelismo y escalabilidad.

El campo de la visión por ordenador, en particular, se beneficia de ciertas capacidades propias del *hardware* moderno, como la aceleración de operaciones matriciales y convolucionales, la gestión eficiente de grandes volúmenes de datos y el mantenimiento del rendimiento en cargas de inferencia en tiempo real.

Unidades de procesamiento gráfico

Las GPU han sido la plataforma fundamental para el desarrollo del aprendizaje profundo. Originalmente diseñadas para renderizado gráfico, su arquitectura basada en miles de núcleos paralelos se adapta excepcionalmente bien al cómputo intensivo requerido por operaciones de convolución, multiplicación matricial y funciones de activación. Desde la introducción de CUDA (un *software* propio especializado) por parte de NVIDIA, las GPU han sido utilizadas de forma generalizada para entrenar redes neuronales profundas, tanto en investigación como en producción.

Modelos como AlexNet o ResNet se entrenaron sobre tarjetas como las NVIDIA Tesla, mientras que soluciones más recientes emplean arquitecturas avanzadas [52], que incorporan capacidades como gran ancho de banda de memoria y bibliotecas específicas. En escenarios distribuidos, se permite el entrenamiento simultáneo de modelos de gran escala.

Para tareas de inferencia en tiempo real y bajo consumo, especialmente en contextos de robótica o *edge computing*, destacan las GPU embebidas como las NVIDIA Jetson (Nano, Xavier, Orin), que permiten ejecutar modelos complejos con baja latencia, manteniendo la eficiencia energética.

Unidades tensoriales y aceleradores específicos

El crecimiento del tamaño y la complejidad de los modelos ha propiciado el desarrollo de aceleradores dedicados. Un ejemplo destacado son las TPU de Google, diseñadas para acelerar tanto el entrenamiento como la inferencia mediante núcleos especializados [52]. También han emergido aceleradores optimizados, como los desarrollados por Graphcore, Habana Labs o Apple, así como los integrados en SoC (del inglés *System on a Chip*, sistema en chip, nombre habitual para pequeños dispositivos integrados) móviles que permiten ejecutar modelos localmente, habilitando funcionalidades como reconocimiento facial, análisis de escenas o realidad aumentada sin necesidad de conexión a la nube.

Este ecosistema de aceleradores ha ampliado el abanico de posibilidades para el despliegue de IA visual en dispositivos personales, embebidos o industriales.

Infraestructura de sensores y adquisición de datos visuales

El análisis de escenas requiere no solo potencia de cálculo, sino también una entrada visual rica y estructurada. Por ello, los sensores utilizados juegan un papel determinante en la calidad y aplicabilidad del sistema, no solo como fuentes de información visual, sino también como parte esencial de un entorno perceptivo, tal como se plantea en el diseño global de este proyecto. Las cámaras RGB estándar son el sensor más común, pero su información es limitada a la intensidad y el color. Para enriquecer la percepción espacial, se utilizan sensores de profundidad como:

- Cámaras RGB-D, que combinan imagen y profundidad.
- Sistemas LIDAR, que generan nubes de puntos tridimensionales mediante láser.
- Cámaras estéreo, que estiman profundidad por triangulación.
- Sensores especializados (térmicos, hiperespectrales, de alta velocidad), aplicables en contextos médicos o industriales.

La combinación de múltiples sensores exige una integración precisa, incluyendo sincronización temporal, calibración espacial y fusión de datos, para lograr una percepción robusta frente a condiciones cambiantes y entornos dinámicos.

Sistemas embebidos y *edge computing*

En entornos como la robótica o la telepresencia inmersiva, el análisis de escenas debe realizarse localmente, con restricciones de tamaño, energía y latencia. El *edge computing* responde a esta necesidad mediante dispositivos capaces de ejecutar modelos en tiempo real sin depender de la nube.

Plataformas como Jetson, Coral o EdgeTPU permiten ejecutar redes comprimidas mediante técnicas como *quantization* o *pruning*, lo que facilita su uso en robots, drones o estaciones remotas. Estas soluciones resultan especialmente relevantes en sistemas distribuidos cliente-servidor como el desarrollado en este proyecto, donde la eficiencia del flujo de datos y el reparto de carga entre dispositivos y servidores condiciona el rendimiento global del sistema.

Impacto estratégico del *hardware* en la IA visual

En conjunto, la evolución del *hardware* es una condición *sine qua non* para el despliegue real de modelos de análisis de escenas. Sin la capacidad de ejecutar inferencia a la velocidad de recepción de cuadros, los modelos más sofisticados carecerían de aplicabilidad en escenarios donde el tiempo de respuesta es crítico.

Además, la disponibilidad de *hardware* embebido accesible, de sensores cada vez más precisos y de aceleradores cada vez más eficientes ha «democratizado» el acceso a tecnologías antes restringidas a centros de datos, permitiendo su adopción en entornos industriales, logísticos, educativos, médicos o incluso domésticos.

En el marco del presente proyecto, el diseño del sistema se apoya directamente en esta infraestructura: el robot zoomórfico, el dispositivo Búho y el servidor principal constituyen una arquitectura heterogénea, en la que la viabilidad del análisis de escenas depende tanto del modelo utilizado como del *hardware* que lo soporta. El éxito del sistema final es, por tanto, fruto de la interacción entre diseño algorítmico y soporte físico.

2.2. Profundización en el análisis de escenas

2.2.1. Fundamentos y alcance

Como se avanzó en el Capítulo 1, el análisis de escenas es uno de los principales ejes del trabajo. Se ha consolidado como un campo transversal que agrupa distintas técnicas avanzadas dentro de la visión por ordenador, con el objetivo común de dotar a los sistemas artificiales de una capacidad interpretativa estructurada y contextual del entorno visual.

Más allá de tareas individuales como la detección o segmentación, este campo articula un conjunto de enfoques que permiten modelar relaciones entre entidades, reconocer acciones, inferir intenciones y generar representaciones semánticas del espacio visual [20].

Entre las diversas tareas específicas que engloba este ámbito se incluyen [21]:

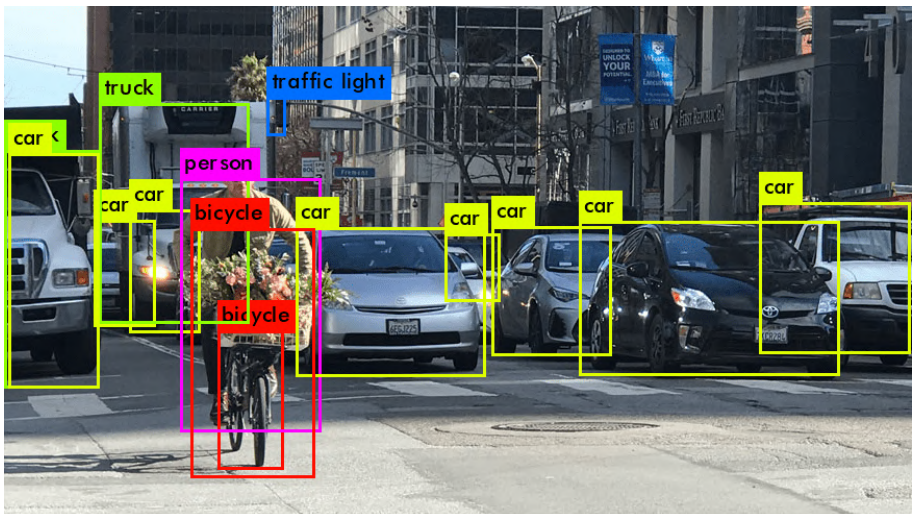
- **Clasificación de escenas:** asignación de una etiqueta general a una imagen, como «playa», «ciudad» o «interior», basándose en su contenido global.
- **Segmentación semántica:** asignación de una etiqueta a cada píxel de la imagen, identificando las diferentes regiones y los objetos presentes.
- **Reconocimiento de actividades:** identificación de acciones o eventos que ocurren en la escena, como «una persona caminando» o «un vehículo detenido».
- **Respuesta a preguntas visuales o VQA**: capacidad del sistema para responder preguntas formuladas en lenguaje natural sobre el contenido de una imagen o vídeo.
- **Descripción automática de cuadros** (más conocida por el inglés, ***captioning***): generación de descripciones en lenguaje natural que resumen el contenido visual de una imagen o secuencia de vídeo.

Estas tareas no se abordan de forma aislada, sino como partes interdependientes de un sistema interpretativo más amplio, en el que la integración de múltiples niveles de análisis visual resulta clave. Así, el análisis de escenas actúa como un puente entre la percepción visual básica y la interpretación significativa del entorno, integrando datos visuales en estructuras coherentes que permiten una interacción autónoma, contextual y adaptativa con el entorno. A modo de introducción, en la Figura 2.4 se muestra una comparativa entre una funcionalidad «básica» de la visión por ordenador, en ese caso la detección de objetos, con una funcionalidad propia del más avanzado análisis de escenas, en este caso, la descripción automática de una imagen.

El desarrollo de este campo continúa en rápida evolución, impulsado por la adopción de modelos generativos, arquitecturas multimodales y la necesidad creciente de dotar a los sistemas de una verdadera capacidad de comprensión visual. En este contexto, se ha vuelto imprescindible el uso de arquitecturas capaces de modelar relaciones complejas, lo que ha favorecido la transición desde enfoques tradicionales basados en redes convolucionales hacia modelos de atención y estructuras tipo *transformer*, como las descritas en el apartado 2.1. Esta evolución refleja una aspiración práctica y teórica por superar la fragmentación del análisis visual y avanzar hacia una representación capaz de unir percepción y significado.

En los siguientes apartados se revisan los principales modelos utilizados históricamente y los más recientes, clasificándolos en dos grandes grupos: modelos clásicos basados en arquitecturas CNN

especializadas para tareas visuales específicas, y modelos multimodales y de VQA basados en *transformers*, así como un análisis comparativo que permite comprender qué tipo de modelo es más adecuado según el contexto de aplicación.



(a) Ejemplo de detección de objetos

The image is a movie poster for the film "Harry Potter and the Sorcerer's Stone" starring Ron Weasley. The poster features a young boy, Ron Weasley, standing in front of a dark background with a castle-like structure in the background. He is wearing a school uniform with a Gryffindor crest on his jacket and a red tie. He has blonde hair and is looking directly at the camera with a serious expression on his face. The title of the film is written in white text at the top of the poster, with the tagline "20 years of movie magic" written in smaller text below.

(b) Ejemplo de descripción de una imagen generado por IA

Figura 2.4. Comparación entre visión por ordenador «básico» y análisis de escenas avanzado

2.2.2. Evolución y tipología: Modelos clásicos

Durante años, las tareas de análisis visual estructurado se abordaron mediante arquitecturas orientadas a resolver problemas concretos como la detección de objetos, la segmentación de instancias o la segmentación semántica. Como se explicó en el apartado 2.1.2, las redes neuronales convolucionales constituyen la base de estas arquitecturas, actuando como extractoras de características. En este apartado se revisan las variantes y especializaciones que han definido los modelos clásicos en tareas específicas de análisis de escenas.

Uno de los modelos más influyentes en detección de objetos es YOLO (*You Only Look Once*, Solo se mira una vez), introducido por [53]. Este modelo propuso por primera vez una arquitectura unificada que procesa un cuadro completo en una única pasada de red, generando simultáneamente las coordenadas de los recuadros y las probabilidades de clase. Su enfoque en tiempo real lo convirtió en el estándar de facto para tareas de localización rápida, siendo utilizado en robótica, videovigilancia y sistemas embarcados. Un ejemplo de este algoritmo es el que puede apreciarse en la Figura 2.4a, presentada en el epígrafe anterior.

En tareas de segmentación de instancias, el modelo Mask R-CNN [54] añadió una rama para predicción de máscaras binarias a nivel de objeto. Este modelo logra segmentar objetos individualmente y generar tanto su ubicación como su forma precisa, lo que resulta esencial para aplicaciones donde la separación entre instancias de la misma clase es relevante.

Para segmentación semántica, es decir, etiquetar cada píxel de una imagen con su clase correspondiente, destacan modelos como DeepLabv3+ [55], que emplean convoluciones dilatadas para ampliar el campo receptivo sin perder resolución espacial, y estructuras específicas para mejorar la precisión en bordes y zonas ambiguas.

Estos modelos clásicos presentan como principal ventaja su madurez, disponibilidad de implementaciones optimizadas y amplio respaldo empírico. Sin embargo, su diseño modular y especializado los hace poco adaptables a tareas abiertas o contextos donde se requiere interpretación semántica, multimodalidad o interacción con lenguaje. Además, su capacidad para modelar relaciones espaciales o funcionales entre objetos está limitada a lo que se pueda aprender implícitamente mediante convoluciones locales.

2.2.3. Evolución y tipología: Modelos multimodales y VQA

Como se ha mencionado en apartados anteriores, la integración de visión y lenguaje ha dado lugar a una nueva generación de modelos capaces no solo de detectar y clasificar objetos, sino también de razonar sobre ellos, describir relaciones, responder preguntas o emitir interpretaciones contextualizadas. Estas capacidades, clave para el análisis avanzado de escenas, han sido posibles gracias al uso de arquitecturas *transformer* que permiten procesar de manera conjunta secuencias mixtas de texto e imagen. Un ejemplo visual es el de la Figura 2.4b.

Uno de los primeros modelos de referencia en este ámbito fue CLIP, desarrollado por OpenAI [9]. CLIP entrena de forma contrastiva un codificador visual y un codificador textual, maximizando la similitud entre las representaciones latentes de un cuadro y su descripción correspondiente. Esta técnica permite utilizar texto como consulta para recuperar imágenes, clasificar contenidos o contextualizar escenas, sin requerir un entrenamiento supervisado específico para cada tarea.

Posteriormente, modelos como ViLT propusieron una arquitectura aún más integrada, en la que los parches de imagen se proyectan directamente a una secuencia conjunta con los *tokens* de texto, que es procesada por un único *transformer*. Aunque este enfoque sacrifica parte de la precisión en tareas puramente visuales, permite un mayor grado de fusión temprana entre modalidades y una reducción significativa de parámetros [56].

Otro avance relevante es Florence, un modelo desarrollado por Microsoft que unifica múltiples tareas visuales (clasificación, detección, descripción automática o *captioning*, VQA…) bajo un marco común de entrenamiento y evaluación [51]. Florence utiliza un potente codificador visual y una capa de alineamiento semántico para generar salidas lingüísticas adaptadas al contexto visual, lo que lo convierte en un modelo versátil para análisis de escenas complejas. Además, otros *frameworks* como LAVIS (*Language and Vision Interface for Systems*) [57] han ganado relevancia por su capacidad de ofrecer una interfaz unificada para tareas de visión y lenguaje, integrando distintos modelos de forma flexible y optimizada para tareas como la descripción automática o el VQA.

Más recientemente, modelos como LLaVA [58] han conectado un codificador visual con un modelo lingüístico de gran escala, como Vicuna o GPT, mediante una capa de proyección que actúa como interfaz. Esta combinación permite procesar instrucciones complejas en lenguaje natural, realizar

razonamiento multimodal y generar respuestas con una fluidez comparable a los asistentes conversacionales actuales. En la misma línea, la iniciativa OpenBMB (*Open Big Model Base*) [59] ha desarrollado modelos avanzados y de código abierto orientados específicamente a la generación de descripciones de vídeo y razonamiento visual, con un fuerte enfoque en la adaptabilidad a diferentes dominios y formatos.

En el marco del presente trabajo, se ha experimentado con varios de estos modelos, evaluando su capacidad para representar escenas, responder preguntas sobre el entorno, generar descripciones precisas y adaptarse a distintas fuentes visuales (cuadros o secuencias de vídeo). En particular, Florence, ViLT, LLaVA-Next-Video, LAVIS y los modelos de OpenBMB han sido seleccionados como herramientas de referencia, tanto por su rendimiento como por su flexibilidad en entornos distribuidos, como se detallará a lo largo de esta memoria.

2.2.4. Evolución y tipología: Análisis comparativo conceptual de modelos

La elección del modelo más adecuado para el análisis de escenas depende en gran medida del contexto de aplicación, los recursos computacionales disponibles y el tipo de información que se requiere extraer del entorno. En términos generales, pueden establecerse las siguientes comparaciones:

En aplicaciones industriales con requisitos estrictos de latencia y tareas bien definidas, los modelos clásicos siguen siendo una opción eficaz, debido a su madurez, predictibilidad y eficiencia computacional. No obstante, en entornos no estructurados, sistemas interactivos o tareas de interpretación cualitativa, los modelos multimodales emergen como la alternativa más prometedora.

Desde una perspectiva estratégica, los modelos multimodales más avanzados tienden a consolidarse como *foundation models* en visión por ordenador, gracias a su flexibilidad, capacidad de adaptación y potencial de transferencia entre tareas. Su integración con LLM habilita un análisis contextualizado y dialógico del entorno visual, lo que abre nuevas oportunidades en ámbitos como la asistencia remota, la formación inmersiva, la accesibilidad o la robótica colaborativa.

En el marco de este proyecto, esta comparación justifica la selección de arquitecturas multimodales y VQA como base para el sistema desarrollado, al ser las que mejor se adaptan a los objetivos de interpretación semántica, interacción y despliegue en entornos distribuidos.

2.3. Evaluación de modelos de visión artificial

La evaluación rigurosa del rendimiento de un modelo de visión por ordenador es un componente crítico en cualquier proyecto aplicado, especialmente en el ámbito del análisis de escenas. A diferencia de tareas más simples como la clasificación, donde el resultado es un único valor, las tareas de detección, segmentación o razonamiento visual requieren evaluar estructuras complejas: múltiples objetos, relaciones, contexto, y en muchos casos, respuestas abiertas que no tienen una única solución correcta.

Este apartado revisa los principales enfoques de evaluación, comenzando por las métricas clásicas objetivas y cuantificables, y continuando con métodos cualitativos y semánticos más adecuados para tareas abiertas y modelos multimodales.

2.3.1. Métricas de evaluación clásicas

Como primera aproximación, la evaluación de los modelos aplicados a tareas de visión artificial con salida lingüística, como la descripción de cuadros o el *Visual Question Answering*, se ha basado tradicionalmente en métricas puramente estadísticas. Evaluarlas sigue siendo un componente esencial para medir su rendimiento y establecer comparaciones significativas. No obstante, se trata de un proceso complejo, ya que los resultados generados no siempre son comparables mediante métricas objetivas tradicionales, especialmente en tareas abiertas donde puede haber múltiples respuestas válidas.

En el caso de la generación de descripciones de cuadros, las métricas clásicas utilizadas históricamente han sido de tipo léxico o basadas en n-gramas, como BLEU (del inglés *BiLingual Evaluation Understudy*, Estudio de Evaluación Bilingüe), ROUGE-L (de *Recall-Oriented Understudy for Gisting Evaluation*, que significa Estudio para la evaluación de la esencia orientado al *recall*), METEOR (sigla de *Metric for Evaluation of Translation with Explicit ORdering*, Métrica para evaluar la traducción con orden explícito) y CIDEr (de *Consensus-based Image Description Evaluation*, Evaluación de la descripción de imágenes basada en el consenso). Estas métricas comparan la salida generada con una o varias referencias humanas, midiendo el grado de coincidencia entre fragmentos de texto (palabras o secuencias cortas). Aunque son fácilmente computables y ampliamente adoptadas, presentan importantes limitaciones: tienden a penalizar la creatividad, no captan bien el significado semántico profundo, y son especialmente sensibles al orden de las palabras o a la literalidad de las expresiones.

Estudios recientes, como [60], han demostrado empíricamente que muchas de estas métricas no son capaces de distinguir entre descripciones humanas de calidad y otras que han sido distorsionadas de forma artificial (mediante sustitución o reordenación de palabras). Por ejemplo, métricas como BLEU y ROUGE-L mostraron escasa correlación con el juicio humano en estos escenarios, mientras que CIDEr ofrece un rendimiento algo más robusto, aunque insuficiente ante ciertas variaciones lingüísticas o semánticas.

En el caso del VQA, según se recoge en el estudio de referencia [61], las métricas predominantes han sido tradicionalmente más simples, basadas en el acierto exacto o en la coincidencia parcial con un conjunto limitado de respuestas humanas, como la exactitud VQA (normalmente referido en inglés, VQA *Accuracy*). Algunas variantes, como WUPS (Wu-Palmer *Similarity*, es decir, semejanza de Wu-Palmer) Score o Valor F1 (de nuevo, referido a menudo por el inglés F1 *Score*), han buscado refinar la evaluación teniendo en cuenta la razonabilidad o la cobertura de las respuestas, pero siguen enfrentándose a dificultades similares para capturar el razonamiento contextual o las inferencias implícitas esperadas en tareas abiertas; sin embargo, al igual que en descripción automática, insuficientes para captar la validez contextual o el razonamiento subyacente en respuestas abiertas, especialmente cuando se espera que el sistema interprete relaciones funcionales o inferencias implícitas.

En conclusión, el uso de métricas clásicas como BLEU o METEOR sigue siendo común por motivos históricos y de comparación con trabajos previos, pero su relevancia es cada vez más cuestionada. Los trabajos recientes evidencian la necesidad de adoptar métricas más sofisticadas que tengan en cuenta la semántica, la sintaxis y el vínculo real entre texto e imagen, especialmente en tareas como el análisis de escenas. El presente proyecto adopta esta perspectiva, incorporando métricas embebidas en la evaluación de modelos, tanto para descripción de cuadros como para VQA.

2.3.2. Evaluación empleando LLM

A medida que los modelos de análisis de escenas avanzan hacia tareas más complejas y abiertas, como la generación de descripciones, el razonamiento visual o el *Visual Question Answering*, las métricas

clásicas resultan insuficientes para valorar adecuadamente la calidad de sus salidas. Estas tareas no tienen una única solución correcta, y exigen una evaluación que considere aspectos como la coherencia semántica, la relevancia contextual o la fluidez lingüística.

Para abordar estas limitaciones, han surgido enfoques de evaluación semántica asistida, que combinan técnicas cualitativas con sistemas automatizados basados en LLM.

Evaluación con *LLM-as-a-judge*

El enfoque conocido como *LLM-as-a-judge* consiste en utilizar modelos lingüísticos avanzados (como GPT-4 o DeepSeek) para comparar y valorar salidas generadas por sistemas de visión artificial. Estos modelos pueden actuar como jueces semánticos, evaluando si una descripción o respuesta es coherente, precisa, relevante o mejor que otra en un contexto dado. Su capacidad para manejar lenguaje natural permite definir criterios de evaluación más flexibles y cercanos al juicio humano.

Recientes estudios han demostrado que los LLM pueden superar a las métricas clásicas en tareas como generación automática de descripciones o VQA, logrando correlaciones más altas con evaluaciones humanas y detectando errores conceptuales que pasarían desapercibidos con métricas tipo BLEU o CIDEr [60]. Herramientas como Pixtral o DeepSeek implementan este enfoque de forma sistemática, proporcionando plataformas reproducibles para evaluación cualitativa a gran escala. Entre algunas de las ventajas más significativas, se pueden destacar:

- Permite evaluar contenido semántico, contexto, adecuación estilística y plausibilidad.
- Evita depender de etiquetas únicas, considerando la naturaleza ambigua o subjetiva de algunas escenas.
- Es escalable, pudiéndose aplicar a miles de ejemplos automáticamente.
- Es ajustable por entrada de texto (lo que se conoce habitualmente por el inglés *prompt*), lo que permite enfatizar precisión, cobertura, relevancia o lenguaje técnico según el uso.

Sin embargo, no está exento de limitaciones y riesgos, especialmente si se tiene en cuenta la poca madurez de las tecnologías empleadas:

- Sensibilidad al *prompt*: pequeños cambios en la formulación pueden modificar el criterio de evaluación.
- Posible sesgo del LLM: la preferencia puede estar condicionada por sesgos entrenados.
- Falta de trazabilidad: difícil interpretar por qué se eligió una respuesta, salvo en modos explicativos.
- Falta de consistencia entre evaluaciones paralelas.

Pese a estas limitaciones, *LLM-as-a-judge* ha sido adoptado como estándar en muchos entornos de evaluación modernos, incluyendo su uso en el desarrollo de *benchmarks* abiertos como OpenEval, así como plataformas comerciales de ranking visual y textual.

Evaluación cualitativa y contextual

Además del uso de LLM como evaluadores automáticos, existen técnicas cualitativas que permiten obtener una visión más rica y matizada del comportamiento de los modelos [60], [61]:

- Visualización de activaciones y mapas de atención, que permiten verificar si el modelo «mira» las regiones correctas de la imagen al generar su salida.
- Clasificación manual de errores, agrupando los fallos en tipologías interpretables: errores de contexto, confusión semántica, fallos sistemáticos, etc.
- Evaluaciones con usuarios o jurados expertos, especialmente útiles en aplicaciones interactivas o inmersivas (telepresencia, accesibilidad).
- Descomposición estructural, analizando por separado la detección de objetos, la inferencia de relaciones y la generación lingüística en tareas complejas.

En conjunto, la combinación de *LLM-as-a-judge* y evaluación cualitativa aporta un enfoque más completo y adaptado al carácter abierto y contextual del análisis de escenas, especialmente en entornos como la robótica distribuida o la interacción humano-máquina. Estas estrategias no sustituyen a las métricas objetivas, sino que las complementan. Permiten detectar problemas que no se reflejan en puntuaciones agregadas, facilitan el diagnóstico de modelos, y sientan las bases para el diseño de soluciones más robustas, interpretables y centradas en el usuario.

En el presente trabajo, se empleará la metodología *LLM-as-a-judge* para comparar descripciones generadas por distintos modelos de análisis de escenas, asignar puntuaciones a sus salidas en lenguaje natural, y obtener una visión más justa y contextualizada del rendimiento de cada modelo.

2.4. Sistema cognitivo para análisis de escenas

2.4.1. Enfoque del sistema

El desarrollo de sistemas artificiales capaces de operar en entornos complejos y cambiantes requiere avanzar más allá de la mera percepción visual. Como se expuso en el Capítulo 1, tareas como la robótica móvil, la telepresencia inmersiva o la interacción hombre-máquina exigen dotar a los sistemas de capacidades cognitivas que integren percepción, comprensión, razonamiento y comunicación. Sobre esta base, el presente trabajo adopta un enfoque cognitivo que busca articular un comportamiento autónomo, flexible y contextualizado [62].

Este paradigma se articula frecuentemente en torno al marco conceptual *Sense-Think-Act* (en inglés, «Sentir-Pensar-Actuar»), ampliamente adoptado en robótica cognitiva y en sistemas de inteligencia artificial distribuida [63]. En este esquema, la capa de «Sentir» comprende los subsistemas de percepción, encargados de procesar los flujos de datos sensoriales (visión, audio, sensores de entorno) y de generar representaciones estructuradas del estado del mundo; la capa de «Pensar» incorpora módulos de razonamiento y aprendizaje que operan sobre dichas representaciones, permitiendo inferencias, planificación y generación de respuestas; mientras que la capa de «Actuar» materializa las decisiones a través de actuadores físicos o interfaces de interacción, cerrando el bucle cognitivo.

Dentro de este marco, un sistema cognitivo se define como aquel que es capaz de realizar de manera integrada las siguientes funciones fundamentales:

- **Percepción:** captación y procesamiento de datos sensoriales multimodales.
- **Comprensión:** transformación de datos en representaciones semánticas coherentes.
- **Razonamiento:** inferencia sobre las representaciones internas, identificación de patrones y relaciones.
- **Aprendizaje:** mejora progresiva de las capacidades mediante experiencia o supervisión.
- **Toma de decisiones:** selección de acciones apropiadas en función de objetivos y contexto.
- **Comunicación:** generación de respuestas comprensibles y contextualizadas, tanto hacia humanos como hacia otros sistemas.

La investigación en arquitecturas cognitivas ha sido impulsada por múltiples comunidades, desde la robótica cognitiva hasta la inteligencia artificial general. Modelos como SOAR u otras arquitecturas más recientes basadas en aprendizaje profundo y redes neuronales multimodales ilustran esta evolución [64]. En paralelo, la integración de LLM como capa de razonamiento y comunicación está redefiniendo el diseño de sistemas cognitivos contemporáneos [42], como se explorará en los apartados siguientes.

El enfoque cognitivo es particularmente relevante en campos como la telepresencia inmersiva y la robótica avanzada, donde la necesidad de interpretar entornos complejos, interactuar de manera fluida y proporcionar realimentación contextualizada exige una integración estrecha entre percepción, razonamiento y comunicación. Los sistemas que adoptan este enfoque no se limitan a actuar como «sensores avanzados», sino que se comportan como agentes cognitivos autónomos capaces de enriquecer la experiencia del usuario y de adaptar su comportamiento a situaciones cambiantes, como se detallará en los siguientes apartados a través de la arquitectura desarrollada en este trabajo.

2.4.2. Arquitectura modular

El desarrollo de sistemas cognitivos aplicados a la percepción visual y al razonamiento se apoya de forma creciente en arquitecturas modulares. Este enfoque permite estructurar el procesamiento en capas diferenciadas, lo que facilita la escalabilidad, la integración de componentes heterogéneos y la adaptación a contextos de aplicación diversos [65]. En particular, en ámbitos como la robótica cognitiva y la telepresencia inmersiva, el diseño modular resulta esencial para gestionar la complejidad de los sistemas distribuidos y para garantizar la interoperabilidad entre subsistemas.

Una arquitectura cognitiva modular típica para análisis de escenas y razonamiento integra, de manera jerárquica, al menos tres capas funcionales principales: percepción, razonamiento y acción. Esta estructura responde conceptualmente al paradigma *Sense-Think-Act*, ya introducido, que ha sido validado ampliamente en el diseño de agentes autónomos y sistemas interactivos [63].

La capa de percepción se encarga de procesar los datos sensoriales y de transformar las señales brutas en representaciones estructuradas. En el contexto del análisis de escenas, esta capa comprende módulos de visión por ordenador capaces de realizar tareas como la inferencia de dinámicas y eventos en la escena [66]. El objetivo de esta etapa es generar una estructura semántica intermedia que sirva como base para el razonamiento posterior.

Sobre esta representación actúa la capa de razonamiento, que integra LLM junto con otros posibles componentes simbólicos o neuronales. Esta capa opera sobre la estructura semántica generada por la percepción, permitiendo realizar inferencias, responder a consultas en lenguaje natural, generar explicaciones o planificar secuencias de acción [42].

Finalmente, la capa de acción articula las salidas del sistema. En aplicaciones de robótica cognitiva, esta capa puede controlar actuadores físicos, mientras que, en contextos de telepresencia, puede gestionar la presentación de la información hacia el usuario remoto, mediante sistemas de diálogo o entornos XR. La existencia de una capa diferenciada de acción permite cerrar el ciclo cognitivo y posibilita una interacción dinámica y adaptativa.

Cada vez con mayor frecuencia, en los sistemas avanzados estas capas se implementan de manera distribuida. En particular, las funciones de percepción y razonamiento no residen en un dispositivo único, sino que se implementan en infraestructuras de procesamiento en la nube o en servidores especializados, mientras que la capa de acción puede estar físicamente distribuida entre dispositivos de usuario, plataformas robóticas o sistemas de red [67]. Este enfoque permite aprovechar recursos computacionales escalables y facilita la integración de modelos de última generación, que requieren capacidades de cómputo considerables.

El diseño modular y distribuido no solo responde a necesidades de eficiencia, sino que habilita nuevas capacidades cognitivas. La separación de las capas permite una evolución independiente de cada componente, la incorporación de nuevos métodos de aprendizaje o razonamiento y la integración flexible de múltiples modalidades sensoriales, como visión, audio o datos hápticos. Además, facilita el cumplimiento de requisitos específicos de latencia y privacidad en función de la aplicación.

En conjunto, la filosofía modular ofrece un marco robusto para la integración del análisis de escenas con capacidades avanzadas de razonamiento. Este enfoque está siendo adoptado en múltiples líneas de investigación, desde los sistemas de interacción hombre-robot hasta las plataformas de telepresencia cognitiva y los sistemas XR inteligentes [66]. En este contexto, el rol específico de los LLM como componente de razonamiento adquiere una relevancia estratégica, como se explora en el apartado siguiente.

2.4.3. Rol de los LLM junto al análisis de escenas

Así, una de las direcciones más prometedoras en el campo de los sistemas cognitivos es la integración entre mecanismos de percepción visual avanzada y LLM. Esta colaboración ha comenzado a consolidarse como un método efectivo para dotar a las máquinas de capacidades más cercanas al razonamiento humano, sin que exista solapamiento funcional entre los módulos involucrados. Lejos de representar una redundancia, la combinación de análisis de escenas y LLM configura una estructura de procesamiento modular en la que cada componente ejerce funciones diferenciadas pero complementarias.

El análisis de escenas se inscribe en la capa de percepción, encargada de transformar la entrada sensorial, habitualmente cuadros o vídeo, en representaciones estructuradas del entorno. Estas representaciones contienen información sobre objetos, relaciones espaciales, atributos, acciones visibles y otras características perceptibles. En arquitecturas avanzadas, como las contempladas en este trabajo, el análisis de escenas ya produce como salida un texto en lenguaje natural perfectamente estructurado, sintáctica y semánticamente correcto, que describe de forma comprensible para humanos lo que ocurre en la escena. Sin embargo, aunque este texto constituye una descripción completa del entorno, no basta

por sí solo para seleccionar la información más relevante, adaptarla al contexto o generar razonamientos o instrucciones específicas en función de la situación o de las necesidades del usuario.

Aquí es donde entran en juego los LLM, en la capa de razonamiento, que toma como entrada este texto estructurado y lo procesa a la luz de un modelo lingüístico y conceptual entrenado sobre una enorme base de conocimiento. Estos modelos no «traducen» la salida del análisis de escenas, que ya es perfectamente legible, sino que operan sobre ella para filtrar la información más relevante, inferir intenciones, responder preguntas, generar instrucciones u órdenes o resumir aspectos clave, en función de la consulta o el flujo de interacción requerido [42]. Su capacidad para vincular conceptos, integrar contexto y razonar sobre el contenido textual los convierte en un complemento ideal para esta arquitectura modular.

Desde una perspectiva arquitectónica, esta colaboración se apoya en el enfoque modular que caracteriza a muchos sistemas cognitivos actuales. La información fluye desde la percepción hacia la acción, pasando por etapas bien definidas: percepción (visión por computador, sensores), representación semántica (estructuración de lo percibido), razonamiento (interpretación, predicción, planificación) y acción/comunicación (respuesta o ejecución). Así, mientras el análisis de escenas responde a la pregunta «¿qué hay ahí y cómo está organizado?», el LLM aborda «¿qué significa y qué debo hacer o decir a continuación?».

Una analogía ilustrativa es el propio sistema cognitivo humano: la percepción visual organiza la información, pero el razonamiento consciente integra contexto, conocimiento previo e intenciones para decidir cómo actuar o qué comunicar: por ejemplo, «una persona está de pie frente a una puerta abierta, con una caja en la mano». Esta información, al llegar a áreas como la corteza prefrontal o al sistema del lenguaje, puede dar lugar a inferencias como «parece que va a salir llevando la caja» o incluso a respuestas verbales si alguien pregunta qué está ocurriendo. El cerebro no necesita que la percepción haga todo el trabajo de inferencia, sino que lo percibido se organice bien para ser interpretado, resumido o contextualizado con flexibilidad. Esta es exactamente la función que ejercen los LLM en sistemas artificiales [68].

Además, los LLM no solo interpretan, sino que integran contexto. Pueden considerar, junto con el texto proveniente del análisis de escenas, factores como el historial de la conversación, el objetivo del sistema, las preferencias del usuario o el conocimiento general sobre el entorno. Esto permite una adaptación más sofisticada de la respuesta y una mayor coherencia en la interacción. Por ello, los LLM se utilizan cada vez más como núcleos de razonamiento general que interactúan con múltiples fuentes de percepción, entre ellas el análisis de escenas, pero también sensores auditivos, texto y otras modalidades.

El interés por este enfoque híbrido se ha materializado en múltiples líneas de investigación recientes. Algunos sistemas emplean representaciones explícitas derivadas de la escena, como árboles semánticos o descripciones relacionales, que luego son integradas como entrada textual para los LLM. Otros optan por arquitecturas que combinan directamente textos completos generados por el análisis de escenas con entradas de texto, aprovechando la capacidad de estos modelos para seleccionar, resumir y razonar sobre la información disponible.

Este tipo de sistemas se aplica ya en campos como la robótica cognitiva, donde los LLM se encargan de interpretar lo percibido y generar comandos simbólicos o explicaciones, y en sistemas de asistencia remota, donde lo importante no es solo percibir correctamente, sino ser capaz de extraer, adaptar y comunicar la información más relevante de forma comprensible y ajustada al interlocutor. En todos

estos casos, el análisis de escenas y LLM permite alcanzar un nivel de razonamiento e interacción que aproxima estos sistemas a una lógica cognitiva flexible y contextual.

Este enfoque seguirá siendo clave en el avance hacia sistemas más robustos, explicables y útiles para la interacción natural con humanos, como se propone en el sistema desarrollado en el presente trabajo.

2.5. Robótica y percepción visual

La robótica moderna requiere capacidades avanzadas de percepción para operar de forma segura, eficiente y adaptativa. En el contexto de sistemas cognitivos modulares, la percepción visual actúa como un componente esencial dentro del ciclo *Sense-Think-Act*, proporcionando al sistema representaciones estructuradas y comprensibles del entorno que alimentan los procesos de razonamiento y decisión. En este marco, el análisis de escenas se incorpora como una fuente clave de información, no solo para la navegación autónoma o la manipulación precisa, sino también para habilitar una interacción contextualizada y fluida entre humanos y robots. Este apartado expone las principales necesidades visuales en entornos robóticos, su impacto sobre el comportamiento del sistema cognitivo, y ejemplos de aplicaciones reales, con especial atención al uso de robots zoomórficos.

2.5.1. Percepción visual en robots autónomos

La percepción visual es un componente esencial en los sistemas robóticos autónomos, ya que permite a los robots interpretar su entorno a partir de datos sensoriales complejos, como cuadros o secuencias de vídeo. Esta capacidad es fundamental para tareas como la navegación, la planificación de trayectorias, la manipulación de objetos o la interacción con personas, en las que el entorno no está previamente modelado o presenta un alto grado de variabilidad.

A diferencia de otros sensores, como el LIDAR o los ultrasonidos, la visión proporciona una información rica y semánticamente densa, que permite no solo detectar obstáculos o identificar formas, sino también comprender la estructura funcional de la escena, interpretar relaciones espaciales y anticipar acciones o eventos. Este valor semántico es clave para que los sistemas cognitivos puedan realizar razonamientos adaptativos sobre su entorno visual. Esta riqueza interpretativa ha llevado a que la percepción visual se considere una de las vías más prometedoras para el desarrollo de autonomía avanzada en robótica móvil y de servicio [69].

En sistemas robóticos autónomos modernos, el flujo de percepción visual abarca desde la adquisición de datos brutos (imágenes RGB, vídeo, profundidad) hasta la generación de representaciones internas que alimentan los módulos de decisión. Esto puede incluir componentes como la detección de objetos, la segmentación semántica, la estimación de postura, el seguimiento de agentes y, cada vez más, el análisis de escenas. Este último permite ir más allá de la mera identificación de elementos visuales, ofreciendo interpretaciones estructuradas y contextualizadas de lo que ocurre en el entorno: qué objetos hay, cómo se relacionan, qué acciones están en curso o quién interactúa con qué. En arquitecturas tipo *Sense-Think-Act*, estas funcionalidades permiten cerrar el ciclo de percepción-razonamiento-acción.

Uno de los retos fundamentales de la percepción visual en robótica es el equilibrio entre precisión, robustez y latencia. La ejecución de modelos de análisis de escenas en entornos reales impone restricciones computacionales y temporales importantes, especialmente en plataformas móviles donde los recursos de procesamiento son limitados. En este contexto, se han desarrollado estrategias como la compresión de modelos, la inferencia acelerada por *hardware* o el procesamiento distribuido cliente-

servidor, como el implementado en este trabajo, que permiten mantener una percepción rica sin comprometer la operatividad del sistema [70].

Desde un punto de vista funcional, la integración del análisis de escenas en la percepción visual robótica aporta una capa semántica crítica que habilita formas de interacción más complejas y adaptativas. Estas representaciones semánticas actúan como insumo fundamental para los LLM u otros módulos de razonamiento, permitiendo que el sistema no solo perciba, sino que también interprete, planifique y comunique de manera más inteligente, permitiendo al robot no solo reaccionar ante estímulos, sino interpretar su contexto, tomar decisiones con base en el significado de la escena y colaborar con humanos de manera más fluida y segura. Esta capacidad es especialmente relevante en aplicaciones como la asistencia remota, la exploración no estructurada o la interacción social, donde la interpretación visual no puede limitarse a una clasificación básica o a la detección de objetos aislados.

En el presente trabajo, la adopción del análisis de escenas integrado en un sistema cognitivo como componente central de la percepción visual se justifica por su capacidad para dotar al robot de una comprensión contextual rica y generalizable, compatible con entornos dinámicos y heterogéneos. Esta capacidad permite además integrar otros modos de entrada visual (como vídeo 360° o cámaras egocéntricas) y adaptar el sistema a escenarios de colaboración humano-robot distribuidos, en los que el entendimiento compartido del entorno es clave para el éxito de la tarea.

2.5.2. Aplicaciones de análisis de escenas en robótica

El análisis de escenas ha adquirido un papel central en muchas de las aplicaciones avanzadas de robótica, especialmente en aquellas que requieren interacción con entornos dinámicos o cooperación con humanos. A través de esta capacidad, los robots no solo detectan y clasifican objetos, sino que pueden interpretar relaciones espaciales, inferir intenciones, anticipar acciones y tomar decisiones basadas en el contexto general de la escena.

Una de las aplicaciones más consolidadas es la navegación autónoma en entornos complejos, donde el análisis de escenas permite identificar rutas transitables, obstáculos dinámicos, zonas de interés o comportamientos humanos que requieren adaptación, como el paso de peatones o la interacción en espacios compartidos. En este tipo de tareas, el modelo no se limita a una segmentación plana del entorno, sino que genera representaciones estructuradas que incluyen elementos funcionales (mesas, puertas, caminos, personas en movimiento) y su significado en la escena. Estas representaciones, tras ser utilizadas por módulos de razonamiento, podrán emplearse para planificar trayectorias, anticipar conflictos o generar explicaciones comprensibles para operadores humanos.

Otra aplicación destacada es la manipulación robótica basada en contexto, donde el robot debe no solo localizar un objeto, sino entender su función, su relación con otros objetos y la secuencia de acciones necesaria para manipularlo correctamente. En escenarios como cocinas, laboratorios o líneas de ensamblaje flexibles, esta capacidad resulta crítica para operar con autonomía y seguridad [71].

En robótica colaborativa y social, la percepción visual se utiliza para detectar intenciones humanas, reconocer gestos, interpretar acciones y adaptar la respuesta del robot en función de señales contextuales. Esto es especialmente útil en tareas de asistencia, educación o interacción natural, donde el robot debe actuar de forma comprensible y proactiva. Por ejemplo, un robot que detecta que una persona se inclina hacia un objeto puede anticiparse y ofrecer asistencia, o uno que reconoce un entorno educativo puede adaptar su lenguaje y comportamiento al perfil del usuario.

Asimismo, en entornos de exploración remota o telepresencia inmersiva, como los que aborda este trabajo, la percepción visual permite sintetizar y transmitir al operador humano una comprensión rica

del entorno, que va más allá del simple envío de vídeo en tiempo real. Mediante la generación de descripciones, la respuesta a preguntas o la detección semántica de eventos, se mejora la eficacia de la supervisión remota y se reduce la carga cognitiva del usuario. Esta capacidad resulta especialmente útil en escenarios de inclusión, donde personas con dificultades de movilidad o acceso pueden beneficiarse de una interfaz visual enriquecida.

Por último, en aplicaciones industriales, logísticas y agrícolas, el análisis de escenas permite desarrollar sistemas más adaptativos y menos dependientes de entornos controlados. Robots equipados con esta capacidad pueden operar en almacenes cambiantes, identificar productos mal colocados, interpretar señales visuales del entorno y adaptarse a nuevas configuraciones sin necesidad de reprogramación específica [70].

En el presente trabajo, la integración del análisis de escenas se plantea como una solución transversal a múltiples desafíos: dotar de mayor autonomía al robot zoomórfico, facilitar la interacción con el entorno mediante lenguaje natural y permitir la supervisión remota eficiente en sistemas distribuidos, dotándolos de una percepción visual que no solo es rica y estructurada, sino también accionable y comprensible. La versatilidad de esta técnica la convierte en un componente estratégico para escenarios donde la percepción visual necesita ser semántica, flexible y adaptativa.

2.5.3. Robótica zoomórfica y su relevancia tecnológica

La robótica zoomórfica, caracterizada por el uso de formas inspiradas en animales, representa una de las líneas más punteras dentro del desarrollo de robots móviles orientados a entornos complejos, no estructurados o de interacción social. Este enfoque no solo persigue ventajas mecánicas, como la estabilidad, la capacidad de desplazamiento en terrenos irregulares o la versatilidad, sino también una mejor integración funcional y social con los entornos donde operan los robots.

Desde el punto de vista técnico, los robots con morfología animal, como los de tipo «perro», ofrecen una plataforma ideal para la exploración autónoma, la inspección remota y la asistencia móvil, gracias a su equilibrio entre movilidad, carga útil y capacidad de adaptación. Esta movilidad avanzada, sin embargo, requiere de una percepción visual sofisticada que no solo detecte obstáculos, sino que interprete contextos complejos, estructuras funcionales y elementos dinámicos que condicionan la locomoción o las decisiones operativas [69].

En este marco, el análisis de escenas se posiciona como un complemento esencial para dotar de inteligencia visual contextual a estos sistemas, pues esta capacidad, integrada con las capas de razonamiento y actuación que se han explicado en los puntos anteriores, resulta especialmente relevante cuando el robot debe moverse de forma autónoma en entornos humanos, asistir en tareas colaborativas o servir como extensión remota de un operador.

Además de ello, la robótica zoomórfica encuentra aplicaciones en entornos sensibles o difíciles de alcanzar por operadores humanos, como zonas industriales, infraestructuras críticas, espacios naturales o situaciones de emergencia. En estos contextos, la percepción visual permite generar descripciones detalladas del entorno, emitir alertas visuales, responder preguntas o facilitar el control remoto mediante interfaces enriquecidas. Así, se maximiza el valor operativo del robot más allá del desplazamiento físico [69].

Desde el punto de vista de la interacción social, los robots zoomórficos pueden generar mayor aceptación e intuición de uso por parte de los humanos, especialmente en contextos educativos, asistenciales o de acompañamiento. El análisis visual permite que estos sistemas interpreten no solo el

entorno físico, sino también comportamientos humanos, señales gestuales o patrones de movimiento, lo cual habilita formas de colaboración más naturales y eficaces.

En el presente trabajo, el uso de una plataforma zoomórfica se aprovecha como soporte móvil versátil sobre el que desplegar capacidades avanzadas de análisis de escenas. Esta combinación permite explorar escenarios donde la movilidad física y la interpretación visual del entorno se integran en un único flujo operativo, y en los que la inteligencia contextual se convierte en una ventaja diferencial frente a enfoques más clásicos de percepción o navegación.

2.6. Telepresencia inmersiva: principios técnicos y aplicaciones

La telepresencia inmersiva tiene como objetivo recrear la experiencia subjetiva de «estar presente» en un entorno remoto, integrando múltiples tecnologías audiovisuales para lograr una percepción realista, coherente y continua del espacio a distancia. A diferencia de los sistemas clásicos de videoconferencia o supervisión remota, la telepresencia inmersiva busca maximizar el grado de inmersión sensorial, la interactividad en tiempo real y la integración contextual de la información visual.

Este tipo de sistemas se basa en una combinación de cámaras de alta resolución, dispositivos de captura 360°, sensores espaciales, transmisión de baja latencia y entornos virtuales o aumentados, que permiten al usuario explorar de forma activa y natural un entorno que no tiene físicamente a su alcance. Tecnologías como la visión estereoscópica o el *head tracking* (seguimiento de la posición de la cabeza) contribuyen a generar una sensación de presencia más envolvente.

Las aplicaciones actuales de la telepresencia inmersiva incluyen desde teleasistencia médica, formación técnica especializada e inspección remota de infraestructuras, hasta iniciativas sociales y educativas centradas en accesibilidad e inclusión, donde la distancia física o la limitación de movilidad son barreras relevantes. En todos estos escenarios, la capacidad de interpretar el entorno remoto de forma estructurada y comprensible es fundamental para la eficacia de la experiencia.

En este contexto, el análisis de escenas se perfila como un componente estratégico para enriquecer la interacción remota con el entorno. Al permitir generar descripciones automáticas, responder preguntas sobre lo que ocurre en la escena o destacar elementos relevantes del espacio, este tipo de análisis aumenta la comprensión y reduce la carga cognitiva del usuario remoto, facilitando una interacción más eficiente y contextualizada. Cuando este análisis se integra dentro de una arquitectura cognitiva modular, complementado con el razonamiento flexible aportado por un LLM, el sistema de telepresencia puede ofrecer capacidades avanzadas de interacción adaptativa y de interpretación contextual, que trascienden las limitaciones de la percepción visual básica.

Este apartado examina los fundamentos técnicos que hacen posible la telepresencia inmersiva, sus ámbitos de aplicación más relevantes, y el modo en que el análisis de escenas, implementado en el presente trabajo, puede contribuir a su desarrollo en entornos reales.

2.6.1. Tecnologías de telepresencia y realidad inmersiva

La implementación de sistemas de telepresencia inmersiva exige la integración coordinada de múltiples tecnologías que permitan capturar, transmitir y reconstruir entornos físicos de forma realista y en tiempo casi real. A diferencia de soluciones convencionales de videoconferencia o vigilancia remota, estos sistemas persiguen una experiencia perceptiva más rica, interactiva y envolvente, en la que el usuario pueda percibir el entorno remoto con profundidad, contexto espacial y libertad de exploración.

Entre los componentes fundamentales de estos sistemas destacan [17]:

- Captura de imagen omnidireccional y de profundidad, mediante cámaras 360°, cámaras estéreo o sensores RGB-D que permiten registrar el entorno con una representación volumétrica o panorámica.
- Procesamiento de escena en tiempo real, incluyendo renderizado, corrección de distorsiones y alineamiento espacial, necesarios para garantizar una experiencia fluida y coherente.
- Sistemas de visualización inmersiva, como visores de realidad virtual, pantallas envolventes, o entornos de realidad aumentada, que permiten al usuario integrarse activamente en el entorno remoto.
- Interfaces de interacción, incluyendo seguimiento de cabeza, control gestual o comandos de voz, que facilitan una exploración natural y sincrónica del espacio observado.
- Canales de comunicación de baja latencia, que permiten el envío continuo de datos visuales, auditivos y posicionales, manteniendo la percepción de inmediatez y presencia.

La calidad de la experiencia inmersiva depende de varios factores técnicos, como el campo de visión, la resolución espacial y temporal, el retardo de transmisión, la fidelidad del audio ambiente y la precisión de la reconstrucción tridimensional. La sincronización entre entrada visual y movimiento del usuario es especialmente crítica para evitar efectos de disociación perceptiva, como el *motion sickness* o la fatiga cognitiva [72].

En la actualidad, estas tecnologías están siendo aplicadas con éxito en múltiples dominios. En entornos industriales y logísticos, permiten inspecciones remotas detalladas sin necesidad de desplazamiento físico. En el ámbito sanitario y educativo, habilitan simulaciones, tutorías y procedimientos supervisados a distancia. Y en el contexto de la inclusión social, facilitan experiencias de participación en entornos inaccesibles, como museos, formaciones técnicas o espacios laborales adaptados.

A medida que estos sistemas ganan capacidad de procesamiento e integración sensorial, se consolida la necesidad de incorporar capacidades de análisis inteligente del entorno, como el análisis de escenas y su integración con otras capas de razonamiento. Esta arquitectura permite filtrar, organizar e interpretar automáticamente el contenido capturado, adaptando su presentación al usuario y mejorando la utilidad de la experiencia. Por ejemplo, un sistema puede generar descripciones automáticas de lo que se ve, señalar zonas de interés, responder a preguntas sobre el entorno o adaptar la vista a una tarea específica [20].

En el presente trabajo, estas tecnologías constituyen la base sobre la que se articula el sistema de telepresencia remota, que combina la movilidad del robot, la visión omnidireccional del Búho y la percepción visual distribuida para enriquecer la experiencia del usuario remoto, especialmente en casos de formación inmersiva o asistencia distribuida.

2.6.2. Aplicaciones actuales

Las tecnologías de telepresencia inmersiva están encontrando una creciente adopción en múltiples sectores donde la presencia física no siempre es posible, segura o eficiente. La combinación de visión inmersiva, transmisión en tiempo real y capacidad de interacción remota ha permitido extender el alcance de la actividad humana a espacios distantes o restringidos, mejorando tanto la eficiencia operativa como la accesibilidad.

En el ámbito industrial y logístico, la telepresencia permite la inspección de instalaciones, control de procesos, mantenimiento predictivo y supervisión de operaciones sin necesidad de desplazamiento físico. Robots móviles equipados con visión panorámica o cámaras montadas en cascos pueden recorrer fábricas, almacenes o plantas remotas mientras un operador supervisa la escena en tiempo real desde un entorno de control centralizado. En estos casos, la integración de análisis de escenas permite resaltar zonas de riesgo, identificar elementos clave o generar informes automáticos basados en lo observado [73].

En el campo de la educación y formación técnica, la telepresencia inmersiva ha demostrado ser especialmente útil para facilitar el acceso a entornos especializados, como laboratorios, talleres o simuladores, donde el aprendizaje presencial puede estar restringido por capacidad, coste o distancia. La posibilidad de recorrer virtualmente el espacio, observar procedimientos desde múltiples perspectivas o interactuar con el instructor en tiempo real favorece una experiencia pedagógica más rica. En este contexto, la generación automática de descripciones o respuestas mediante análisis de escenas complementado con LLM, contribuye a reforzar la comprensión y el acompañamiento del estudiante.

En el área de la asistencia remota y teleintervención, la telepresencia se emplea para facilitar el soporte técnico en campo, la supervisión médica, o incluso la mediación cultural o educativa con personas en situación de vulnerabilidad. En estos escenarios, la interpretación automática de escenas permite al sistema filtrar información irrelevante, resaltar acciones críticas o responder consultas mediante lenguaje natural, mejorando así la experiencia de usuarios con limitaciones físicas, sensoriales o cognitivas.

Una línea emergente es el uso de estos sistemas en iniciativas de inclusión social y laboral, donde el acceso remoto a espacios laborales, entornos formativos o eventos públicos puede compensar barreras de movilidad, salud o contexto. Ejemplos como el proyecto INCLUVERSO 5G, una iniciativa conjunta de la Fundación Juan XXIII y Nokia, han explorado este enfoque para facilitar la participación de personas con dificultades de presencialidad en actividades laborales o formativas, mediante sistemas de telepresencia que integran vídeo inmersivo, interacción natural y acompañamiento humano.

En el presente trabajo, estas aplicaciones se toman como referencia para justificar el diseño de un sistema que combina robótica móvil, captura de vídeo inmersivo y análisis de escenas. El objetivo es facilitar el acceso remoto contextualizado a espacios técnicos o educativos, mejorar la experiencia del usuario mediante comprensión automática del entorno, y explorar nuevas formas de interacción sin requerir una interfaz compleja o entrenamiento especializado.

2.6.3. Papel del análisis de escenas en telepresencia

La efectividad de un sistema de telepresencia inmersiva no depende únicamente de la fidelidad visual del entorno transmitido, sino también de su capacidad para facilitar la interpretación del entorno remoto por parte del usuario. En contextos donde la experiencia es asincrónica, parcial o condicionada por limitaciones cognitivas, sensoriales o técnicas, la información visual debe ser complementada con un procesamiento inteligente que permita entender qué ocurre en la escena de forma estructurada y significativa.

El análisis de escenas responde precisamente a esta necesidad, al ofrecer una capa adicional de comprensión automática sobre los contenidos capturados por el sistema. A través de modelos de visión por ordenador y aprendizaje profundo, es posible identificar objetos, describir acciones, inferir relaciones espaciales o responder a preguntas sobre lo que aparece en el cuadro o en el vídeo. Con esta

capacidad el sistema adquiere la capacidad no solo de describir, sino de interpretar, razonar y adaptar dinámicamente su comportamiento en función del contexto y de las necesidades del usuario

En el marco de la telepresencia, esta capa cognitiva puede ser utilizada de múltiples formas:

- Generación de descripciones automáticas del entorno, útiles como apoyo para personas con discapacidad visual o como complemento verbal en entornos educativos o formativos.
- Respuestas a consultas expresadas en lenguaje natural (el ya conocido *Visual Question Answering*), que permiten interactuar con la escena sin necesidad de manipular interfaces gráficas complejas. En este caso, el análisis de escenas provee el contenido base y el LLM aporta la capacidad de generar respuestas adaptadas y comprensibles.
- Detección de eventos o cambios relevantes, como movimientos, presencia de personas, aparición de objetos clave o situaciones anómalas, que pueden servir de alertas contextuales para el operador remoto.

Diversos estudios han destacado el potencial del análisis de escenas como mecanismo de enriquecimiento semántico en entornos inmersivos, permitiendo una interacción más eficaz, accesible y personalizada con el entorno remoto [74]. Esta aproximación es especialmente útil en sistemas que buscan ser inclusivos, adaptativos o aplicables a contextos con alta exigencia informativa, como los que combinan robótica, accesibilidad y asistencia distribuida.

En el presente trabajo, el análisis semántico de escenas se integra como componente clave del sistema de telepresencia inmersiva, permitiendo generar salidas interpretables a partir del vídeo capturado por el robot o el dispositivo egocéntrico, y facilitando así la comprensión remota del entorno.

2.7. Limitaciones actuales y retos tecnológicos

A pesar de los avances recientes, el análisis de escenas enfrenta importantes retos: modelos poco generalizables, procesamiento en condiciones adversas, gestión de grandes volúmenes de datos, integración multisensorial y cuestiones éticas. Esta sección sintetiza los principales desafíos tecnológicos identificados, que constituyen también la base para definir líneas futuras de mejora.

2.7.1. Limitaciones de los modelos actuales

A pesar del avance notable en las arquitecturas de aprendizaje profundo, la integración multimodal y la visión por ordenador aplicada a robótica y telepresencia, los modelos actuales presentan limitaciones que afectan directamente a su aplicabilidad en entornos reales, dinámicos y distribuidos como los que aborda este trabajo. Estas limitaciones se manifiestan tanto a nivel algorítmico como operativo, y condicionan la viabilidad, eficiencia y utilidad de los sistemas que los implementan.

Una de las principales restricciones sigue siendo la dependencia de grandes volúmenes de datos anotados para lograr un entrenamiento supervisado eficaz. Aunque los modelos auto-supervisados y contrastivos han reducido esta necesidad, la calidad de las representaciones sigue estando fuertemente influida por la diversidad y coherencia del corpus de entrenamiento, lo que limita la capacidad de generalización a dominios nuevos o visualmente complejos.

En segundo lugar, muchos modelos siguen teniendo una comprensión limitada del contexto visual, especialmente en tareas que requieren razonamiento más allá de la detección de objetos o la clasificación de escenas. Aunque los modelos multimodales han mejorado la interacción con lenguaje natural y la

interpretación de relaciones complejas, su precisión en tareas abiertas, como VQA o generación de descripciones funcionales, continúa siendo irregular, especialmente cuando se enfrentan a escenas ambiguas, entradas incompletas o formatos no estándar (como vídeo 360° o vista egocéntrica).

Además, en el contexto de sistemas cognitivos distribuidos, la colaboración entre los módulos de percepción (análisis de escenas) y los LLM en la capa de razonamiento aún presenta desafíos importantes. Los flujos de información entre estos componentes no siempre están bien estandarizados, y las representaciones intermedias generadas por el análisis de escenas pueden no ser óptimas para el procesamiento lingüístico posterior, generando pérdidas de significado o ambigüedades que impactan en la calidad del razonamiento.

Asimismo, se observa una escasa capacidad para incorporar conocimiento externo o adaptarse dinámicamente a nuevas instrucciones sin reentrenamiento. Esto afecta directamente la flexibilidad del sistema, especialmente en contextos colaborativos o de telepresencia, donde el entorno y los objetivos pueden variar constantemente. La mayoría de los modelos actuales no ofrecen mecanismos eficaces para ajustar su comportamiento en función del usuario, el contexto situacional o la tarea en curso.

A nivel operativo, otra limitación relevante es el coste computacional de los modelos más avanzados, tanto en entrenamiento como en inferencia. El uso de modelos multimodales de gran tamaño o LLM como evaluadores o generadores implica cargas que no siempre son compatibles con sistemas móviles o arquitecturas distribuidas con limitaciones de ancho de banda y latencia. Esto obliga a explorar enfoques híbridos que deleguen ciertas tareas al servidor, sin comprometer la experiencia del usuario final.

También persisten desafíos en la interpretabilidad y trazabilidad de las decisiones. Muchos modelos operan como cajas negras, dificultando la detección de errores, la depuración o la mejora progresiva del sistema. Esta opacidad es especialmente crítica en contextos donde la robustez, la responsabilidad y la confianza en el sistema son requisitos clave, como ocurre en aplicaciones asistenciales, educativas o colaborativas.

Por otro lado, la capacidad de los sistemas para gestionar la interacción en tiempo real entre percepción, razonamiento y acción sigue siendo limitada. La integración fluida de bucles de percepción-razonamiento-acción, fundamental en arquitecturas cognitivas como la implementada en este trabajo, requiere que los distintos módulos operen con baja latencia, coherencia semántica y adaptabilidad dinámica, aspectos que aún no están plenamente resueltos en los sistemas actuales.

Por último, a pesar de la riqueza de información que ofrecen los sistemas visuales, la fusión semántica entre múltiples entradas visuales y lingüísticas sigue siendo frágil. La integración de vídeo en tiempo real, descripciones generadas, respuestas contextuales y navegación simultánea aún presenta problemas de sincronización, redundancia o incoherencia narrativa.

En resumen, los modelos actuales han permitido avances notables en análisis de escenas y percepción multimodal, pero presentan limitaciones estructurales y operativas que restringen su aplicación directa en escenarios distribuidos, inmersivos y centrados en el usuario. Estas limitaciones justifican el enfoque adoptado en este trabajo, basado en una arquitectura modular y distribuida que combina procesamiento local y remoto, modelos adaptativos y mecanismos de evaluación semántica enriquecida para mejorar tanto la robustez como la utilidad práctica del sistema.

2.7.2. Retos de procesamiento y evaluación

Junto con las limitaciones inherentes a los modelos actuales, la aplicación del análisis de escenas en sistemas reales plantea retos técnicos concretos en dos áreas clave: el procesamiento distribuido en tiempo real y la evaluación fiable de salidas complejas y abiertas. Superar estos desafíos resulta esencial

para que los sistemas de análisis semántico puedan integrarse con eficacia en entornos como la robótica móvil, la telepresencia inmersiva o la asistencia remota.

Desde el punto de vista del procesamiento, uno de los principales retos es la latencia global del sistema. En arquitecturas distribuidas como la planteada en este trabajo, los datos visuales deben capturarse, codificarse, transmitirse, procesarse en un servidor remoto y, finalmente, devolverse al dispositivo cliente en forma de texto, anotaciones o respuestas. Este flujo impone exigencias elevadas de sincronización y eficiencia, especialmente en tareas interactivas donde el usuario espera una respuesta inmediata o contextualizada.

Este reto se acentúa en sistemas cognitivos distribuidos, donde el ciclo *Sense-Think-Act* requiere un flujo continuo y coherente entre los módulos de percepción visual (análisis de escenas), los módulos de razonamiento (LLM) y los componentes de acción o comunicación.

La gestión eficiente de recursos computacionales es otro reto relevante. Aunque existen modelos optimizados, muchas arquitecturas de análisis de escenas, especialmente las multimodales o basadas en LLM, requieren potencia de cómputo que solo está disponible en servidores centralizados o en la nube. Diseñar sistemas que repartan inteligentemente la carga entre cliente y servidor, sin comprometer la calidad del análisis ni la experiencia del usuario, es una necesidad clave en escenarios móviles o de bajo consumo.

Además, la integración de múltiples flujos de entrada visual, por ejemplo, vídeo RGB, vista egocéntrica, panorámica 360° o datos multimodales, plantea problemas de alineamiento temporal, coherencia semántica y redundancia de información. El procesamiento simultáneo de estas fuentes requiere estructuras de control que aseguren una presentación consistente y relevante de los datos interpretados.

En cuanto a la evaluación, el reto principal es la ausencia de métricas objetivas universalmente válidas para tareas como la generación de descripciones, VQA o razonamiento visual. Las métricas clásicas basadas en n-gramas o coincidencia exacta son insuficientes para valorar la calidad semántica, contextual o lingüística de las salidas, especialmente en tareas abiertas donde puede haber múltiples respuestas válidas o interpretaciones plausibles. Esto complica la comparación entre modelos, el ajuste de hiperparámetros y la validación sistemática del sistema.

La introducción de métodos como *LLM-as-a-judge* representa un avance significativo, pero también plantea nuevos retos: cómo garantizar la consistencia del evaluador, cómo controlar sesgos lingüísticos, y cómo integrar estas evaluaciones en ciclos automáticos de mejora o selección de modelos. Además, su coste computacional y su falta de explicabilidad limitan su aplicación directa en tiempo real o en sistemas embebidos.

En conjunto, estos retos de procesamiento y evaluación exigen soluciones técnicas que combinen modularidad, eficiencia y adaptabilidad, y que permitan desplegar modelos de análisis de escenas no solo con alto rendimiento teórico, sino con utilidad práctica demostrable en escenarios reales. El presente trabajo aborda estas cuestiones mediante una arquitectura distribuida optimizada, un flujo de procesamiento secuencial y semántico, y una evaluación híbrida que combina métricas objetivas, evaluación cualitativa y validación contextual mediante LLM.

2.7.3. Integración multisensorial y adaptabilidad

En aplicaciones reales como la robótica autónoma o la telepresencia inmersiva, la percepción no se limita a una única fuente de entrada, sino que se basa en la combinación de múltiples modalidades sensoriales, como imagen RGB, vídeo en tiempo real, visión panorámica, profundidad, audio ambiental,

datos posicionales o incluso señales de interacción gestual o verbal. La integración eficaz de esta información heterogénea plantea desafíos significativos tanto a nivel técnico como de interpretación de los resultados.

Por un lado, los sistemas deben ser capaces de sincronizar flujos de datos capturados a diferentes frecuencias, resoluciones y marcos temporales, manteniendo la coherencia espacial y temporal entre ellos. Esta fusión exige estructuras de control que aseguren una interpretación consistente del entorno, evitando solapamientos, retrasos, contradicciones o pérdida de información relevante. En entornos dinámicos, como los que involucran visión egocéntrica o vídeo 360°, esta dificultad se multiplica por la variabilidad de la escena y la velocidad de cambio.

Este problema se agrava cuando los sistemas deben operar en entornos distribuidos, donde distintas partes del sistema (robot, cámara remota, servidor de procesamiento) disponen de diferentes sensores o tipos de entrada. En estos casos, la adaptabilidad no es solo deseable, sino necesaria para que el sistema mantenga su funcionalidad bajo condiciones de conectividad variable, cambios en la disponibilidad de sensores o interrupciones parciales del flujo de datos.

A nivel funcional, la falta de adaptabilidad impide que los sistemas prioricen elementos relevantes según el contexto o el perfil del usuario. Por ejemplo, un sistema debería poder generar descripciones más detalladas para un usuario con discapacidad visual, o resaltar zonas de riesgo en tareas industriales, ajustando dinámicamente la semántica y el nivel de detalle de la salida. La ausencia de este tipo de flexibilidad reduce la utilidad real del sistema y limita su escalabilidad a nuevos campos.

Asimismo, los sistemas cognitivos distribuidos deben ser capaces de adaptar dinámicamente el flujo de información entre módulos, priorizando ciertos tipos de datos o de análisis en función del contexto operativo y de las capacidades disponibles en cada nodo del sistema. Esto exige no solo flexibilidad a nivel de percepción, sino también mecanismos de control cognitivo que permitan gestionar de forma inteligente la carga informativa y las prioridades de procesamiento.

En el presente trabajo, estos desafíos se abordan mediante una arquitectura modular y distribuida, capaz de recibir entradas visuales de distintos formatos (cuadros, secuencias de vídeo, vista equirectangular o rectilínea) y de adaptar el tipo y complejidad de la salida en función del uso previsto. Esta orientación hacia la adaptabilidad multisensorial no solo mejora la robustez del sistema, sino que amplía su aplicabilidad en escenarios reales y heterogéneos.

2.7.4. Consideraciones éticas y privacidad

El desarrollo y despliegue de sistemas de análisis de escenas en entornos reales plantea una serie de cuestiones éticas y de privacidad que no pueden ignorarse, especialmente cuando estos sistemas operan en espacios compartidos, graban a personas o interactúan con usuarios en tiempo real. La creciente capacidad de los modelos actuales para capturar, interpretar y generar información a partir de cuadros o fragmentos de vídeo hace necesario establecer criterios claros de uso responsable, protección de datos y trazabilidad de decisiones.

Una de las principales preocupaciones gira en torno a la captación y procesamiento de datos personales [75]. En entornos donde los sistemas capturan imágenes de personas, como sucede en aplicaciones de telepresencia, asistencia remota o entornos educativos, existe el riesgo de registrar, analizar o almacenar información sin el consentimiento explícito del individuo. Aunque muchos modelos no identifican directamente a las personas, la posibilidad de reconstruir patrones de comportamiento, interacciones o situaciones sensibles genera un nivel de riesgo que debe ser gestionado mediante políticas de control de acceso y minimización de datos.

Otro aspecto crítico es la veracidad de las salidas generadas. Cuando un sistema emite descripciones automáticas, respuestas a preguntas o resalta ciertos elementos de una escena, es importante poder justificar por qué lo ha hecho y si esa interpretación es fiel al contexto. La presencia de LLM en la capa de razonamiento añade complejidad a este desafío, ya que estos modelos pueden generar respuestas lingüísticamente válidas, pero no necesariamente verídicas o justificadas a partir del contenido visual original. Esta trazabilidad no solo mejora la fiabilidad del sistema, sino que permite detectar sesgos, errores o conductas no deseadas, especialmente cuando el sistema actúa en nombre de un operador remoto o en interacción con usuarios finales.

Además, la posibilidad de grabar y analizar entornos privados o sensibles, como domicilios, aulas o espacios médicos, obliga a establecer límites normativos y contractuales claros sobre el uso, retención y distribución de los datos capturados. El cumplimiento de marcos legales como el Reglamento General de Protección de Datos (RGPD) en la Unión Europea es una condición indispensable para garantizar el respeto a los derechos fundamentales de las personas implicadas.

En este trabajo, estas consideraciones han sido tenidas en cuenta desde el diseño del sistema: no se almacenan imágenes capturadas más allá del tiempo necesario para su análisis; se evita la identificación personal; y se mantiene un enfoque funcional, no intrusivo, basado en interpretar la escena con fines informativos o asistenciales. El uso de modelos explicables y la posibilidad de auditar las decisiones del sistema son elementos clave para asegurar una implementación ética y transparente.

2.8. Líneas de trabajo derivadas del estado del arte

El análisis realizado a lo largo de este capítulo ha puesto de manifiesto el enorme potencial del análisis de escenas y su integración con modelos lingüísticos de gran escala como elemento clave para dotar a los sistemas cognitivos de capacidades avanzadas de comprensión e interacción con entornos complejos. La combinación de percepción visual estructurada con razonamiento flexible basado en lenguaje natural abre nuevas posibilidades en campos como la robótica, la telepresencia inmersiva o la asistencia remota. Sin embargo, también se ha constatado que este enfoque, aunque en auge, presenta todavía importantes retos en términos de generalización, eficiencia, adaptabilidad y evaluación semántica, lo que justifica la pertinencia de su estudio y desarrollo en el presente trabajo.

A partir de los resultados de la investigación y del análisis de las principales tendencias tecnológicas, se ha seleccionado un conjunto de algoritmos de referencia con los que se desarrollará y evaluará el sistema propuesto. Para tareas de VQA se trabajará con los modelos **CLIP** y **ViLT**, que permiten explorar enfoques contrastivos y *transformers* integrados, respectivamente. Para la generación de descripciones en cuadros se utilizarán **Florence** y los modelos **LAVIS**, que destacan por su versatilidad y rendimiento en descripción automática de cuadros. Para la descripción de vídeos, se han seleccionado **LLaVA-Next-Video** y los modelos de **OpenBMB**, que representan el estado del arte en integración de entrada visual secuencial y razonamiento lingüístico.

La evaluación de estos algoritmos se llevará a cabo mediante un enfoque híbrido. Se utilizarán tanto métricas clásicas, BLEU, ROUGE-L, METEOR o CIDEr, que permitirán mantener la comparación con trabajos previos, como métodos basados en *LLM-as-a-judge*, que ofrecen una valoración semántica mucho más ajustada a las tareas abiertas y contextuales que se abordan en este proyecto. Se dará prioridad a este último enfoque de evaluación, por su capacidad para reflejar con mayor fidelidad la calidad interpretativa y comunicativa de las salidas generadas, aspecto esencial en escenarios de interacción humano-máquina.

Capítulo 3

RECURSOS Y MEDIOS EMPLEADOS

Este capítulo recoge los principales recursos y medios utilizados durante el desarrollo del trabajo. Se estructura en dos grandes bloques: por un lado, las herramientas *software* empleadas para el diseño, implementación y evaluación de los modelos; por otro, los elementos *hardware* y los sistemas de integración que han permitido el procesamiento distribuido y la interacción con dispositivos físicos. La selección de estos recursos ha estado guiada tanto por la revisión del estado del arte como por las necesidades concretas del proyecto.

3.1. Herramientas *software* utilizadas

El desarrollo de este TFM ha requerido una combinación de herramientas *software* que abarcan desde entornos de desarrollo y gestión de dependencias hasta *frameworks* de inteligencia artificial y modelos avanzados de visión y lenguaje. La elección de cada componente responde tanto a diversos criterios que se irán detallando a lo largo de esta sección, tomando como referencia la investigación realizada y cuyos principales puntos se mostraron en el Capítulo 2 sobre estudio del estado del arte. En los siguientes apartados se detallan los principales recursos empleados, partiendo de lo general para llegar a lo particular. Así, primero se introducirá el entorno de desarrollo habitual, seguido del lenguaje de programación Python y la manera en que se han creado y encapsulado los diferentes proyectos, después se hablará de algunas bibliotecas concretas, para terminar explicando los diferentes modelos de análisis de escenas empleados.

3.1.1. Entorno de desarrollo y gestión de entornos

En primer lugar, cabe destacar que el sistema operativo habitual de trabajo ha sido Ubuntu, una distribución de Linux especialmente extendida en el ámbito de la inteligencia artificial. Ubuntu ofrece una excelente compatibilidad con bibliotecas de aceleración por GPU como CUDA, así como con *frameworks* como PyTorch y TensorFlow (que se tratarán en el siguiente apartado 3.1.2). Esta distribución es ampliamente utilizada tanto en entornos académicos como en servidores de producción debido a su robustez, su gestión eficiente de recursos y el soporte comunitario y empresarial que ofrece [76]. Aunque el sistema operativo no ha sido un factor crítico en este proyecto, sí ha contribuido indirectamente a la estabilidad y rendimiento general del entorno.

El lenguaje de programación principal empleado ha sido Python, considerado actualmente el estándar de facto en los campos de la inteligencia artificial, la ciencia de datos y el aprendizaje automático. Python destaca por su sintaxis clara y sencilla, una comunidad muy activa y una enorme variedad de bibliotecas especializadas, que permiten abordar desde tareas matemáticas básicas hasta el

desarrollo de modelos avanzados de *deep learning*. Su capacidad de integración con otros lenguajes, así como su portabilidad y soporte en entornos tanto de investigación como de producción, lo convierten en la opción ideal para sistemas complejos como el desarrollado en este trabajo, donde confluyen visión por computador, modelos de lenguaje y análisis distribuido [77].

No obstante, trabajar con Python implica gestionar cuidadosamente las dependencias, ya que proyectos distintos pueden requerir versiones diferentes de las mismas bibliotecas, lo que puede generar conflictos. Para evitar este problema, es imprescindible utilizar sistemas de aislamiento de entornos que permitan mantener organizadas y separadas las configuraciones de cada proyecto.

Entornos virtuales: venv

Uno de los métodos más sencillos para gestionar entornos en Python son los entornos virtuales «venv», una herramienta integrada desde Python 3.3. Esta utilidad permite crear un proyecto dedicado que contiene su propio intérprete de Python y una instalación aislada de paquetes, lo que evita interferencias con el sistema global y otras versiones de los mismos paquetes que puedan ser necesarias para otros proyectos. En la Figura 3.1 se muestra de manera esquemática cómo cada proyecto en un mismo equipo tiene un entorno propio, de manera que queden satisfechos los diferentes requerimientos de cada uno de ellos.

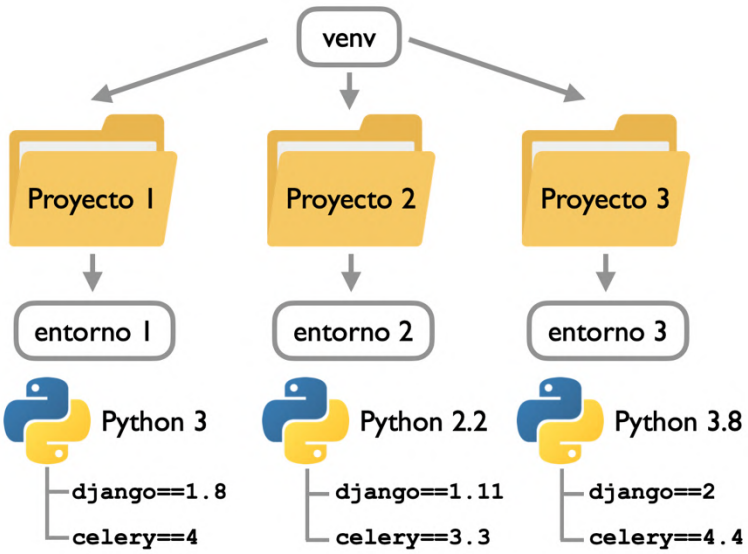


Figura 3.1. Esquema del funcionamiento de venv

Gracias a su bajo consumo de recursos y a su facilidad de uso, venv es ideal para tareas de prototipado, pruebas rápidas y desarrollos ligeros. En este proyecto, se ha utilizado principalmente durante las primeras fases de desarrollo, en combinación con el entorno de desarrollo integrado (IDE, por el inglés *Integrated Development* Environment) de uso habitual, PyCharm, que ofrece soporte directo para entornos virtuales venv [78].

Conda

Para configuraciones más avanzadas, especialmente aquellas que requieren bibliotecas de gran tamaño, dependencias externas o aceleración por GPU, se ha empleado Conda. Conda permite gestionar tanto paquetes como entornos completos, incluyendo versiones específicas de Python y drivers de NVIDIA, lo que lo hace especialmente adecuado para proyectos de *deep learning*. Además, facilita la instalación de bibliotecas de otros lenguajes como C o C++ que pueden resultar difíciles de compilar manualmente [79].

Docker

Finalmente, cuando ha sido necesario garantizar la reproducibilidad total del entorno, así como facilitar el despliegue en distintos sistemas (locales o remotos), se ha hecho uso de Docker. Docker es una

plataforma de contenedores que permite empaquetar una aplicación junto con todo su entorno: código fuente, dependencias, bibliotecas del sistema y configuraciones. A diferencia de las máquinas virtuales tradicionales, los contenedores son más ligeros, más rápidos de arrancar y comparten el núcleo del sistema operativo, lo que los hace más eficientes.

Gracias a Docker, ha sido posible trasladar el proyecto garantizar la compatibilidad de algunos modelos en distintos equipos o servidores, evitando así los habituales problemas de entorno. Además, se ha utilizado como soporte para ejecutar modelos en GPU [80].

3.1.2. *Frameworks* y bibliotecas de inteligencia artificial

Los algoritmos empleados en este trabajo se sustentan en una serie de *frameworks* y bibliotecas especializadas en inteligencia artificial. A continuación, se describen los principales recursos utilizados en esta categoría.

PyTorch

El *framework* principal sobre el que se ha desarrollado toda la arquitectura de *deep learning* es PyTorch, una biblioteca de código abierto desarrollada originalmente por Facebook AI Research. PyTorch permite construir y entrenar redes neuronales con una sintaxis intuitiva, lo que facilita enormemente la depuración y experimentación en entornos donde los modelos evolucionan con rapidez [81].

Uno de los motivos fundamentales para elegir PyTorch frente a alternativas como TensorFlow ha sido su curva de aprendizaje más directa, especialmente en tareas de prototipado rápido y experimentación exploratoria, así como su mayor claridad en el flujo de ejecución, que resulta más similar al de un programa tradicional en Python. Esta característica ha sido especialmente útil en el presente trabajo, para integrar distintas arquitecturas de análisis de escenas y modelos de lenguaje, realizando ajustes frecuentes en la estructura y el flujo de datos.

Asimismo, PyTorch ofrece una compatibilidad robusta con CUDA y otras bibliotecas de aceleración por GPU, lo que ha permitido ejecutar los modelos en entornos optimizados, reduciendo considerablemente los tiempos de inferencia. A nivel práctico, la comunidad activa que mantiene PyTorch, junto con su estrecha integración con bibliotecas como torchvision, lo convierten en un ecosistema completo y en constante actualización.

Biblioteca Transformers

Junto con PyTorch, se ha hecho uso extensivo de la biblioteca Transformers, desarrollada por *Hugging Face*, que actualmente es una de las principales plataformas de referencia en el ámbito del aprendizaje automático, tras haberse consolidado en los últimos años como un repositorio centralizado y accesible de modelos preentrenados, acompañados de herramientas que simplifican su aplicación en tareas reales de procesamiento de lenguaje natural, visión por ordenador o generación multimodal [82].

La biblioteca Transformers, en particular, proporciona una interfaz unificada para una amplia gama de modelos de última generación, lo que ha facilitado enormemente el proceso de carga, preprocesamiento e inferencia. Su estructura modular y estandarizada ha permitido integrar de forma fluida diferentes modelos en los distintos componentes del sistema desarrollado en este TFM.

Otro aspecto a destacar es que la biblioteca está diseñada para funcionar tanto con PyTorch como con TensorFlow, pero su uso con PyTorch ha resultado especialmente fluido, permitiendo acceder a funcionalidades avanzadas con una carga de configuración mínima. Además, *Hugging Face* facilita el

acceso a modelos entrenados por grandes instituciones académicas y empresas tecnológicas, lo que ha permitido explorar variantes de arquitecturas recientes sin incurrir en costes de entrenamiento.

Otras bibliotecas de soporte

Junto a los *frameworks* anteriores, se han empleado diversas bibliotecas de soporte que resultan esenciales en cualquier entorno de trabajo con modelos de inteligencia artificial. En el ámbito de la visión por ordenador clásica, se ha utilizado OpenCV (*Open Source Computer Vision Library*), una biblioteca ampliamente consolidada para la manipulación y análisis de imágenes. OpenCV ha sido fundamental para tareas de lectura, redimensionamiento, transformación y visualización de imágenes, sirviendo como puente entre la entrada visual cruda y el preprocesamiento necesario para los modelos [83].

Asimismo, se han utilizado de forma intensiva NumPy y Pandas, dos bibliotecas esenciales en el tratamiento de datos científicos. NumPy ha proporcionado estructuras eficientes de matrices multidimensionales y funciones matemáticas vectorizadas para la manipulación de tensores y matrices, mientras que Pandas ha sido clave en la organización, limpieza y análisis de los resultados obtenidos, especialmente en la comparación de salidas generadas por diferentes modelos en tareas evaluativas.

Estas herramientas, aunque no forman parte del núcleo del aprendizaje profundo, son imprescindibles para garantizar una gestión robusta del pipeline de datos y mantener la trazabilidad de los experimentos realizados.

3.1.3. Modelos utilizados para análisis de escenas

El análisis de escenas constituye uno de los pilares técnicos del trabajo realizado, al proporcionar la capacidad de interpretación del entorno a los sistemas desarrollados. La mayoría de los modelos seleccionados se basan en arquitecturas de tipo *transformer*, constituyen un enfoque de inteligencia artificial multimodal, pues están adaptadas para trabajar con cuadros, secuencias de vídeo, texto o ambos a la vez. Estas arquitecturas destacan por su capacidad para capturar relaciones contextuales a distintas escalas y su flexibilidad para integrar entradas heterogéneas. En concreto, CLIP, ViLT, Florence y LAVIS trabajan con cuadros, mientras que LLaVA y OpenBMB lo hacen con fragmentos de vídeo.

A continuación, se detallan los modelos específicos utilizados a lo largo del trabajo:

CLIP

CLIP es un modelo desarrollado por OpenAI que ha supuesto un punto de inflexión en la investigación sobre inteligencia artificial multimodal. Su funcionamiento se basa en un entrenamiento contrastivo a gran escala, mediante el cual aprende a asociar representaciones de texto e imagen que corresponden a una misma descripción, sin necesidad de anotaciones explícitas por categorías [38]. Esto lo convierte en un modelo altamente flexible, capaz de realizar tareas de clasificación, búsqueda o análisis semántico sin necesidad de un reajuste supervisado convencional.

A nivel técnico, CLIP consta de dos codificadores independientes: uno visual y otro textual. El codificador visual puede ser una red convolucional o un *transformer*, mientras que el textual suele estar basado en otros tipos de arquitectura. Ambos generan vectores embebidos que se comparan mediante un producto escalar para determinar su grado de afinidad. El sistema es entrenado sobre cientos de millones de pares imagen-texto extraídos de internet, lo que le confiere una gran capacidad de generalización y robustez frente a variaciones de contexto o dominio. Además, su compatibilidad con

bibliotecas como Transformers facilita facilitado su integración en distintos experimentos, actuando en algunos casos como punto de partida para modelos más recientes y especializados.

En el contexto del presente trabajo, se ha planteado el uso de CLIP como algoritmo de VQA, en el que, tras recibir una entrada de texto, típicamente una pregunta, calcula las probabilidades de cada una de las posibles respuestas. En la Figura 3.2 se puede ver representada una salida típica de este algoritmo.

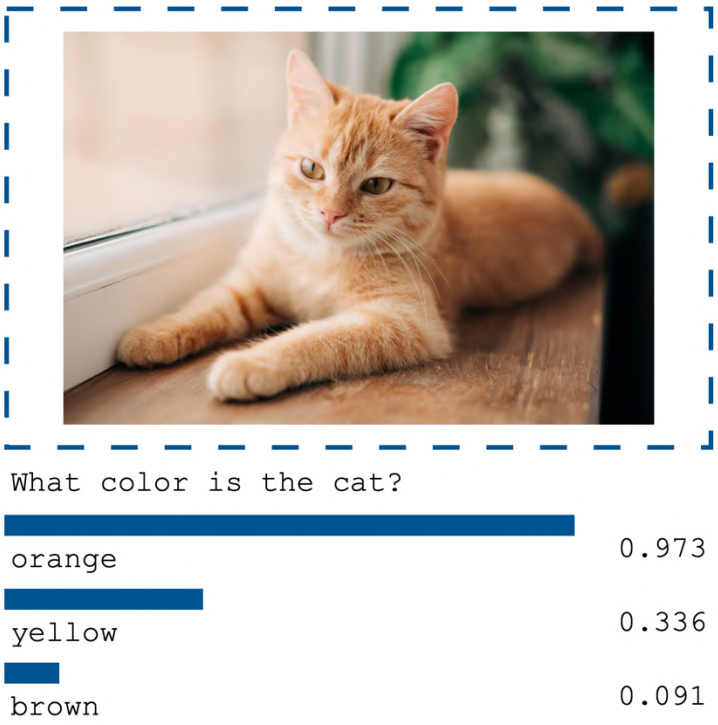


Figura 3.2. Ejemplo de entrada y salida de CLIP

ViLT

ViLT es un modelo propuesto por Kim et al. en 2021 con el objetivo de simplificar la arquitectura de los modelos multimodales tradicionales, eliminando el uso de convoluciones o redes visuales profundas previas al procesamiento conjunto con el texto [56]. A diferencia de enfoques como CLIP, que procesan el cuadro y el texto por separado antes de fusionar las representaciones, ViLT apuesta por una fusión temprana de ambos datos, integrando el procesamiento de imagen y el de texto en un único *transformer*.

Esta decisión arquitectónica tiene dos consecuencias fundamentales: por un lado, reduce el coste computacional y el número de parámetros del modelo, lo que permite una ejecución más ágil y escalable; por otro, fuerza al sistema a aprender representaciones conjuntas desde las primeras capas, lo que puede mejorar su rendimiento en tareas donde las relaciones visual-lingüísticas son sutiles o contextuales.

ViLT ha sido entrenado con una combinación de objetivos multitarea, incluyendo el enmascaramiento de texto, el alineamiento entre imagen y texto (*image-text matching*) y la predicción de correspondencias cruzadas. Esta diversidad de tareas le proporciona una base robusta para enfrentarse a múltiples aplicaciones, desde la clasificación multimodal hasta la respuesta automática a preguntas sobre cuadros.

En este trabajo, se ha empleado especialmente la faceta VQA de ViLT, en la que ante una pregunta, el algoritmo responde de manera concisa con una única palabra, como se muestra en el ejemplo de la Figura 3.3. Su menor demanda de recursos computacionales respecto a otros modelos, junto con su estructura simplificada, lo han convertido en una opción adecuada para pruebas comparativas y prototipado rápido dentro del sistema desarrollado.

Figura 3.3. Ejemplo de entrada y salida de ViLT

Florence

Florence es un modelo de visión desarrollado por Microsoft como parte de su esfuerzo por construir una plataforma generalista de percepción visual, capaz de abordar múltiples tareas (como descripción de cuadros o segmentación, entre otras) dentro del campo de la visión por computador mediante una arquitectura unificada. Esto, a diferencia de los modelos de tareas específicas, lo convierte en una herramienta muy versátil para aplicaciones donde se requiere una interpretación rica y estructurada del entorno visual [51].

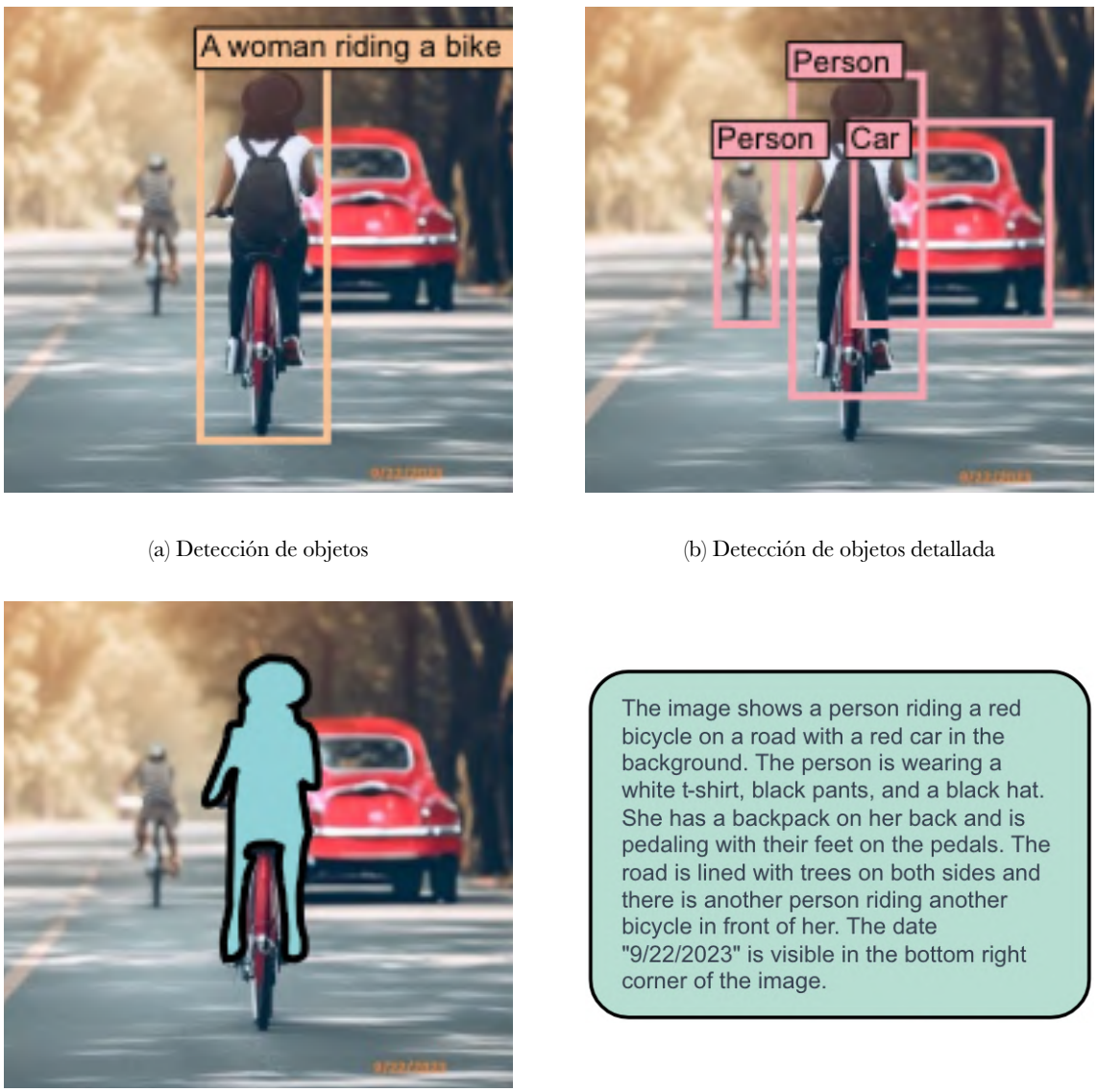
Florence se basa en una arquitectura *transformer* visual que emplea como entrada una representación dividida en sectores, similar a otros modelos como ViT, pero combinada con un entrenamiento previo a gran escala sobre una colección masiva de imágenes y anotaciones lingüísticas. Su entrenamiento ha incluido más de 900 millones de pares cuadro-texto extraídos de bases de datos diversas, lo que le permite manejar situaciones visuales complejas y escenarios poco estructurados, como los que habitualmente se dan en aplicaciones robóticas o de telepresencia inmersiva.

Las principales funcionalidades de Florence incluyen:

- **Clasificación de imágenes:** asignación de etiquetas a cuadros completos en función del contenido general, con una precisión competitiva frente a modelos entrenados específicamente para esta tarea.
- **Detección de objetos:** identificación y localización de múltiples entidades dentro del cuadro, con salida en forma de cuadros delimitadores y etiquetas.
- **Segmentación semántica y segmentación de instancias:** diferenciación precisa entre regiones de cada cuadro asociadas a distintas clases, tanto a nivel general como por objeto individual.

- **Detección de anomalías y atributos visuales:** identificación de patrones o elementos inusuales, así como la descripción de propiedades específicas de los objetos, como color, forma o estado.
- **Generación de descripciones automáticas (*captioning*):** creación de textos en lenguaje natural que resumen o explican el contenido del cuadro, basándose en relaciones semánticas y espaciales entre los elementos visuales.
- **Comprensión multimodal:** capacidad para integrar texto e imagen en tareas conjuntas, como responder preguntas sobre un cuadro, comparar contextos visuales o detectar inconsistencias entre texto e imagen.

Algunas de estas funcionalidades son las que se muestran, a modo de ejemplo, en la Figura 3.4.



(a) Detección de objetos

(b) Detección de objetos detallada

(c) Segmentación

(d) Descripción del cuadro con nivel de detalle medio

Figura 3.4. Algunos ejemplos de las funcionalidades de Florence [84]

En el presente trabajo, Florence ha sido utilizado como uno de los modelos de referencia para tareas de análisis detallado de escenas. Su enfoque generalista ha permitido validar resultados en contextos donde otros modelos más simples (como los de detección de objetos o VQA) no ofrecían un nivel de detalle suficiente, como la identificación simultánea de múltiples objetos con descripciones asociadas o la interpretación de escenas panorámicas complejas.

En el proyecto, su uso ha estado centrado en tareas de generación de descripciones de cuadros, funcionando como un punto de contraste frente a arquitecturas más especializadas o ligeras.

LAVIS

LAVIS es una plataforma desarrollada por Salesforce Research que ofrece una infraestructura modular y extensible para el desarrollo, evaluación e integración de modelos multimodales, especialmente centrados en tareas de visión y lenguaje [57]. Aunque el término LAVIS hace referencia al conjunto de herramientas y no a un único modelo específico, su relevancia radica en que actúa como un marco unificador que permite trabajar de forma estructurada con múltiples modelos preentrenados, facilitando tareas como la inferencia, la comparación de arquitecturas y la generación de salidas en formato estandarizado.

LAVIS se caracteriza por ofrecer una implementación accesible y cuidadosamente optimizada de modelos punteros de visión-lenguaje, como BLIP, ALBEF o CLIP, además de facilitar la experimentación con variantes personalizadas mediante un sistema de configuración flexible. A diferencia de otras bibliotecas más genéricas, LAVIS está específicamente orientada a tareas multimodales y proporciona un ecosistema completo que incluye:

- Modelos preentrenados accesibles desde una única API, lo que permite su uso inmediato sin necesidad de configurar manualmente arquitecturas complejas.
- Comandos integrados para tareas como generación de descripciones o VQA.
- Soporte avanzado para entrenamiento y ajuste fino, con acceso a bases de datos y configuraciones optimizadas para diversas tareas.

En el contexto de este trabajo, LAVIS ha resultado especialmente útil como modelo de análisis de escenas, al emplear fundamentalmente su funcionalidad de generar descripciones de cuadros, de lo que se muestra un ejemplo en la Figura 3.5.

OpenBMB

OpenBMB es una iniciativa de código abierto impulsada por el equipo de investigación y desarrollo de la Universidad Tsinghua (Pekín), orientada al diseño y despliegue de modelos de gran escala en el ámbito del procesamiento multimodal, incluyendo lenguaje, visión y razonamiento. Aunque no se trata de un modelo específico, sino de un conjunto de herramientas y modelos preentrenados, OpenBMB destaca por ofrecer una infraestructura optimizada para la ejecución eficiente de arquitecturas complejas, así como por su orientación a tareas cognitivas avanzadas que requieren la integración de múltiples flujos de información [59].

El interés de OpenBMB dentro del marco de este trabajo reside en su capacidad para procesar y generar descripciones o responder preguntas sobre secuencias de vídeo, en lugar de hacerlo sobre cuadros sueltos, como los modelos vistos hasta ahora, un aspecto particularmente relevante en aplicaciones de análisis de escenas donde la información visual no está anotada explícitamente o donde se requiere interpretar acciones o intenciones implícitas en el entorno.

Figura 3.5. Ejemplo de descripción generada por LAVIS

Entre sus funcionalidades principales, OpenBMB proporciona:

- Modelos entrenados para comprensión multimodal y razonamiento visual, incluyendo variantes especializadas para generar descripciones, VQA o inferencia semántica sobre imagen.
- Compatibilidad con arquitecturas como GPT o CLIP, adaptadas a entornos multimodales.
- Soporte para entrenamiento distribuido y ejecución optimizada en GPU, mediante técnicas avanzadas como segmentación de parámetros, lo que facilita su uso en plataformas con recursos limitados.

Como se adelantó en los párrafos anteriores, en este trabajo, OpenBMB ha sido utilizado principalmente para generar descripciones de fragmentos de vídeo.

LLaVA-Next-Video

LLaVA-Next-Video es una extensión del modelo LLaVA, adaptada específicamente para el análisis de vídeo mediante integración temporal. Esta arquitectura combina un LLM con un codificador visual e incorpora un mecanismo para procesar secuencias de cuadros de vídeo, permitiendo razonar sobre contenidos dinámicos y contextos temporales continuos [58].

El modelo original, LLaVA, fue diseñado como un asistente visual capaz de generar respuestas detalladas a partir de cuadros, apoyándose en una arquitectura basada en Vicuna (una versión optimizada de LLaMA). Su evolución hacia LLaVA-Next y posteriormente LLaVA-Next-Video responde a la necesidad de ampliar esta capacidad hacia dominios donde la información visual no es estática, sino que cambia con el tiempo y presenta relaciones causales, secuenciales o condicionales.

Las principales funcionalidades de LLaVA-Next-Video incluyen:

- Procesamiento de múltiples cuadros o secuencias de vídeo, combinando representaciones visuales codificadas temporales para mantener la coherencia.

- Respuesta a preguntas sobre el contenido visual dinámico, integrando elementos como el orden de eventos, detección de cambios, o interpretación de acciones a lo largo del tiempo.
- Incorporación de conceptos temporales, lo que permite identificar no solo qué aparece en el vídeo, sino cuándo y cómo ocurre.
- Generación de descripciones narrativas, adaptadas al contenido progresivo de la escena, con estructura temporal coherente.

En el contexto de este trabajo, LLaVA-Next-Video ha sido evaluado como una herramienta para tareas de análisis de escenas sobre flujos de vídeo, especialmente relevantes en situaciones de telepresencia inmersiva donde se requiere interpretar entornos dinámicos de forma continua.

3.1.4. Modelos de lenguaje a gran escala

Aunque el funcionamiento de los LLM ya ha sido abordado en capítulos anteriores, en este apartado se describen específicamente los modelos utilizados durante el desarrollo del presente trabajo, así como los motivos de su selección. En lugar de recurrir a API propietarias y de coste elevado como las de OpenAI o Anthropic, se ha optado alternativas que han permitido mantener una flexibilidad operativa completa en entornos de prueba y garantizar la escalabilidad del sistema sin restricciones económicas.

DeepSeek

DeepSeek es un modelo de lenguaje de gran escala desarrollado por el equipo de DeepSeek AI, una compañía emergente del panorama chino que ha alcanzado notoriedad internacional desde su irrupción en la primera mitad de 2024. El modelo fue lanzado con el propósito explícito de ofrecer una alternativa potente y en abierto, frente a los grandes modelos comerciales dominantes del mercado, como GPT-4 de OpenAI o Claude de Anthropic. Su aparición coincidió con el desarrollo del presente trabajo, lo que permitió integrarlo y probarlo en tiempo real, beneficiándose de las actualizaciones sucesivas que fue recibiendo durante los meses del proyecto.

DeepSeek ha generado un gran impacto tanto por su rendimiento como por su enfoque radicalmente abierto, lo que ha suscitado un importante interés entre desarrolladores, investigadores e instituciones académicas. Su diseño modular, basado en una arquitectura *transformer* optimizada, le permite abordar tareas complejas de generación, razonamiento y comprensión del lenguaje con resultados comparables a modelos propietarios mucho más restrictivos en cuanto a acceso y costes [85].

Uno de los elementos más destacados de DeepSeek es su extensión multimodal, denominada DeepSeek-VL (*Vision-Language*), que permite al modelo procesar e interpretar directamente entradas visuales, como imágenes o figuras, siendo esta capacidad determinante para su inclusión en este trabajo.

El uso de DeepSeek en este proyecto se ha articulado en dos roles diferenciados:

- Primero, como herramienta para la evaluación automática de modelos de análisis de escenas, en el marco de la técnica *LLM-as-a-Judge*. Gracias a su capacidad de razonamiento visual, DeepSeek-VL ha sido capaz de interpretar cuadros y emitir juicios cualitativos sobre las descripciones generadas, actuando, así como evaluador externo.
- Segundo, como componente de razonamiento dentro del sistema cognitivo general propuesto en este trabajo. En este rol, DeepSeek ha sido empleado para procesar, contextualizar y generar interpretaciones más amplias a partir de los resultados del análisis de escenas. Este tipo de

razonamiento representa la capa superior de comprensión, alineada con la aspiración de dotar al sistema de una auténtica capacidad interpretativa.

Además de su versatilidad técnica, la decisión de utilizar DeepSeek responde también a criterios de eficiencia y sostenibilidad, al tratarse de un modelo disponible a través de una API sin coste, lo que permite su integración en sistemas experimentales sin incurrir en gastos adicionales ni restricciones de uso.

Qwen

Qwen es una familia de modelos de lenguaje desarrollada por el equipo de Alibaba, orientada a ofrecer soluciones avanzadas en procesamiento del lenguaje natural, razonamiento y comprensión multimodal. Una de sus particularidades más relevantes es que, al igual que DeepSeek, mantiene una política de acceso abierto a sus modelos más recientes, incluyendo variantes especializadas con capacidades visuales y audiovisuales, como Qwen-VL y Qwen-VL-Chat, accesibles a través de una API pública y gratuita [86].

En el contexto de este trabajo, Qwen ha sido empleado específicamente en su versión multimodal con capacidad de lectura de vídeo, lo que lo convierte en una herramienta perfecta para tareas que requieren interpretar información dinámica en formato audiovisual. Hoy en día, es uno de los pocos modelos con API pública que permite el análisis directo de vídeo, lo que justifica su elección frente a modelos propietarios como GPT-4V de OpenAI, cuya utilización está restringida y sujeta a costes significativos.

La funcionalidad principal de Qwen en este proyecto ha sido su uso como evaluador externo (*LLM-as-a-Judge*) en tareas relacionadas con vídeo, proporcionando un mecanismo automatizado para valorar la calidad y precisión de las interpretaciones generadas por modelos como LLaVA-Next-Video u OpenBMB. A través de su capacidad para analizar secuencias de imagen en movimiento, Qwen ha sido capaz de emitir juicios razonados sobre la fidelidad de las descripciones generadas, identificar inconsistencias temporales o semánticas y establecer comparaciones con descripciones de referencia o redactadas manualmente.

El uso de Qwen ha permitido mantener la coherencia metodológica del enfoque *LLM-as-a-Judge* aplicado previamente a cuadros con DeepSeek, extendiéndolo de manera natural al ámbito del vídeo.

3.2. *Hardware* y sistemas de integración

Además del componente *software* descrito en el apartado anterior, el desarrollo del presente trabajo ha requerido el uso de una serie de recursos físicos y sistemas de integración que han hecho posible la ejecución práctica de los modelos, la captura de datos y la interacción entre todos los dispositivos. Estos elementos abarcan tanto el *hardware* utilizado para el procesamiento y la visualización, como los medios de comunicación necesarios para integrar todos los componentes distribuidos en entornos locales o remotos. La selección de estos medios ha estado determinada por las necesidades funcionales del sistema y por la disponibilidad tecnológica del entorno de desarrollo, el Nokia XRLab.

3.2.1. Equipos y recursos físicos empleados

La experimentación se ha apoyado en equipos físicos reales, esenciales para validar las funcionalidades desarrolladas, especialmente en las tareas de captura de datos, análisis de escenas y

telepresencia inmersiva. A continuación, se describen los principales dispositivos utilizados, comenzando por la plataforma robótica cuadrúpeda empleada en las pruebas de integración.

Unitree Go2

El Unitree Go2 [87] es el robot cuadrúpedo que puede verse en la Figura 3.6. Desarrollado por la empresa china Unitree Robotics, especializada en plataformas robóticas bioinspiradas. Su diseño imita a un perro para ofrecer una movilidad versátil, estable y adaptable a terrenos irregulares, lo que lo convierte en una herramienta idónea para tareas de exploración, inspección y asistencia remota en entornos semiestructurados.

Figura 3.6. Robot Unitree Go2 [87]

El modelo Go2 se caracteriza por una notable integración de *hardware* y *software*, orientada tanto al uso industrial como a la investigación avanzada. Entre sus especificaciones más relevantes se encuentran:

- Doce grados de libertad de movimiento (3 por extremidad), que le permiten ejecutar patrones de marcha complejos, mantener el equilibrio dinámico y realizar maniobras como giros en el sitio, escalada moderada o recuperación ante tropiezos.
- Procesador de a bordo con soporte para ejecución de scripts y control en tiempo real, además de una unidad de expansión para añadir módulos externos.
- Sensores integrados, incluyendo cámaras gran angular, IMU (*Inertial Measurement Unit*, en inglés, Unidad de Medición Inercial), sensores de posición y, en las versiones avanzadas, un sensor LiDAR 2D de 360º, que permite una percepción tridimensional precisa del entorno. Este sensor es especialmente útil para tareas de navegación autónoma, detección de obstáculos y mapeado espacial, ampliando significativamente la capacidad del robot para operar en escenarios reales o no estructurados.

- Conectividad inalámbrica y capacidad ROS (*Robot Operating System*), lo que permite integrarlo con pipelines de visión por computador, control remoto o algoritmos de IA en tiempo real.
- Velocidad de marcha de hasta 1,6 m/s y autonomía superior a los 60 minutos en condiciones normales de uso.

En el marco de este trabajo, el Unitree Go2 ha sido utilizado como soporte físico para pruebas de movilidad y análisis visual del entorno, en combinación con los sistemas de percepción desarrollados. Si bien el procesamiento de los modelos de análisis de escenas no se ha realizado directamente en el procesador del robot sí se ha establecido una comunicación directa entre el sistema cognitivo externo (basado en GPU) y el robot.

Además de su uso como plataforma de adquisición de imagen, el Unitree Go2 ha servido como modelo representativo de robótica zoomórfica móvil, un tipo de robótica especialmente adecuada para tareas de inspección, rescate o asistencia en entornos donde la morfología de un robot cuadrúpedo ofrece ventajas frente a diseños con ruedas o cadenas.

El Búho

El sistema inmersivo llamado Búho (que se muestra en la Figura 3.7) representa una solución portátil y de bajo coste para la comunicación remota en tiempo real mediante vídeo 360°, con funcionalidades avanzadas de telepresencia e interacción entre los usuarios. Este sistema ha sido desarrollado en el XRLab de Nokia, y está diseñado específicamente para facilitar la participación inmersiva de usuarios remotos en entornos físicos reales, permitiendo que un usuario con cascos de VR se «teletransporte» virtualmente al lugar donde el dispositivo se encuentra físicamente. A diferencia de otras plataformas basadas en entornos simulados o avatares, el Búho transmite vídeo real en tiempo real, capturado por cámaras 360° y codificado para ser reproducido sin latencia perceptible [88], [89]. Este sistema incorpora audio bidireccional, lo que facilita la interacción fluida entre el emisor y los receptores. Además, al utilizar cascos de realidad virtual, el usuario receptor experimenta un cambio dinámico del punto de vista del vídeo en función del movimiento de su cabeza, generando la sensación de estar físicamente presente en el entorno donde se encuentra la cámara. En el lado del receptor, esta experiencia puede complementarse con representaciones virtuales o avatares que permiten visualizar e identificar a otros participantes dentro del entorno inmersivo.

Figura 3.7. Sistema de telepresencia inmersiva Búho

Desde el punto de vista físico, el sistema está compuesto por cámaras capaces de capturar vídeo en 360° y alta resolución. Estas cámaras están montadas sobre una estructura impresa en 3D, ligera y

transportable, que puede fijarse a un trípode o colocarse sobre superficies planas. La cámara se conecta mediante USB a la tarjeta, que se encarga de la comunicación con el *backend* y de tareas de control, así como del proceso de codificación del vídeo en H.264 o H.265 [89].

La transmisión de vídeo se realiza a través del protocolo RTP sobre UDP, lo que garantiza un flujo constante y con baja latencia, incluso en condiciones de red variables. Además, el sistema cuenta con conectividad multimodal (Ethernet, wifi y redes móviles mediante módems 4G/5G) y es alimentado por una batería portátil, lo que permite su despliegue en exteriores o entornos no equipados.

Para la comunicación bidireccional se utiliza un altavoz y micrófono, gestionado mediante una serie bibliotecas propias, que garantizan una calidad de audio nítida y sincronizada con el vídeo. En el plano *software*, tanto la aplicación inmersiva (para dispositivos como gafas de realidad virtual) como la versión 2D han sido desarrolladas en Unity, con bibliotecas personalizadas compiladas de forma cruzada para distintas plataformas [89].

El sistema admite la interacción entre usuarios locales y remotos no solo mediante presencia visual y auditiva, sino también mediante el uso compartido de contenido (por ejemplo, presentaciones), que puede mostrarse como pantallas flotantes con posición y tamaño configurables.

El Búho se ha concebido como una herramienta accesible y de fácil uso, incluso para personas sin experiencia previa en tecnologías inmersivas. Su arquitectura modular y escalable permite su aplicación en múltiples escenarios, como la formación industrial, visitas guiadas, eventos sociales o reuniones académicas, superando las limitaciones de las videoconferencias tradicionales y ofreciendo una experiencia mucho más rica en cuanto a contexto espacial, comunicación no verbal e inmersión sensorial [90].

3.2.2. Sistemas de integración y comunicación

La arquitectura de integración de este proyecto combina herramientas consolidadas y algunos desarrollos propios para lograr una transferencia robusta, eficiente y sincronizada de datos audiovisuales e información entre los distintos componentes del sistema. A continuación, se detallan los principales medios usados en esta infraestructura: GStreamer, y las algunas bibliotecas internas desarrolladas en el XRLab.

GStreamer

GStreamer [91] es una biblioteca de código abierto multiplataforma diseñada para la construcción de flujos multimedia. Su arquitectura se basa en un modelo de canales de procesamiento (denominadas en inglés *pipeline*) en las que cada elemento actúa como un bloque funcional (fuente, codificador, decodificador…) que gestiona el flujo de datos.

En este proyecto, GStreamer se ha empleado para capturar, codificar y retransmitir flujos de vídeo en tiempo real, principalmente desde el robot Unitree Go2, pero también como mecanismo de prueba en fase de desarrollo del *software*. El vídeo capturado por las cámaras se comprime usando H.264, y se encapsula mediante RTP y UDP, según las necesidades del cliente de recepción. La elección de GStreamer se debe, fundamentalmente, a las siguientes características:

- Su alta modularidad, que permite construir flujos personalizados según el entorno (latencia baja, ancho de banda limitado, resolución variable…).

- Su compatibilidad con protocolos de red, ampliamente aceptados en el ecosistema multimedia, así como con entornos tan diversos como un robot, un sistema de telepresencia o una biblioteca de Python propia, cuya integración hubiera sido mucho más compleja de hacerse de otra manera.
- Su soporte para sincronización de tiempo y control de calidad del flujo.

GStreamer ha sido clave en la etapa de prototipado y en la conexión con el robot zoomórfico, integrándose posteriormente con las bibliotecas propias como fuente de datos.

Algunas bibliotecas propias

En este trabajo se han utilizado dos bibliotecas propias, desarrolladas específicamente para facilitar la integración entre hardware embebido y sistemas de análisis en arquitecturas distribuidas.

- La primera biblioteca está orientada a la transmisión modular de imágenes y metadatos. A la hora de hacer referencia a ella en este memoria se hará referencia a ella como «biblioteca propia de transmisión de datos», o de alguna manera similar. Su uso ha permitido enviar, de forma eficiente y sincronizada, información visual junto con datos contextuales relevantes desde los dispositivos de captura hasta el servidor de procesamiento

  Ha sido empleada para transmitir imágenes captadas por algunas cámaras u obtenidas a través de flujos de GStreamer hacia un servidor de análisis, donde se analizan en tiempo real. En este contexto, funciona como el puente que une el *hardware* embarcado con el sistema cognitivo central

- La segunda biblioteca permite la recepción de datos en tiempo real. Está diseñada para sincronizar múltiples flujos procedentes de sensores, como vídeo o audio, permitiendo una alineación de la información recibida.

  En esta aplicación concreta, ha sido usada como receptor principal de imágenes, operando como complemento de la anterior en los casos en los que el dispositivo emisor (el Búho) ya estaba configurado para transmitir datos mediante este protocolo. A lo largo del trabajo, se la llamará «biblioteca propia de recepción del Búho» o de alguna manera similar.

Ambas bibliotecas cumplen funciones complementarias y han sido fundamentales para garantizar una comunicación fluida y eficaz entre los distintos componentes del sistema.

Capítulo 4

DESARROLLO DEL TRABAJO, RESULTADOS Y VALIDACIÓN

4.1. Desarrollo e implementación del sistema

4.1.1. Exploración inicial de algoritmos de análisis de escenas

Para comprender el funcionamiento de los algoritmos de análisis de escenas, se comenzó con implementaciones locales sencillas en un entorno controlado, lo que permitió validar su comportamiento, conocer sus requisitos técnicos y sentar una base práctica antes de avanzar hacia modelos más complejos y especializados.

La selección de los algoritmos empleados respondió a la investigación ya presentada en el estudio del estado del arte, donde se anticipó que los modelos elegidos por su relevancia científica y tecnológica según las tendencias actuales son ViLT, CLIP, Florence, LLaVA, LAVIS y OpenBMB. Sin embargo, a la hora de iniciar el desarrollo, se optó por comenzar con una herramienta más sencilla y accesible, YOLO, dado su amplio uso, documentación y facilidad de integración. No obstante, los resultados obtenidos confirmaron que simple identificación de clases y el dibujo de cajas sobre el cuadro original, resultaban claramente insuficientes para los niveles de comprensión contextual requeridos en entornos de telepresencia inmersiva y robótica, reforzando la necesidad de utilizar arquitecturas más sofisticadas.

Por ello, se pasó directamente al uso de algoritmos más potentes basados en *transformers*. En todos los casos se desarrollaron scripts funcionales capaces de ejecutar inferencias en tiempo real a partir de la webcam del sistema o de cuadros, con preguntas predefinidas o variables:

- En el caso de ViLT, se implementó un prototipo local que permitía capturar vídeo en directo desde la webcam del sistema, o bien desde un archivo local del sistema, y lanzar preguntas predefinidas en lenguaje natural, obteniendo respuestas breves directamente en la consola. Este sistema sirvió como primer acercamiento real a la tarea de VQA, permitiendo trabajar por primera vez con una arquitectura que fusionara texto e imagen.
- En paralelo, se desarrolló un *script* para CLIP, orientado a medir la similitud entre una descripción textual y un conjunto de imágenes. En este caso, se probó un conjunto reducido de imágenes cargadas desde URL o desde el propio equipo. Para probar su funcionamiento había dos opciones principales: dejar que el propio algoritmo identificara qué había en las imágenes, o dar un listado

de alternativas, de las que calculaba las probabilidades y escogía la más plausible. Esta segunda opción de funcionamiento es similar a la mostrada en la Figura 3.2 del capítulo anterior.

- Posteriormente, se trabajó con Florence. Se desarrolló un sistema que ejecutaba en bucle las inferencias sobre un cuadro en vivo (ya fuera captándolo de la webcam o de un vídeo u otras imágenes almacenadas en el ordenador), permitiendo al usuario seleccionar entre las doce tareas diferentes que ofrece mediante un índice numérico. La capacidad del modelo para generar descripciones precisas y adaptadas a las distintas tareas visuales demostró su potencial para ser integrado en sistemas XR avanzados, convirtiéndose en una pieza central de las pruebas posteriores.

- De una manera similar, se creó otro programa que empleaba el algoritmo LAVIS. La principal diferencia con respecto al anterior es que entre sus funcionalidades no era «elegibles» mediante una entrada o índice, sino que se limitaba a generar un cierto número (dado por el usuario) de descripciones diferentes para cada cuadro.

- Después, se llegó a LLaVA-Next-Video. Al principio se realizó una adaptación idéntica a la de los programas anteriores, es decir, captaba un cuadro de un vídeo, ya fuera el grabado por la webcam o el de un archivo local, y le aplicaba una pregunta. Sin embargo, esto llevaba a una serie de errores que, tras comparar con la arquitectura de los anteriores programas, parecían causados por el propio modelo.

  Tras realizar una pequeña investigación, se vio que el algoritmo debía trabajar sí o sí con fragmentos de vídeo, no pudiendo generar descripciones de un cuadro. Para garantizar el correcto funcionamiento, se diseñó un *script* capaz de capturar secuencias de un número determinado de cuadros que se almacenaban en un *array*, de manera que, al llegar a esa cantidad, las cuadros o cuadros se concatenaban, se procesaba ese nuevo pequeño vídeo, se vaciaba el *array* y se volvía a crear un nuevo fragmento de vídeo.

  De esta manera, además, se exprimía más adecuadamente la principal ventaja de LLaVA con respecto a los algoritmos anteriores: la capacidad de explorar tareas de razonamiento temporal y análisis de relaciones entre acciones continuas y distribuidas a lo largo del tiempo, en lugar de cuadros estáticos.

- Por último, se aprovechó esta misma solución para crear un programa similar con entradas de vídeo, aunque usando el modelo OpenBMB en lugar de LLaVA-Next-Video. En este caso, el funcionamiento de ambos programas era prácticamente idéntico, permitiendo una entrada de texto para acotar el procesamiento, y ofreciendo ambos una descripción de cada fragmento de vídeo como salida.

Desde el punto de vista operativo, el uso de estos modelos requirió configurar entornos específicos para cada caso. Dada la elevada demanda de recursos computacionales, en muchos casos se superaron las capacidades del equipo local, en especial para los modelos de gran tamaño como LLaVA-Next-Video y OpenBMB, por lo que fue necesario ejecutar los programas en equipos con una tarjeta gráfica de mayor memoria. Para garantizar la portabilidad entre equipos, se emplearon tanto entornos Conda como entornos virtuales venv, e incluso se recurrió en algunas ocasiones al uso de contenedores Docker para aislar dependencias y facilitar la reutilización de código en entornos heterogéneos.

Con esta sencilla implementación se comprobó, además, que el número de cuadros por segundo que se estaba empleando era demasiado alto. El vídeo adquirido era de 30 fps (fotogramas por segundo), y pese a que eso hacía que fuera de una gran calidad, se llegó a la conclusión de que era redundante que

un algoritmo evaluase prácticamente el mismo cuadro por treinta veces en tan solo un segundo, pues sobrecargaba mucho los modelos sin aportar ninguna nueva información relevante. Para tratar de solventar este problema, se intentó modificar (sin éxito) la frecuencia de fotogramas de la webcam y de lectura de archivos locales; sin embargo, para la primera opción se requería modificar algunos parámetros internos de la configuración del equipo que se prefirió no alterar. Así las cosas, se optó por incluir un pequeño fragmento de código que «saltaba» un número de fotogramas definido por el usuario de entre el flujo a 30 fps que se recibía. Esta modificación se realizó, además, con un enfoque a medio plazo, pues posteriormente este «salto de fotogramas» podría adaptarse en el cliente del sistema a cualquier equipo de adquisición de vídeo, como el robot o el Búho.

Inicialmente, se modificó la frecuencia de cuadros a 1 fps, pues se pensó que una lectura de la información de la escena cada segundo sería suficiente; no obstante, esto generó un nuevo problema: se perdía la sensación de «continuidad» o «fluidez» en el vídeo, de manera que los algoritmos como LLaVA y OpenBMB, cuya principal ventaja es precisamente la identificación de acciones a lo largo del tiempo perdían su utilidad y empeoraban significativamente sus resultados. Finalmente, tras realizar diferentes pruebas, se llegó al compromiso de los 3 fps, en el que no se sobrecarga en exceso al sistema y los algoritmos que tienen vídeo como entrada no veían mermado su rendimiento.

Esta etapa, aunque centrada en pruebas locales y *scripts* independientes, supuso un paso decisivo en la construcción del sistema completo. Más allá de los resultados funcionales, gracias a ella fue posible evaluar el comportamiento de los modelos en condiciones reales, anticipar los problemas de integración y seleccionar, con criterios fundados, qué componentes debían formar parte de la arquitectura final. Al mismo tiempo, ofreció una comprensión profunda de las fortalezas y debilidades de cada enfoque, lo que resultó clave para diseñar una solución robusta y escalable para análisis de escenas en contextos de telepresencia inmersiva.

4.1.2. Introducción a la arquitectura cliente-servidor planteada

Como se expuso en el capítulo dedicado al estado del arte, una de las estrategias más efectivas para abordar entornos de inferencia visual en tiempo real es el uso de arquitecturas distribuidas basadas en la filosofía cliente-servidor. Este enfoque, ampliamente adoptado en sistemas de análisis visual avanzado, no solo permite desacoplar funcionalmente los distintos bloques que componen el flujo de procesamiento, sino que facilita, además, la modularidad, la escalabilidad y el mantenimiento del sistema.

Por ello, la arquitectura planteada se compone, de tres módulos diferenciados: un cliente de envío, uno o varios servidores y un cliente de recepción de resultados, tal y como se ve en el esquema de la Figura 4.1.

Figura 4.1. Esquema básico de la arquitectura planteada

Esta estructura lineal, aunque conceptualmente sencilla, ofrece un alto grado de flexibilidad técnica, lo que resulta especialmente útil cuando se trabaja con múltiples algoritmos de naturaleza distinta. Gracias a esta separación de responsabilidades, cada uno de los servidores puede estar dedicado en exclusiva a un modelo de inferencia visual concreto, sin interferir en la lógica ni el rendimiento de los

demás. Asimismo, esta segmentación permite que cada servidor pueda ser ejecutado en una máquina distinta, e incluso con capacidades de cómputo especializadas, como GPU, sin necesidad de modificar el resto de componentes del sistema.

Además de su robustez técnica, esta solución resulta da un buen resultado en entornos reales por su capacidad de adaptarse a diferentes fuentes de entrada. El cliente de envío se ha diseñado para ser compatible con distintas fuentes de vídeo, permitiendo la captación desde una webcam local, archivos almacenados en el equipo o dispositivos externos, entre los que se incluyen tanto el robot como el Búho, cuyas características específicas de integración y funcionamiento se explican en los apartados siguientes. En todos los casos, el cliente de envío tiene como responsabilidad leer y procesar los cuadros obtenidos desde la fuente correspondiente, y transmitirlos al servidor de destino utilizando una codificación adecuada y un sistema de envío robusto.

En este punto conviene destacar el papel central de la biblioteca propia encargada de la transmisión de los datos, introducida en el Capítulo 3, y que constituye uno de los pilares técnicos de la arquitectura completa. Esta herramienta, desarrollada específicamente para el presente proyecto, permite la transmisión eficiente de imagen y metadatos entre los distintos nodos del sistema, manteniendo una latencia mínima incluso en condiciones de red exigentes. Además de su ligereza y fiabilidad, la biblioteca ofrece mecanismos nativos de empaquetado, control y extensión, lo que ha permitido adaptarla fácilmente a las necesidades de cada algoritmo y flujo de datos.

El segundo bloque de la arquitectura está constituido por los servidores de inferencia. Cada uno de estos módulos está vinculado a un único algoritmo de tratamiento visual y se encarga de recibir los paquetes de imagen desde el cliente emisor, ejecutar el modelo correspondiente y generar tanto una imagen procesada, por ejemplo, con anotaciones, marcadores o regiones segmentadas, como una serie de metadatos asociados. Estos metadatos incluyen, entre otros, el resultado de la inferencia, el tiempo total de procesamiento, el nombre del modelo utilizado y cualquier otro indicador relevante para la visualización o supervisión del sistema. En los casos en que el modelo opere sobre secuencias de cuadros o fragmentos de vídeo breves, el servidor también se encarga de gestionar la agrupación de cuadros y la sincronización temporal.

Por último, los resultados obtenidos son enviados desde el servidor a un cliente de recepción. Este módulo tiene como función mostrar en pantalla tanto la imagen procesada como los datos complementarios que acompañan al resultado. Además, su diseño modular permite que cada cliente esté vinculado a un único servidor, lo cual evita colisiones o pérdidas de paquetes en situaciones de alta demanda. En caso de ser necesario, la visualización puede adoptar distintos formatos según el entorno de ejecución: desde una simple ventana emergente hasta un panel interactivo con visualización enriquecida.

Cabe señalar que, aunque el flujo general de información sigue un esquema unidireccional (cliente de envío → servidor → cliente de recepción), en determinados casos el servidor puede enviar señales de control hacia atrás, como por ejemplo peticiones de pausa temporal. Este canal de control, aunque accesorio, permite dotar al sistema de mecanismos básicos de retroalimentación sin comprometer su simplicidad estructural.

En conjunto, esta arquitectura ha demostrado ser especialmente adecuada para un sistema como el aquí descrito, que integra múltiples algoritmos de inferencia, requiere una baja latencia en la transmisión y debe mantenerse flexible ante posibles cambios de fuente, algoritmo o configuración. La combinación de una estructura clara, una biblioteca de comunicación propia y una separación estricta de funciones ha permitido construir una plataforma robusta, escalable y fácilmente integrable en escenarios reales.

4.1.3. Clientes de envío de imagen

Una vez completada la fase de pruebas locales de los modelos de análisis de escenas y comprobado su correcto funcionamiento en entornos simulados, el siguiente paso natural consistió en aplicar estos algoritmos sobre datos visuales reales, ya sean captados en tiempo real (desde la propia webcam, desde una plataforma robótica o desde un dispositivo de telepresencia inmersiva) o leídos de algún archivo almacenado localmente.

Así, los clientes emisores de imagen constituyen el punto de partida del flujo de datos dentro de la arquitectura cliente-servidor planteada. Su misión principal es capturar imagen desde una fuente determinada, ya sea una cámara integrada, un archivo local, un robot o un dispositivo externo como el Búho, y enviarlas a uno o varios servidores de inferencia para su posterior análisis. En todos los casos, el vídeo capturado se redimensiona automáticamente a una resolución estándar de 640 × 480 píxeles, lo que garantiza una uniformidad en el tratamiento de las imágenes independientemente de la fuente de origen. Aunque existen variantes específicas según el origen del flujo visual, todos los clientes comparten una base funcional común que ha ido evolucionando conforme lo ha requerido la complejidad del sistema.

Como se ha explicado en el apartado anterior, el envío de datos entre módulos se realiza fundamentalmente a través de las bibliotecas propias, adaptada a los requerimientos de baja latencia y robustez del sistema. Esta herramienta permite el envío de paquetes de imagen y metadatos a través de canales independientes, cada uno identificado por una dirección IP y puerto concretos. Una de las premisas de diseño del sistema es que cada puerto debe estar vinculado de forma exclusiva a un único flujo, evitando así posibles colisiones o errores en la recepción.

La primera versión funcional del cliente emisor era deliberadamente sencilla. Se trataba de un módulo capaz de capturar imagen de una webcam o vídeo local, y enviarlos a un servidor específico a través de una única dirección IP y puerto. No se incluía, en esta etapa, ningún tipo de metadato ni canal secundario. Esta implementación permitía validar el correcto funcionamiento del envío de imagen en tiempo real y sentó las bases para el desarrollo posterior.

Poco después, se amplió el diseño para incorporar dos flujos paralelos, uno dedicado exclusivamente a las imágenes y otro para los metadatos generados o adjuntos. Esta separación, facilitada por la manera en que están construida la biblioteca propia que se emplea en la transmisión de datos, permitía mantener una mayor claridad semántica en los canales de comunicación, simplificar la lógica de recepción en el servidor y añadir datos relevantes (tiempo de inferencia, tipo de fuente, configuración del cliente, etc.) sin interferir con el contenido visual. En este momento se introdujo también la posibilidad de ajustar la frecuencia de cuadros que se envían a los servidores, como se explicó antes.

Al incrementarse el número de modelos en el sistema, surgió la necesidad de que varios servidores pudieran recibir simultáneamente los mismos cuadros, con el fin de evaluar el comportamiento de cada uno con idénticos datos de entrada. Este nuevo requisito planteó desafíos relevantes, ya que la estructura inicial del cliente emisor no estaba concebida para gestionar múltiples destinos de forma sincronizada.

Se estudiaron entonces dos alternativas principales:

- Una posibilidad era dotar al cliente de un comportamiento más próximo al de un servidor, permitiendo que cualquier servidor pudiera «conectarse» a él. En el contexto de la biblioteca empleada, esto se consigue mediante el parámetro «bind=True», que permite al *socket* del cliente actuar como punto de conexión pasiva, en lugar de como emisor activo («bind=False»), de igual manera a la que suele trabajar un servidor.

Aunque esta solución permitía que varios servidores recibieran datos del mismo cliente, presentaba inconvenientes significativos. Al alterar la lógica natural cliente-servidor, se producían fallos esporádicos en la transmisión y situaciones poco controladas en el envío, generando un comportamiento difícil de depurar y poco escrupuloso en contextos exigentes.

- La segunda opción era instanciar varios clientes independientes, cada uno configurado para enviar datos a un servidor concreto, utilizando sus correspondientes pares IP/puerto (imagen + metadatos). Esta estrategia respetaba la lógica tradicional y ofrecía una mayor limpieza técnica.

  Sin embargo, presentaba dos grandes problemas. En primer lugar, era difícil garantizar que todos los clientes enviaran exactamente los mismos cuadros en el mismo instante, lo que podía dar lugar a resultados dispares entre modelos. En segundo lugar, cualquier canal de retroalimentación (como una orden de pausa) quedaba limitado a un solo cliente y no podía sincronizarse entre todos, comprometiendo así la integridad del sistema.

La solución finalmente implementada fue desarrollar un único cliente de «macroenvío», que permitiera gestionar dentro de un mismo proceso el envío a todos los servidores en paralelo. Este cliente recibe cada imagen de entrada y la transmite a un conjunto de pares IP/puerto predefinidos, asignando dos puertos por cada servidor: uno para imagen y otro para metadatos.

Aunque esta solución implicaba enviar múltiples copias simultáneas y podía suponer un leve aumento en la carga de red, demostró ser la más robusta y precisa. Garantizaba que todos los servidores recibieran los mismos datos exactamente en el mismo instante, eliminaba el problema de sincronización y permitía mantener la trazabilidad y consistencia entre los resultados generados por cada modelo.

La última mejora introducida en el cliente fue la incorporación de un canal de control inverso, configurado en un puerto exclusivo para recepción. Este canal permite que el cliente reciba instrucciones procedentes de alguno de los servidores, especialmente útil en situaciones donde el procesamiento de los modelos conlleva tiempos elevados. En esos casos, el servidor puede emitir una orden de pausa, de forma que el cliente detenga temporalmente el envío de cuadros hasta recibir una nueva señal de reactivación. Esta funcionalidad dota al sistema de un mecanismo simple de gestión de carga y control adaptativo, sin necesidad de complejos protocolos de comunicación bidireccional.

En la Figura 4.2 se presenta un esquema elemental de cómo funciona este cliente, que se irá expandiendo de manera sucesiva a lo largo del capítulo, cuando se expliquen funcionalidades adicionales de este mismo cliente, los servidores y otros programas realizados. Como resolución del vídeo de entrada se ha señalado por defecto 1280 × 720 px, que es la de la webcam del equipo, aunque variará en función de la fuente de vídeo.

Emisor desde webcam o archivo local

El cliente más básico y directo de los desarrollados es aquel que obtiene cuadros de una webcam local conectada al sistema. Este módulo fue, de hecho, el primero en implementarse, sirviendo como banco de pruebas para validar el funcionamiento de las bibliotecas propias y el flujo de datos básico entre cliente y servidor.

Una evolución natural del cliente anterior fue la implementación de un módulo capaz de leer desde archivos locales (tanto vídeos como cuadros). Esta funcionalidad resulta especialmente útil para pruebas reproducibles, ya que permite enviar una misma secuencia visual a distintos modelos en diferentes momentos, manteniendo la coherencia de entrada.

Figura 4.2. Esquema con el funcionamiento básico del cliente de envío

El funcionamiento de este tipo de clientes es continuo y cíclico: cada cuadro capturado se empaqueta y se envía por el canal de imagen a la dirección IP y puerto asignados. En paralelo, si el sistema lo requiere, se envían también metadatos complementarios (como la resolución del cuadro, la hora de captura, información del dispositivo de entrada, etc.) a través del canal de metadatos. Esta estructura doble es gestionada de forma transparente por la biblioteca propia empleada, que permite mantener separados ambos flujos.

Uno de los puntos fuertes de este cliente es su baja latencia y facilidad de uso. En los casos en que el flujo proviene de la webcam, es posible trabajar con una entrada de vídeo en tiempo real sin necesidad de configuración adicional, mientras que emplear un archivo guardado anteriormente puede ser valioso para desarrollo o *benchmarking*.

Conexión con el robot Unitree Go2

La conexión con el robot supuso una fase crítica del proyecto, en la que convergían aspectos de red, visión por computador, compatibilidad de sistemas y diseño modular.

La primera medida adoptada fue utilizar la aplicación oficial proporcionada por el fabricante del robot, que permite visualizar en tiempo real el flujo de vídeo captado por la cámara principal del sistema. Este paso tenía un propósito meramente comprobatorio: confirmar que el *hardware* estaba operativo, que el sistema interno de transmisión de vídeo funcionaba como se esperaba, y que existía una señal visual estable disponible dentro del robot. La aplicación cumplió su función y mostró un vídeo fluido, lo que permitía concluir que no había ningún problema en la captación ni en el tratamiento interno por parte del robot.

El verdadero reto era conseguir extraer ese flujo de vídeo del entorno cerrado de la aplicación y redirigirlo hacia el ordenador de trabajo, para poder tratarlo posteriormente con los modelos ya desarrollados. Este paso, que podría parecer trivial, implicó una investigación detallada de la arquitectura interna del robot, ya que la documentación proporcionada por el fabricante no incluía ninguna guía para este tipo de integración.

Se descubrió que el robot está dotado de dos tarjetas independientes, cada una con una funcionalidad específica. La primera tarjeta, no accesible desde el exterior, se encarga de gestionar todos los datos críticos del robot: señales de sensores, control de actuadores, navegación, decisiones autónomas, etc. Esta tarjeta opera como núcleo operativo del sistema y no permite conexión remota, ya que el fabricante no proporciona acceso SSH ni ninguna interfaz estándar. La segunda tarjeta, en cambio, sí permite establecer una conexión por terminal mediante el protocolo SSH, aunque su funcionalidad es mucho

más limitada. Esta segunda unidad está diseñada únicamente para la ejecución de *scripts* sencillos y pruebas de entorno, y no ofrece acceso directo al sistema de control principal del robot.

Pese a estas limitaciones, se planteó utilizar esta tarjeta secundaria como puerta de entrada. Se revisó el manual del fabricante y se comprobó que, para establecer una sesión SSH con éxito, era necesario conectar el robot al ordenador mediante un cable Ethernet directo, y definir ambos dispositivos dentro de un dominio IP concreto especificado por el fabricante. Se configuraron manualmente las direcciones IP de ambos extremos, se estableció una red local, y finalmente se consiguió establecer una sesión SSH estable entre el ordenador y la tarjeta accesible del robot.

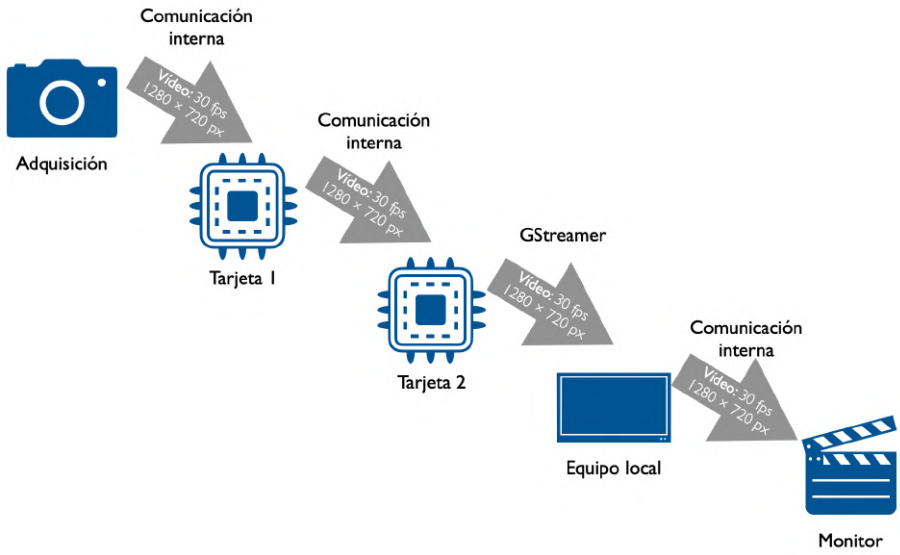
El siguiente paso consistió en redirigir el flujo de vídeo desde la cámara del robot hacia el ordenador, utilizando herramientas que permitieran una transmisión con baja latencia y buena calidad visual. Tras evaluar varias alternativas, se optó por emplear GStreamer, ya que, como se anticipó en capítulos anteriores, su versatilidad, amplia documentación y compatibilidad con los principales formatos de compresión la convierten en una solución muy extendida en entornos profesionales. Se diseñó una primera orden de envío, ejecutada en el robot, que transmitía el vídeo comprimido a una dirección IP local.

Este sistema, que es el que se muestra esquemáticamente en la Figura 4.3a permitió, tras varias pruebas de ajuste, obtener por primera vez un flujo de vídeo fluido, continuo y de buena calidad visual desde el robot hasta el equipo de desarrollo. La configuración de *buffers* y la elección del codificador resultaron clave para lograr una latencia suficientemente baja que hiciera viable el análisis en tiempo real.

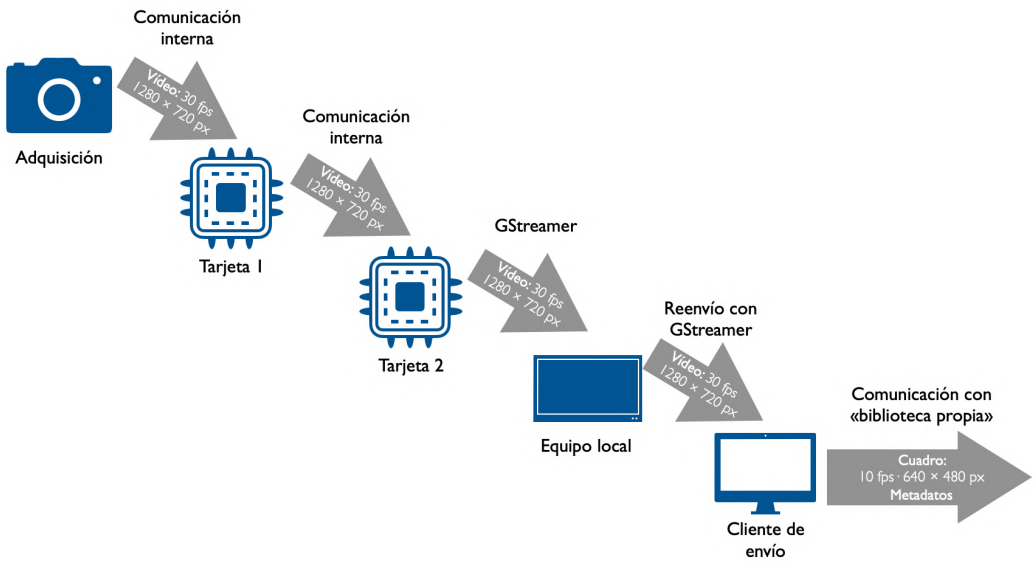
No obstante, pronto se detectó una limitación importante. Aunque el vídeo podía visualizarse correctamente desde la terminal local utilizando GStreamer, no era posible capturarlo directamente desde Python utilizando OpenCV, lo que impedía integrar el flujo en el análisis de escenas. Se intentaron varias formas de establecer esta comunicación, como la instalación de dicha biblioteca OpenCV con GStreamer integrado, sin éxito. Tras una revisión de la bibliografía se vio que, en efecto, la incompatibilidad entre versiones de GStreamer, OpenCV y sus vínculos con Python, sumada a las diferencias entre distribuciones de Linux y entornos de desarrollo, hacía que esta integración fuera técnicamente muy compleja y prácticamente imposible en la práctica. Aunque se logró una integración funcional en un contenedor Docker en el que todas las versiones coincidían, esta solución presentaba varios inconvenientes: por un lado, introducía una latencia adicional considerable, y por otro, complicaba enormemente la estructura del sistema, haciéndolo menos portable, menos reproducible y más difícil de mantener.

Se exploró entonces una vía alternativa: emplear el SDK oficial basado en ROS 2 proporcionado por el fabricante del robot. El objetivo era acceder directamente al tópico en el que se publica el vídeo de la cámara, utilizando las herramientas estándar de ROS. Sin embargo, esta solución resultó aún menos viable. El SDK requería una versión obsoleta de Ubuntu, incompatible con el entorno actual de desarrollo, y la propia documentación del fabricante era escasa y poco precisa. Además, el entorno de ROS en este caso resultó altamente inestable, con frecuentes errores en la suscripción a tópicos, fallos en los nodos y problemas de compatibilidad entre versiones. Esta situación se interpretó como una muestra de la falta de fiabilidad del fabricante en el mantenimiento de sus entornos de desarrollo, algo relativamente común en ciertos productos procedentes del mercado asiático, donde la integración con estándares occidentales como ROS no siempre está garantizada. Como resultado, esta vía fue finalmente descartada.

La solución definitiva consistió en reinterpretar el uso de GStreamer, que pese a la dificultad de su integración con OpenCV era la opción que había dado mejores resultados hasta ese momento. Se decidió adoptar una estrategia diferente: en lugar de intentar leer directamente el flujo original en Python, se decidió utilizar GStreamer como herramienta de reformateo y reemisión del flujo. La idea consistía en que el vídeo recibido fuese captado por un flujo adicional de GStreamer, recodificado y reenviado a una dirección local a través de UDP. Este nuevo flujo sí era reconocible por OpenCV de forma natural. Este flujo de vídeo es el que se muestra en forma de esquema en la Figura 4.3b.



(a) Adquisición del vídeo del robot para mostrarlo en pantalla



(b) Cliente de envío con el vídeo captada por el robot

Figura 4.3. Diversos tipos de comunicación con el robot Unitree Go2

Esta solución fue finalmente adoptada como definitiva, tanto por su estabilidad y baja latencia, como por su facilidad de integración con el resto de módulos del sistema. Una gran ventaja de esta opción es que el comando para acceder en Python a este vídeo es prácticamente idéntico al empleado para acceder a la webcam o a un archivo local, lo que, a diferencia de los intentos anteriores, permitía trabajar con estructuras de código conocidas, escalar los algoritmos desarrollados en pruebas locales y mantener la modularidad necesaria para su incorporación en arquitecturas distribuidas. Gracias a esta

configuración, el sistema pudo ser conectado de forma funcional al flujo visual del robot, completando así una de las piezas clave del proyecto.

Pese a ello, hasta este punto el sistema requería que el robot estuviese físicamente conectado mediante un cable Ethernet al equipo. Esta limitación se volvió especialmente relevante al constatar que la movilidad del robot quedaba restringida a la longitud del cable, lo cual resultaba incompatible con el objetivo último del sistema. Se intentó entonces activar un supuesto módulo wifi interno con el que, en teoría, el robot contaba, pero tras múltiples pruebas, búsquedas y consultas a documentación, no se encontró forma alguna de hacerlo operativo. Es probable que este módulo no estuviera incluido en el modelo concreto presente en la oficina, o que, de nuevo, su activación estuviera limitada por la fiabilidad y el mediocre soporte técnico del fabricante.

Dada la imposibilidad de usar conectividad inalámbrica integrada, se tomó la decisión de adquirir un módulo wifi externo, que pudiera conectarse físicamente al robot y ofrecer una funcionalidad equivalente. Tras instalar el módulo, fue necesario descargar e instalar los drivers adecuados, lo cual implicó identificar la arquitectura interna del sistema, compilar los controladores desde la terminal SSH y realizar pruebas para verificar la conectividad. Una vez configurado correctamente, se repitió todo el proceso de conexión descrito anteriormente, esta vez sin necesidad de cable. El resultado fue una conexión SSH plenamente funcional a través de wifi, que permitía al robot operar de forma autónoma y alejada del equipo de desarrollo, abriendo así la puerta a experimentos más realistas, maniobras de navegación libre y pruebas en entornos controlados más amplios.

Con esta integración se cerraba así un hito fundamental del proyecto: conectar el flujo visual del robot con el sistema de análisis de escenas, manteniendo una arquitectura abierta, extensible y compatible con entornos de experimentación reales.

Conexión con el Búho

El Búho es un dispositivo avanzado de captura de imagen que emplea un sistema de cámaras duales con óptica tipo ojo de pez, ofreciendo una cobertura de 360º mediante la fusión de dos hemisferios. La arquitectura de este dispositivo no permite acceder a las cámaras como una webcam estándar, sino que expone un flujo de datos de alta frecuencia a través de una biblioteca propia desarrollada para ello, que está pensada específicamente para entornos de transmisión audiovisual robusta, eficiente y con bajo retardo. Dicha biblioteca se introdujo en el anterior capítulo.

El primer paso para integrar este dispositivo en el sistema fue establecer un cliente especializado capaz de recibir datos del Búho a través de dicha biblioteca propia. Esto permite la suscripción a un flujo que transporta distintos tipos de información (imagen, eventos, audio, telemetría…). Sin embargo, para nuestro sistema únicamente se emplea la imagen capturada por las cámaras.

Una vez implementada esta capacidad de recepción, se planteó cómo integrar esa imagen en el flujo de trabajo global, basado en una arquitectura distribuida Cliente de envío → Servidor → Cliente de recepción, y donde todas las comunicaciones están mediadas por la otra biblioteca propia encargada de la transmisión de datos. Surgieron entonces varias alternativas.

Una primera aproximación fue diseñar un servidor de análisis de escenas que actuase como receptor directo del flujo de vídeo, sin pasar por ningún cliente intermedio. Este servidor implementaba una rutina de recepción continua, almacenando o procesando cada cuadro conforme llegaba.

Si bien esta solución tenía el atractivo de acortar la cadena Cliente → Servidor (el propio Búho hacía de cliente), generaba un problema arquitectónico importante: forzaba a crear una clase de servidores especializados, capaces de escuchar flujos directamente, diferentes del resto de servidores, que estaban

diseñados para recibir desde la biblioteca propia específica para transmisión de datos. Esto rompía la modularidad perseguida desde el inicio del proyecto y obligaba a duplicar muchos fragmentos de lógica, dificultando el mantenimiento y escalado. Además, perdía la ventaja de tener un único punto de control para todos los flujos, lo que complicaba también la depuración y el testeo.

Frente a lo anterior, se optó finalmente por una solución mucho más coherente y escalable: implementar un cliente especializado que actúa como traductor, recibiendo cuadros desde el Búho y reenviándolas mediante la biblioteca propia de transmisión al servidor correspondiente, exactamente igual que el resto de clientes (webcam, archivos, robot…).

Esta decisión permite tratar al Búho como una «caja negra» equivalente a una cámara más. El cliente se encarga de normalizar el origen de la imagen, y el servidor puede trabajar de forma completamente transparente, sin preocuparse del origen ni del protocolo inicial. Esta aproximación preserva la homogeneidad en la arquitectura, reduce la complejidad del sistema y mantiene la interoperabilidad completa entre componentes, que era uno de los principios clave del diseño. En la Figura 4.4 se muestra la estructura del cliente de envío conectado con el Búho, en el que se ha señalado la «caja negra» referida anteriormente.

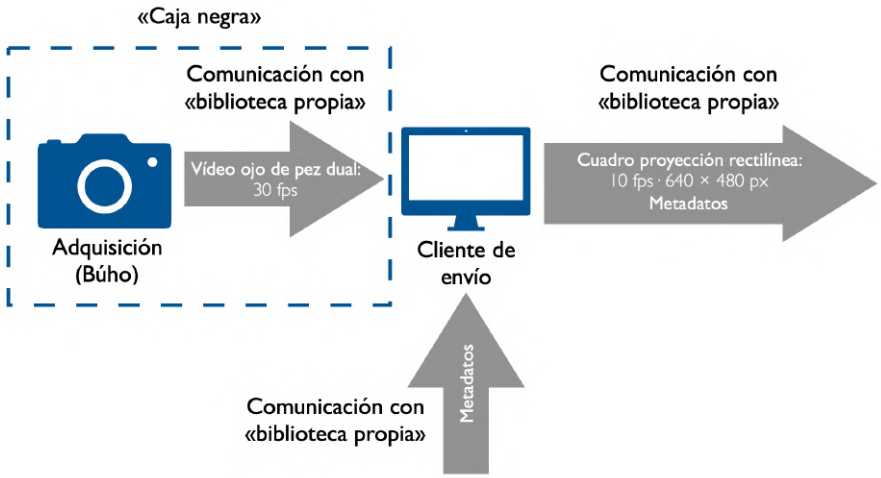


Figura 4.4. Cliente de envío con el vídeo captado por el Búho

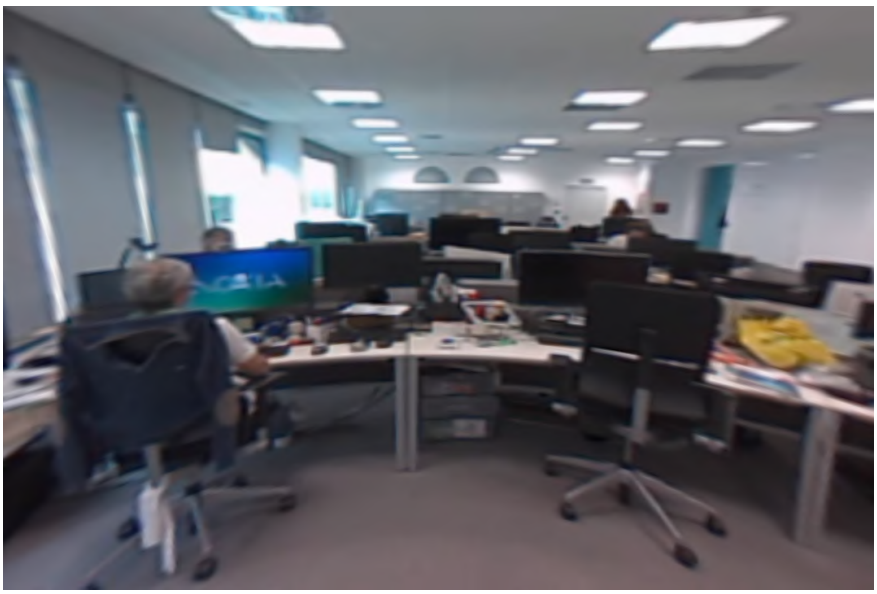
Una vez definida la arquitectura general del sistema, se abordó un aspecto crucial relacionado con el formato del vídeo recibido desde el dispositivo Búho. Este equipo captura el entorno mediante dos cámaras ojo de pez, generando una imagen inicial en formato ojo de pez dual o *dual fisheye*, donde se representan dos hemisferios visuales deformados. Este formato, aunque adecuado para la captación de entornos en 360°, resulta inadecuado para el análisis directo, ya que la mayoría de los algoritmos de análisis de escenas están entrenados sobre imágenes rectilíneas y con proporciones naturales. Para que la imagen sea utilizable, es necesario transformarla mediante un proceso secuencial realizado en dos pasos, que pueden verse en la Figura 4.5, siendo la Figura 4.5a un cuadro en formato de ojo de pez dual, tal y como lo capta del Búho.

Primero, se incorporó al cliente una rutina de *stitching* de ambas capturas, combinándolas en una única imagen continua. A continuación, se aplicó un proceso de mapeo que proyecta la esfera resultante sobre un plano bidimensional, generando así una imagen en proyección equirectangular. Este formato conserva toda la información del entorno 360°, pero presenta deformaciones propias del aplanamiento, especialmente en los bordes. La Figura 4.5b muestra un cuadro en esta etapa, ya reconstruido tras el *stitching* y mapeado.

Sin embargo, como la imagen equirectangular continuaba presentando limitaciones para el análisis automatizado, se implementó una segunda transformación que permite generar una vista localizada en una dirección concreta del entorno. Para implementar esta funcionalidad, se partió de un repositorio de código abierto en GitHub, el cual fue adaptado, reescrito y encapsulado como una biblioteca propia, con el objetivo de integrarla directamente en el sistema sin dependencias externas. Esta operación simula una cámara virtual orientada en una dirección determinada, generando una imagen en proyección rectilínea. El resultado es una ventana configurable que emula el campo de visión «normal» de una persona desde una posición específica dentro de la escena. Esta imagen completamente rectificada es la mostrada en la Figura 4.5c.

(a) Cuadro en formato ojo de pez dual, tal y como es captada por el Búho

(b) Cuadro tras la rectificación de ojo de pez, en formato equirectangular



(c) Cuadro en proyección rectilínea que se envía al servidor

Figura 4.5. Etapas en la rectificación de la imagen captada por el Búho

4.1.4. Servidores

El sistema se articula sobre una arquitectura distribuida donde cada módulo cumple una función concreta, permitiendo escalabilidad, versatilidad y claridad estructural. Dentro de esta arquitectura, los servidores desempeñan un papel clave: son los encargados de aplicar los modelos de análisis de escenas, ya sea en cuadro o vídeo, sobre los datos recibidos desde los clientes de envío. Aunque el tipo de modelo puede variar (desde VQA a descripción de cuadros o de vídeo), todos los servidores siguen una lógica común cuidadosamente diseñada para garantizar modularidad, eficiencia y facilidad de mantenimiento.

Cada servidor se basa en un esquema simple pero efectivo: escucha por un puerto concreto, recibe imagen (y potencialmente metadatos) desde un mismo cliente de envío, ejecuta un modelo de inferencia sobre ellas, y devuelve un cuadro procesado junto con metadatos informativos sobre el resultado.

Este funcionamiento está orquestado mediante la biblioteca propia de transmisión de datos, que ha sido utilizada en todo el sistema por su fiabilidad y capacidad para gestionar flujos de datos en tiempo real. A diferencia de los clientes, donde suele establecerse una conexión a un destino IP específico (bind=False), en los servidores se emplea sistemáticamente bind=True. Esto significa que el servidor actúa como un nodo pasivo que «espera» conexiones entrantes de cualquier cliente, sin necesidad de conocer su IP de antemano. Esta elección no es menor: permite diseñar sistemas en los que los servidores pueden ejecutarse en paralelo, ya sea en el mismo equipo o en máquinas distintas, o en la nube, sin necesidad de actualizar su configuración ante cada cambio de red, siempre que el puerto se mantenga fijo.

Además, el hecho de que el servidor envíe los cuadros procesados y sus metadatos a todas las IP conectadas al puerto habilitado facilita la integración con múltiples componentes del sistema sin necesidad de duplicar esfuerzos o crear reglas complejas de direccionamiento. Basta con asegurarse de que los puertos empleados por cada servidor son únicos y no colisionan entre sí.

En la fase inicial de desarrollo, los servidores estaban diseñados únicamente para recibir y devolver imagen, sin información adicional. Sin embargo, se reconoció pronto que, para un sistema de análisis de escenas con vocación visual e interpretativa, no bastaba. Era esencial que cada cuadro procesado viniera acompañada de metadatos que contextualizaran la inferencia realizada.

Así se introdujo el envío de metadatos junto con cada cuadro: actualmente, el servidor añade a cada paquete enviado el número de cuadro procesado (contador interno), el tiempo exacto del procesamiento, la salida del modelo (descripciones, respuesta, etiquetas, etc.) y el nombre del algoritmo aplicado. Esta información resulta especialmente valiosa en escenarios de depuración, presentación visual, análisis posterior o trazabilidad del sistema. En la Figura 4.6 se muestra un ejemplo de los metadatos que envía el servidor, en este caso del algoritmo «ViLT».

```
packet_meta:
    seq: 1
    ts: 3
    data:
        duracion: 3
        info: 'man'
        algoritmo: 'ViLT'
    seq: 2
    ts: 5
    data:
        duracion: 2
        info: 'chair'
        algoritmo: 'ViLT'
```

Figura 4.6. Ejemplo de metadatos enviados por el servidor

Aunque la capacidad de recibir metadatos también está implementada en todos los servidores, no se utiliza en la versión actual, ya que el flujo de datos se considera unidireccional en esta etapa del desarrollo. En cambio, el envío de metadatos es una pieza central del diseño, y se ha convertido en un estándar en todos los servidores desplegados.

Otra de las decisiones estructurales más importantes ha sido la de mantener un servidor independiente para cada modelo de análisis, incluso si algunos modelos pudieran técnicamente convivir en una misma instancia. Esta decisión responde a múltiples motivaciones:

- **Aislamiento de dependencias:** cada modelo tiene sus propios requisitos (bibliotecas, versiones de Python, uso o no de GPU, etc.). Aislarlos en servidores distintos evita conflictos.
- **Escalabilidad por demanda:** algunos modelos requieren más recursos (RAM, VRAM, núcleos) que otros. Separarlos permite distribuirlos en función de la disponibilidad de recursos en cada máquina.
- **Aislamiento de errores:** si un modelo se bloquea o falla, solo su servidor se ve afectado.
- **Actualización independiente:** si un modelo se reentrena, se cambia por otro más eficiente, o se prueba una nueva versión, no se afecta al resto del sistema.
- **Modularidad de despliegue:** en entornos como Docker, cada servidor puede encapsularse y desplegarse de forma autónoma, lo que facilita tareas de orquestación y mantenimiento.
- **Simplicidad conceptual:** cada servidor tiene un propósito único, fácilmente documentable y predecible.

Esta estrategia ha resultado particularmente útil en pruebas sobre máquinas con distinta capacidad (por ejemplo, una máquina con GPU dedicada para Florence o LLaVA, y otra sin GPU para servidores

de CLIP). En entornos de laboratorio o producción, permite ajustar los recursos asignados a cada tipo de análisis sin necesidad de rediseñar el sistema completo.

Aunque, como se dijo anteriormente, la estructura lógica es común, existen diferencias importantes entre los servidores en función del tipo de modelo aplicado. Se han establecido tres grandes categorías:

1. **Servidores de VQA: ViLT y CLIP**

   Estos servidores se encargan de responder preguntas sobre cuadros. Funcionan a partir de una entrada dual: el cuadro recibido y la pregunta asociada, que se encuentra en los metadatos o es fijada previamente por el sistema. Tras recibir el cuadro, el modelo lo analiza y genera una respuesta en lenguaje natural.

   Posteriormente, la salida visual generada consiste en el cuadro original con un recuadro en que se muestra la pregunta y la respuesta del modelo, sin que se realice segmentación, detección de objetos ni transformaciones visuales adicionales.

   Son servidores ligeros, rápidos y adecuados para tareas donde se desea validar rápidamente la interpretación contextual del modelo.

2. **Servidores de descripción de cuadros: Florence y LAVIS**

   Estos servidores tienen una lógica más rica y son los que mayor variedad de salidas visuales pueden generar. En el caos que principalmente se ha explorado en este trabajo, se parte de un cuadro y se genera una descripción detallada de su contenido, que se muestra sobre el cuadro o en un panel adyacente, en función del nivel de detalle.

   En el caso de Florence, si se activa la detección de objetos o segmentación, se sobrescriben en el cuadro los cuadros delimitadores, áreas coloreadas y etiquetas correspondientes.

   Se trata de los servidores más versátiles, y por tanto más exigentes en términos de recursos.

3. **Servidores de análisis de vídeo: LLaVA y OpenBMB**

   Estos servidores presentan características diferentes de los dos tipos anteriores:

   - No trabajan con cuadros aislados, sino con secuencias de los mismos. Por lo tanto, como ya se explicó al hablar de los programas más simples, se agrupa una cantidad definida de cuadros consecutivos, que se concatenan y convierten en un fragmento de vídeo temporal. Ese número es configurable por el usuario, y depende de la memoria disponible en la GPU en la que se ejecute el modelo, pues a mayor longitud del vídeo, mayor cantidad de recursos se demandan. En este trabajo, se ha fijado dicha cantidad en 25, pues permite la ejecución del programa sin problemas.
   - Procesan el vídeo completo y devuelven los resultados solo al final del procesamiento, no cuadro a cuadro.
   - Por eficiencia, envían una señal de pausa al cliente para que deje de enviar durante el análisis. De esta manera, se evita la saturación de la cola interna de la transmisión, que podría provocar la pérdida de paquetes y la degradación del rendimiento. En la Figura 4.7 se muestra un ejemplo de los metadatos que se envían en este tipo de algoritmos, cuya diferencia estriba en que se escribe el *status* como «Pause» o «Continue» para determinar si se pausa o no el envío por parte del servidor, así como el número de cuadros por fragmento de vídeo, tal y como se ha remarcado.

```
packet_meta:
    seq: 1
    ts: 3
    data:
        duracion: 3
        info:'The video shows a person standing in a kitchen,
        holding a knife and a bowl. The person is wearing a white
        chef's outfit and appears to be preparing food. The person
        is holding a knife and a bowl, and is standing in front of a
        stove. The person is also holding a piece of paper and
        appears to be writing something on it. The video seems to be
        focused on food preparation and cooking.'
        algoritmo: 'LLaVA'
        status: 'Continue'
        'lenght_clip': 25

    seq: 2
    ts: 5
    data:
        duracion: 2
        info: 'The image shows a man standing in a kitchen. He is
        wearing a white chef's jacket with a blue logo on the left
        side of his chest. He has blue gloves on and is wearing
```


Figura 4.7. Ejemplo de metadatos enviados por algoritmos de análisis de vídeo

- Los cuadros no son modificados visualmente (no se dibujan cuadros ni textos), pero los metadatos del análisis de vídeo se envían en bloque al finalizar la inferencia.

Estos servidores son usados para tareas de resumen de vídeo, análisis temporal o generación de descripciones narrativas a partir de escenas dinámicas.

Una vez explicados los diferentes tipos de servidores, se presenta la Figura 4.8, en que se muestra de manera esquemática cómo se conectan de manera bidireccional con el cliente de envío, así como la comunicación con los clientes de recepción, a los que se mandan los cuadros procesados, los resultados y demás metadatos.

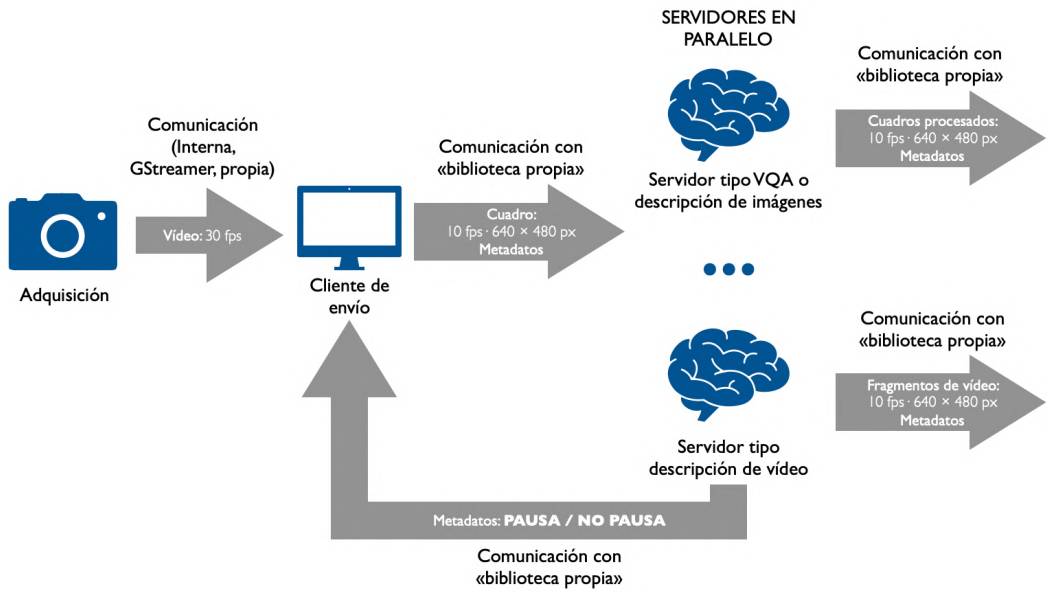


Figura 4.8. Esquema del funcionamiento del sistema con cliente de envío y servidores

Como conclusión, es importante destacar que, en conjunto, este sistema de servidores representa un pilar fundamental del diseño, permitiendo que los modelos de análisis de escena se ejecuten con eficiencia, claridad organizativa y posibilidades reales de evolución futura.

4.1.5. Clientes de recepción

Una vez los servidores han procesado los cuadros o vídeos, se hace necesario un componente que recoja dichos resultados para visualizarlos, almacenarlos o integrarlos en otros sistemas. Esta tarea recae en los denominados clientes de recepción, programas diseñados para recoger, desde los servidores, tanto la imagen procesada como los metadatos generados por el algoritmo aplicado.

En un primer momento, al igual que ocurrió con los clientes de envío, se diseñó un cliente muy básico, capaz simplemente de recibir una imagen procesada y mostrarla en pantalla mediante OpenCV. Sin embargo, al observar el potencial de los metadatos generados por los modelos (la información sobre el algoritmo utilizado, los resultados…) se amplió el diseño para integrar una doble recepción: por un lado, las imágenes JPEG codificadas, y por otro, los metadatos en formato JSON. Ambos flujos se reciben, como viene siendo habitual en el proyecto, mediante la biblioteca propia empleada para ello, que, en este caso, se configura como cliente (con bind=False). Esta arquitectura dual permite separar completamente ambos tipos de datos, facilitando la depuración, la modularidad y futuras ampliaciones, como podrían ser el envío de datos de sensores, comandos de control, o resultados de otros módulos.

Una consideración arquitectónica importante fue el modo en que se gestionaban estos clientes. En una primera etapa se valoró la posibilidad de diseñar un único cliente capaz de recibir datos desde todos los servidores. Sin embargo, durante las pruebas se detectó un problema clave: si alguno de los servidores no enviaba datos (por ejemplo, porque estaba desconectado o ejecutando un modelo más pesado), esto pausaba o ralentizaba la recepción del resto de canales, introduciendo una dependencia cruzada. Por este motivo, se optó por mantener un cliente de recepción por cada servidor, como se muestra en el esquema de la

Figura 4.9 , en la que ya están las tres partes fundamentales del sistema. Este enfoque tiene varias ventajas, en primer lugar, permite un mayor control sobre qué se recibe y cómo se visualiza o almacena; en segundo lugar, asegura que un fallo en un servidor no interfiere con el funcionamiento del resto; finalmente, mantiene la coherencia con la filosofía de modularidad que ha guiado el diseño general del sistema.

Cada cliente está preparado para manejar tanto modelos por cuadros como por fragmentos de vídeo:

- En el primer caso, el funcionamiento es sencillo o intuitivo: cada vez que se recibe un resultado de inferencia se muestra el cuadro procesado y se imprime en pantalla el resultado en forma de texto. Tanto la secuencia de vídeo como la escritura son continuos y prácticamente en tiempo real, pues recuérdese que se estaban procesando 3 cuadros por segundo.
- En el caso, que sucede con servidores de LLAVA u OpenBMB, el cliente funciona de un modo levemente diferente. De manera similar a antes, se reciben de manera casi instantánea los cuadros enviados por el servidor y se van mostrando en pantalla; sin embargo, en lugar de mostrar un solo vídeo continuo, cada cierto número de cuadros (25 en este proyecto) se cierra la ventana y se abre una nueva. Ese es precisamente el fragmento de vídeo para el que se imprime el resultado del modelo en la pantalla, lo que garantiza la sincronización entre lo que se visualiza y lo que se interpreta. En resumen, se muestra el fragmento de vídeo que procesa el algoritmo, se imprime el resultado, se cierra el vídeo y se comienza de nuevo. Esta lógica de agrupación se adapta automáticamente gracias al campo «lenght_clip» incluido en los metadatos enviados.

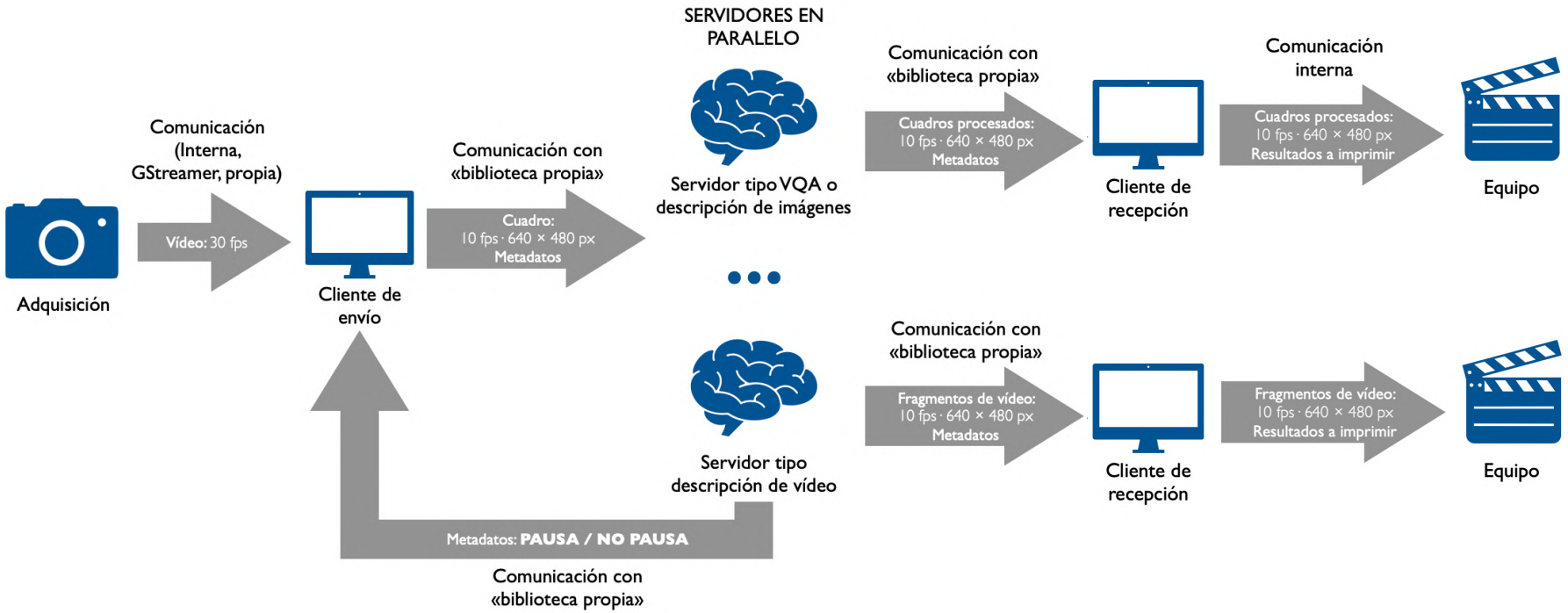


Figura 4.9. Esquema del funcionamiento del sistema cliente-servidor

En la Figura 4.10 se muestra la salida típica que se ve al ejecutar en el equipo local uno de los clientes de recepción.

En una primera etapa del desarrollo, estas eran las únicas funcionalidades de los clientes de recepción, sin que realizasen nada más allá de mostrar los resultados del procesamiento en la pantalla del ordenador. Esta solución resultaba suficiente para las primeras pruebas, pero conforme el proyecto fue avanzando, surgió la necesidad de conservar los resultados de forma persistente, especialmente para fases posteriores de análisis, validación o integración.

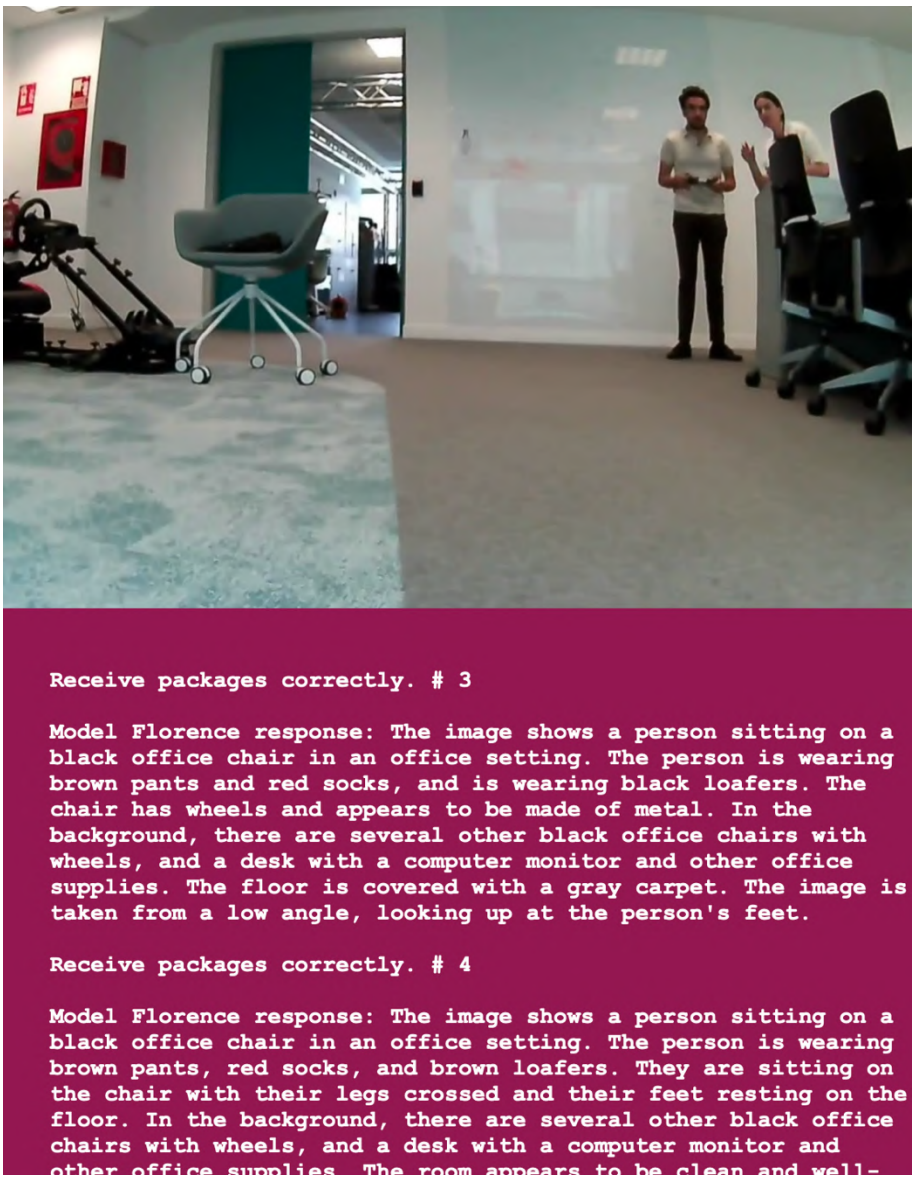


Figura 4.10. Ejemplo de entrada y salida de uno de los clientes de recepción, con vídeo y texto

Inicialmente se consideró que cada cliente de recepción pudiera escribir los resultados en archivos individuales, ya fuera en formato JSON o en otro formato estructurado. Sin embargo, esta solución se reveló problemática debido a la propia naturaleza distribuida del sistema: dado que hay varios servidores ejecutándose en paralelo, y por tanto varios clientes de recepción activos de forma simultánea, la dispersión de resultados entre archivos separados dificultaba el tratamiento conjunto de los datos y complicaba la trazabilidad de las inferencias.

Con el fin de resolver este problema y facilitar un almacenamiento centralizado, se optó por introducir, de forma opcional y configurable por el usuario, la posibilidad de que cada cliente de recepción envíe los resultados recibidos a un servidor Flask, especialmente diseñado para esta tarea, que funciona como un punto de recogida centralizado para todos los resultados generados por los modelos. Cada cliente, al recibir la imagen procesada y los metadatos desde su correspondiente servidor, reenvía esos metadatos a este servidor común. Allí, cada resultado es organizado por cuadro y clasificado por algoritmo, permitiendo que los resultados de varios modelos aplicados sobre un mismo cuadro queden reunidos en un único archivo estructurado. La lógica interna del servidor se encarga de gestionar esta agrupación, asegurando que no se pierda ninguna respuesta y que todas se integren bajo un mismo formato.

Además, el servidor permite consultar en cualquier momento el estado actual de los resultados acumulados y al almacenarlos en un archivo unificado, este podrá usarse en análisis posteriores, presentaciones, evaluaciones comparativas o entrenamientos supervisados. Esta solución elimina la dispersión de archivos y evita inconsistencias que podrían surgir al intentar guardar datos en paralelo desde múltiples procesos, a la vez que añade robustez, organización y continuidad al flujo de trabajo, garantizando que los resultados generados en tiempo real puedan ser conservados, reutilizados y analizados de manera eficiente.

En resumen, los clientes de recepción constituyen el último eslabón de la arquitectura cliente-servidor implementada en este sistema. Aunque su funcionalidad pueda parecer secundaria respecto a los algoritmos que procesan los cuadros, resultan esenciales para verificar visualmente los resultados, entender el funcionamiento del modelo en tiempo real y, si así se desea, guardar las salidas.

En la Figura 4.11 se muestra la estructura del modelo cliente-servidor, en el que se ha añadido la funcionalidad de guardar en el servidor Flask todos los resultados.

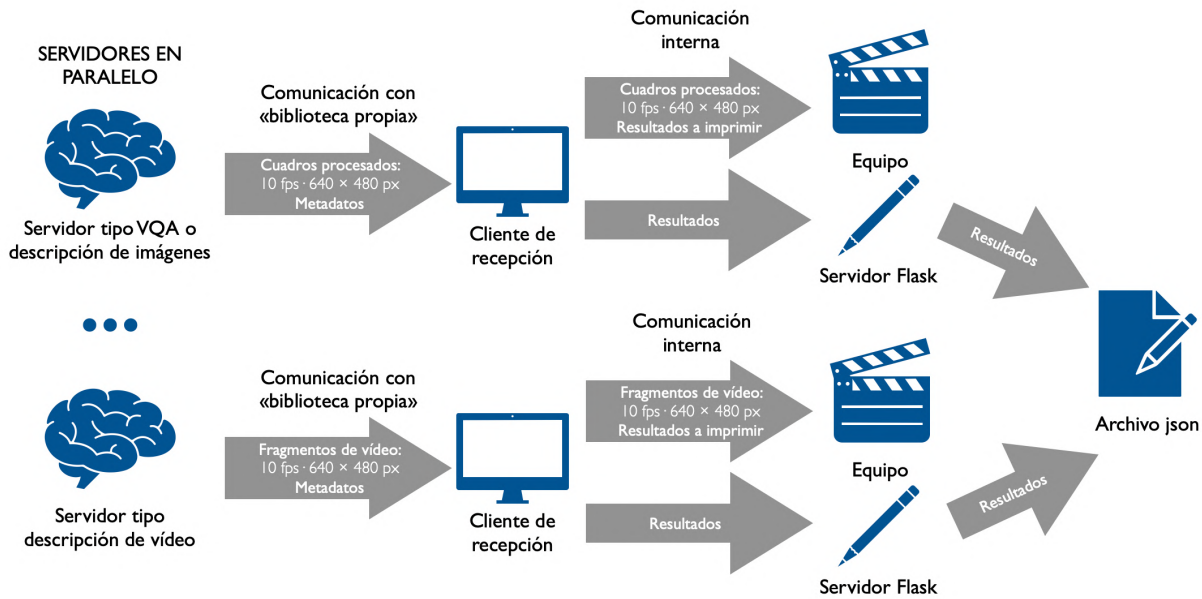


Figura 4.11. Estructura del servidor-cliente de recepción con escritura en el json

4.2. Evaluación de los resultados

Una vez finalizada la fase de desarrollo del sistema cliente-servidor y verificado el correcto funcionamiento de todos los componentes, se abre paso la siguiente etapa del proyecto: la evaluación del rendimiento de los distintos algoritmos.

Esta fase de evaluación constituye una parte esencial del proyecto, y su interés radica, fundamentalmente, en dos motivos diferentes:

- **Validación funcional:** hasta el momento, el trabajo con los modelos se ha basado principalmente en la observación directa de los resultados, guiándose por la intuición y la experiencia subjetiva. Por ello, se hace necesario contar con un método empírico y sistemático que permita verificar de forma objetiva el buen desempeño de los algoritmos implementados.
- **Comparación entre modelos:** para cada una de las tareas abordadas, VQA, descripción de cuadros y descripción de vídeo, se han seleccionado dos modelos distintos, elegidos por su relevancia dentro del estado del arte. La evaluación permite contrastar sus resultados de manera rigurosa y así identificar cuál ofrece un rendimiento superior en cada caso.

El objetivo último de esta sección es, por tanto, determinar qué modelo resulta más adecuado en cada tarea, no solo en términos de precisión técnica, sino también en cuanto a su adecuación contextual. Esta selección servirá como base para futuras versiones del sistema, en las que se integrarán únicamente aquellas soluciones que demuestren un buen rendimiento.

4.2.1. Etiquetado de datos

Antes de evaluar cualquier modelo, es necesario contar con dos elementos fundamentales:

- Un conjunto suficientemente amplio y representativo de datos sobre los que evaluar el modelo, ya sean cuadros o vídeos, junto con los resultados de referencia asociados. En este proyecto, se ha prestado especial atención a emplear únicamente subconjuntos de validación o test de cada base de datos, evitando en todo momento utilizar conjuntos de entrenamiento, ya que se desconoce si alguno de los modelos evaluados ha sido entrenado con esos mismos datos. Usar datos de entrenamiento comprometería la validez de la evaluación.
- Las predicciones generadas por los modelos evaluados sobre dichos datos, de modo que puedan compararse con las etiquetas de referencia y valorarse objetivamente. Disponer tanto de las etiquetas como de las salidas del modelo es la única forma de establecer comparaciones justas y fiables.

Una vez entendido esto, comenzó la búsqueda de bases de datos adecuadas. Esta tarea, que podría parecer trivial a primera vista, resultó sorprendentemente compleja. En la práctica, muy pocos conjuntos de datos cumplen con todos los requisitos necesarios para una evaluación rigurosa. Los problemas más habituales incluían la ausencia de un subconjunto de validación (estando disponible únicamente el conjunto de entrenamiento), la existencia de imágenes sin resultados de referencia o viceversa, e incluso, anotaciones incorrectas o incompletas, mal formateadas o no asociadas de forma inequívoca a un cuadro o vídeo determinado.

Tras un proceso minucioso de selección, los conjuntos de datos finalmente escogidos fueron los siguientes:

- Para la tarea de *Visual Question Answering*, se empleó un subconjunto de COCO adaptado a esta finalidad. Contiene un número muy elevado de imágenes, cada una asociada a múltiples preguntas, con diez respuestas distintas por cada pregunta. A efectos del presente trabajo, este conjunto se denominará COCO.

- Para la descripción de cuadros, se utilizaron tres conjuntos distintos:

  - Un conjunto de datos procedente de Flickr, con entre 2000 y 3000 imágenes reales, acompañadas de un archivo CSV con descripciones asociadas a cada una de ellas. Se hará referencia a este conjunto como Flickr.
  - Además, se generaron dos conjuntos propios, de 36 y 44 imágenes reales cada uno de ellos, que se llamarán Cocina y Viaje, en función de la temática de las imágenes, que han sido etiquetadas manualmente por miembros del XRLab de Nokia. En ambos casos se partía de imágenes equirectangulares, tras haber sido captadas con el Búho, que fueron mapeadas para obtener imágenes en proyección rectilínea, tal y como se explicó al hablar de los clientes de envío.

- Para la descripción de vídeo, a las dificultades mencionadas anteriormente se sumó el hecho de que los archivos de vídeo requieren una gran cantidad de almacenamiento y procesamiento. Esto redujo significativamente las opciones disponibles, siendo finalmente seleccionados algunos fragmentos concretos que permitieron realizar una evaluación comparativa válida. El conjunto en cuestión de denomina TACOS, y las acciones que se muestran están relacionadas con tareas básicas de cocina.

Una vez seleccionadas las bases de datos, fue necesario adaptarlas para su tratamiento en las siguientes fases del proyecto. En primer lugar, se renombraron todas las imágenes y vídeos, sustituyendo los nombres originales, que en la mayoría de los casos correspondían a códigos automáticos generados por los responsables de la base de datos, por una numeración secuencial simple, del 1 en adelante. Esta modificación tenía como objetivo hacer más sencilla y sistemática la lectura y el procesamiento de los archivos desde *scripts* automatizados.

En paralelo, se modificaron los archivos de anotaciones (normalmente en formato .csv o .txt) para adaptarlos a un formato homogéneo en .json. Además de convertir los formatos, se eliminaron todos los campos no relevantes, como fechas, duraciones de los vídeos, códigos internos irrelevantes, etc. El formato final adoptado seguía estructuras simples como las que se muestran en la Figura 4.12. Se puede apreciar cómo, en el caso del conjunto de datos COCO (el usado en VQA), hay una pregunta y hasta diez respuestas diferentes por cada imagen, tal y como muestra la Figura 4.12a, mientras que para los conjuntos de descripción, ya sea de cuadros o de vídeo, aparece tan solo un texto por cada entrada, como se muestra en la Figura 4.12b.

Con estos formatos limpios y coherentes, se garantizó una lectura rápida, intuitiva y robusta desde cualquier programa que necesitara acceder a los datos.

```
{
    "image_filename": "45.jpg",
    "question": "What is the woman in the room doing?",
    "answers": [
      "talking to someone",
      "cleaning",
      "looking out window",
      "talking",
      "talking",
      "cleaning",
      "talking to someone outside of window",
      "talking",
      "talking out window",
      "cleaning"
    ]
  },
```


(a) Ejemplo de resultados de referencia en VQA


```
{
    "image_id": "8.jpg",
    "caption": [
      "A girl is on rollerskates talking on her cellphone standing in a parking lot.",
      "A trendy girl talking on her cellphone while gliding slowly down the street.",
      "A young adult wearing rollerblades, holding a cellular phone to her ear.",
      "There is a young girl on her cellphone while skating.",
      "Woman talking on cellphone and wearing rollerskates."
    ]
  },
```


(b) Ejemplo de resultados de referencia en descripción de cuadros

Figura 4.12. Ejemplos de resultados de referencia tras reformateado

El siguiente paso fue generar las salidas de los modelos, es decir, etiquetar los datos con los algoritmos desarrollados:

- En el caso de VQA, para cada una de las preguntas ya existentes en el conjunto de datos de referencia, se generó una respuesta utilizando los modelos implementados. Todos estos resultados se almacenaron en un nuevo archivo .json, que recogía, para cada image_id (el identificador de la imagen), la pregunta y la respuesta generada.
- En la tarea de descripción de imágenes, se generó una descripción automática para cada cuadro de cada uno de los tres conjuntos de datos seleccionados (Flickr, Cocina y Viaje). Estos resultados se guardaron igualmente en un .json con los campos *image_id* y *caption*.

  En ambos casos, se adaptaron tanto el cliente de envío como el de recepción para que recorrieran automáticamente todos los archivos de cada conjunto de datos, generando las predicciones correspondientes y escribiéndolas de forma ordenada y continua en los archivos de salida. De manera similar a la Figura 4.12, en la Figura 4.13 se muestra un ejemplo del archivo resultante del etiquetado por parte de los algoritmos de VQA y de descripción, en las Figura 4.13a y Figura 4.13b, respectivamente.

```
{
        "image_filename": "45.jpg",
        "question": "What is the woman in the room doing?",
        "answer": "standing"
    },
```


(a) Ejemplo de resultados etiquetados por ViLT (de tipo VQA) para su evaluación


```
    {
        "image_id": "8.jpg",
        "caption": "The image shows a woman in a blue shirt and black shorts on
roller skates, talking on a cell phone while standing on the road. In the
background, there is a vehicle, a pillar, plants, grass, and a building."
    },
```


(b) Ejemplo de resultados etiquetados por Florence, descripción de cuadros, para su evaluación

Figura 4.13. Ejemplos de resultados etiquetados por los algoritmos

- Por último, en el caso de descripción de vídeo, se intentó inicialmente aplicar el mismo sistema automatizado de etiquetado. Sin embargo, debido a diversas dificultades técnicas y a los limitados resultados obtenidos con dicho método, se optó por realizar el proceso de forma manual: se dividieron los vídeos en fragmentos, se procesaron uno a uno y se registraron sus descripciones en un archivo .json de forma directa.

Con ello se completó la primera parte del proceso de evaluación: disponer de todos los datos necesarios, tanto los conjuntos de referencia como los resultados generados por los modelos. Esta preparación es la base sobre la que se construye toda la comparación posterior.

4.2.2. Evaluación con métricas clásicas

Una vez que se han generado todos los resultados necesarios para evaluar el rendimiento de los modelos, el siguiente paso es, naturalmente, en calcular las métricas que permitan cuantificar la calidad de dichas predicciones de manera objetiva. Como se ya se explicó, esta evaluación basada en métricas clásicas, aunque no siempre capta matices contextuales o semánticos más sutiles, resulta fundamental por su rapidez, su transparencia y su reproducibilidad. Cada una de las tareas evaluadas, VQA, descripción de cuadros y descripción de vídeo, plantea distintos retos, y por tanto ha requerido enfoques métricos diferenciados.

Evaluación de *Visual Question Answering*

Para esta tarea, se utilizó como métrica principal el porcentaje de acierto, medido por la Exactitud VQA, que mide cuántas veces la respuesta generada por el modelo coincide con las respuestas de los humanos. Sin embargo, se apoyó en otras medidas complementarias, que, aunque se explicaron en capítulos anteriores, se recuerdan de nuevo brevemente:

- **Exactitud VQA:** mide el grado de coincidencia entre la respuesta del modelo y las respuestas de referencia, considerando correcta una respuesta si coincide con al menos tres de diez anotaciones humanas. Entre 0 y 1, cuanto más cercana a la unidad, mayor coincidencia con el criterio humano estándar.
- **WUPS:** evalúa si la respuesta generada es razonable, segura y no imposible, penalizando en especial las respuestas claramente erróneas o fuera de contexto. De nuevo, un valor más cercano a 1 indica mayor solidez y coherencia contextual de las respuestas.

- **Valor F1:** combina precisión (respuestas correctas entre las dadas) y *recall* (respuestas correctas entre las posibles) para valorar la calidad general de las respuestas. Es especialmente útil cuando hay múltiples respuestas válidas parciales. Una vez más, un F1 cercano a 1 indica un buen equilibrio entre precisión y *recall*.

Para ello, se empleó el conjunto de datos COCO que se expuso anteriormente, que incluye múltiples preguntas por cuadro y, para cada una, diez respuestas humanas diferentes, imprescindible para poder calcular la Exactitud VQA.

A continuación, la Tabla 4.1 muestra la comparativa entre las medias y desviaciones típicas (σ) de las puntuaciones de los dos modelos evaluados:

Tabla 4.1. Resultados de las métricas de evaluación para algoritmos de VQA

Modelo	**Exactitud VQA**		**WUPS**		**Valor F1**	
	Media	**σ**	**Media**	**σ**	**Media**	**σ**
CLIP	0,49	0,43	0,54	0,48	0,61	0,21
ViLT	0,72	0,34	0,78	0,36	0,75	0,08

Como puede observarse, el modelo ViLT supera a CLIP en las tres métricas, tanto en términos de valor medio como de consistencia. A pesar de que CLIP ha servido de base para multitud de arquitecturas recientes y extremadamente avanzadas, incluyendo sistemas empleados por modelos como ChatGPT, su aplicación directa como mero calculador de probabilidades en tareas de VQA presenta limitaciones importantes.

Además, los resultados de CLIP presentan una mayor variabilidad, como reflejan sus desviaciones típicas frente a las de ViLT. Esta diferencia sugiere que este último ofrece un rendimiento más estable y predecible.

En conjunto, ViLT se consolida como una arquitectura más sólida y robusta, especialmente diseñada para tareas de comprensión visual-lingüística, y que demuestra una mayor coherencia, precisión y estabilidad en su comportamiento. No obstante, la decisión final sobre qué modelo resulta más adecuado no se toma únicamente a partir de estas métricas clásicas, sino que se esperará a los resultados de los modelos de lenguaje de gran escala para contrastar y confirmar los resultados.

Evaluación de descripción de cuadros

Para la evaluación de los modelos de descripción automática de cuadros, se han empleado cuatro métricas ampliamente consolidadas en el campo de la descripción de cuadros que ya se explicaron en el estudio del estado del arte: BLEU, METEOR, ROUGE-L y CIDEr:

- **BLEU**: mide la coincidencia de n-gramas entre la descripción generada y las de referencia, premiando la precisión léxica. Aunque sus valores no están normalizados entre 0 y 1, son preferibles los más altos.
- **METEOR**: mejora la sensibilidad a sinónimos y orden de palabras mediante alineamientos flexibles. Valores cercanos a la unidad indican mejor correspondencia semántica.

- **ROUGE-L**: se basa en la subsecuencia común más larga entre ambas descripciones, valorando similitudes estructurales. De nuevo, está normalizada entre 0 y 1, denotando el 1 una mayor similitud.
- **CIDEr**: pondera la coincidencia de términos en función de su frecuencia en un conjunto de referencias, evaluando el alineamiento con un consenso humano. Una vez más, el 1 es la opción deseable.

Se han evaluado los modelos Florence y LAVIS sobre los tres conjuntos de datos etiquetados. La media de los resultados para Flickr, Cocina y Viaje se muestran en la Tabla 4.2a, Tabla 4.2b y Tabla 4.2c, respectivamente.

Tabla 4.2. Resultados de las métricas de evaluación para algoritmos de descripción de cuadros

(a) Resultados sobre el conjunto de datos «Flickr»

Modelo	**BLEU**		**METEOR**		**ROUGE-L**		**CIDEr**	
	Media	**σ**	**Media**	**σ**	**Media**	**σ**	**Media**	**σ**
Florence	7,65	5,35	0,35	0,10	0,27	0,08	1,12	0,35
LAVIS	10,78	6,42	0,43	0,14	0,33	0,07	0,91	0,33

(b) Resultados sobre el conjunto de datos «Cocina»

Modelo	**BLEU**		**METEOR**		**ROUGE-L**		**CIDEr**	
	Media	**σ**	**Media**	**σ**	**Media**	**σ**	**Media**	**σ**
Florence	8,68	3,18	0,30	0,06	0,39	0,06	0,89	0,038
LAVIS	9,09	4,23	0,19	0,05	0,35	0,07	0,96	0,39

(c) Resultados sobre el conjunto de datos «Viaje»

Modelo	**BLEU**		**METEOR**		**ROUGE-L**		**CIDEr**	
	Media	**σ**	**Media**	**σ**	**Media**	**σ**	**Media**	**σ**
Florence	3,91	2,11	0,21	0,08	0,22	0,08	0,93	0,27
LAVIS	2,9	1,95	0,15	0,08	0,22	0,09	0,87	0,23

Una vez obtenidos los resultados de los algoritmos de descripción de cuadros sobre los tres conjuntos de datos, se observa una serie de patrones. En primer lugar, el modelo LAVIS, pese a generar descripciones más breves y genéricas que Florence, obtiene mejores puntuaciones en la métrica BLEU en dos de las tres bases de datos. Florence, en cambio, tiende a generar descripciones mucho más completas, detalladas y ajustadas al contenido visual, algo que parece penalizado por la métrica BLEU, probablemente por su baja tolerancia a desviaciones del patrón literal de las descripciones de referencia. Esta situación sugiere que el exceso de detalle, aunque beneficioso desde el punto de vista semántico o

humano, no siempre es premiado por las métricas automáticas tradicionales, especialmente cuando el sistema de evaluación se basa en coincidencia exacta de n-gramas.

Con respecto a la estabilidad y coherencia de las predicciones, explicada por la desvición típica, Florence muestra en general una menor dispersión en sus resultados, especialmente en las métricas METEOR y ROUGE-L, lo que indica una mayor regularidad en la calidad de sus descripciones. LAVIS, aunque obtiene puntuaciones medias superiores en algunos indicadores, presenta una variabilidad más elevada, lo que puede interpretarse como una menor consistencia. Esta diferencia es relevante si se busca un comportamiento predecible y robusto del modelo.

Asimismo, es destacable que no hay un modelo que domine de forma clara en todos los indicadores dentro de un mismo conjunto de datos, dándose un auténtico «empate técnico». Por ejemplo, en el conjunto Flickr, LAVIS lidera en BLEU, METEOR y ROUGE-L, pero es superado por Florence en CIDEr. En Cocina, la situación se invierte parcialmente: Florence obtiene mejor puntuación en BLEU y ROUGE-L, pero es superado en CIDEr. En Viajes, aunque Florence gana claramente en BLEU, METEOR y CIDEr, ambos modelos empatan en ROUGE-L. Esta variabilidad refuerza la idea de que las métricas cuantitativas clásicas no siempre capturan adecuadamente la calidad contextual o semántica de una descripción generada, y que su interpretación debe hacerse con precaución.

En cualquier caso, se propone esperar al análisis posterior, basado en el juicio de un LLM, que servirá como evaluador semántico más afinado.

Evaluación de descripción de vídeo

En el caso de los algoritmos aplicados a vídeo, no se han encontrado en la literatura métricas objetivas y estandarizadas para evaluar este tipo de tareas.

Se exploró la posibilidad de adaptar métricas de descripción de cuadros, como las empleadas en el epígrafe anterior, pero los resultados obtenidos fueron dispares e inconsistentes, en parte debido a la naturaleza multimodal y secuencial de la información contenida en los vídeos. Estos enfoques, más allá de ofrecer valores numéricos, no permitían distinguir de manera robusta entre descripciones verdaderamente acertadas y aquellas que simplemente coincidían en palabras clave o estructura superficial.

Ante esta situación, se optó por no tomar decisiones prematuras basadas en estas métricas dudosas y reservar el proceso de evaluación para una fase posterior basada en LLM.

4.2.3. Evaluación con *LLM-as-a-judge*

Como se ha indicado en anteriores ocasiones a lo largo de este proyecto, las métricas clásicas no pueden ser aplicadas y por ello, se recurrió a la filosofía *LLM-as-a-judge*, un programa evaluador automático que, de forma consistente, aplica el mismo criterio estructurado sobre todos los experimentos.

La estructura de evaluación ha sido uniforme en las tres tareas principales: VQA, descripción automática de cuadros y descripción automática de vídeo. En todos los casos, el evaluador recibe como entrada el cuadro o fragmento de vídeo correspondiente, junto con la respuesta generada por el modelo, y emite una puntuación sobre 10 tras un breve razonamiento. Cabe destacar que, en una primera versión, los LLM tendían a evaluar con excesiva benevolencia, otorgando sistemáticamente puntuaciones de 10, con razonamientos escuetos e incluso contradictorios. Esto hizo necesario cambiar y «afinar» las instrucciones dadas al LLM, hasta conseguir una evaluación lo suficientemente crítica, flexible y sensible a los matices de cada respuesta.

Con estos ajustes, el sistema logró distinguir adecuadamente entre respuestas vagas, completas, detalladas o redundantes, y sus puntuaciones comenzaron a reflejar de forma más razonada las diferencias cualitativas reales. Esta evolución del proceso evaluador fue clave para llegar a resultados sólidos y respaldar las elecciones finales. En la Figura 4.14 se muestra un ejemplo de salida de uno de estos algoritmos, es este caso en particular, una evaluación, realizada por Deepseek, de la generación de descripciones de cuadros, en la que compara a Florence (Modelo 1) y a LAVIS (Modelo 2). Se ha incluido y remarcado, además, al ganador global.


```
{
        "image_id": "10891.jpg",
        "evaluations": [
            {
                "model": "Modelo 1",
                "caption": "The image shows a woman standing in front of a display case
filled with a variety of vegetables, including broccoli. She is wearing a black dress
and yellow gloves, and there is a dustbin to her right. In the background, there are
other people standing, and the ceiling is illuminated with lights.",
                "evaluation": "Rating: 8.5/10. Explanation: The description is detailed
but could be more specific about the woman's attire and the exact type of vegetables
displayed. It also includes some context regarding the setting without being overly
descriptive.<|end_of_sentence|>",
                "score": 8.5
            },
            {
                "model": "Modelo 2",
                "caption": "a woman is looking at broccoli in a grocery store. a woman
is looking at broccoli in a grocery store. a woman is looking at broccoli in a grocery
store.",
                "evaluation": "Rating: 7.5/10. Explanation: The description accurately
captures the main subject (a woman) and setting (a grocery store), but lacks detail
about her actions or the specific type of broccoli being displayed. It also does not
mention any other relevant elements such as price tags or additional people present in
the scene.<|end_of_sentence|>",
                "score": 7.5
            }
        ],
        "winner": "Modelo 1"
    },

    {
        "average_scores": {
            "Modelo 1": 8.82,
            "Modelo 2": 7.55
        },
        "global_winner": "Modelo 1"
    }
```


Figura 4.14. Ejemplo de evaluación de la descripción de cuadros realizada por Deepseek

Evaluación de *Visual Question Answering*

La evaluación se llevó a cabo usando el modelo Deepseek-VL. Se presentaron a este modelo pares compuestos por el cuadro, la pregunta y la respuesta generada por cada sistema. En la Tabla 4.3 se tienen los resultados que arroja LLM en este caso.

ViLT obtiene una mejor puntuación media y muestra una menor desviación típica, lo que lo confirma como el algoritmo seleccionado dentro de los dedicados a VQA.

Tabla 4.3. Resultados de LLM para evaluación de algoritmos de VQA

Modelo	Puntuación	
	Media	σ
CLIP	4,5	3,41
ViLT	7	4,20

Evaluación de descripción de cuadros

En este caso también se utilizó Deepseek-VL, evaluando comparativamente las descripciones generadas por Florence y LAVIS para los mismos cuadros. Se pidió al evaluador que puntuara cada descripción del 0 al 10 y que, tras evaluar todos los ejemplos, indicara cuál de los dos modelos mostraba mayor calidad global. Los resultados son los recogidos en la Tabla 4.4.

Tabla 4.4. Resultados de LLM para evaluación de algoritmos de descripción de cuadros

Modelo	Puntuación de Flickr		Puntuación de Cocina		Puntuación de Viaje	
	Media	σ	Media	σ	Media	σ
Florence	8,82	0,22	8,4	0,30	7,35	0,36
LAVIS	7,55	0,47	7,6	0,30	7,1	0,49

A pesar de que LAVIS había obtenido en varias métricas cuantitativas (como BLEU o METEOR) puntuaciones más altas, el modelo evaluador Deepseek otorga consistentemente mejores calificaciones a las descripciones generadas por Florence. Este resultado resulta especialmente significativo en el caso del conjunto Flickr, donde la diferencia entre ambos modelos es notable (más de 1,2 puntos de media), con una desviación típica notablemente menor en Florence. Probablemente el hecho de que sea el conjunto con mayor número de imágenes y descripciones esté relacionado con ello.

Como se adelantó, la explicación de esta aparente contradicción parece estar ligada a la naturaleza de las métricas clásicas, que tienden a penalizar desviaciones del texto de referencia, aunque dichas desviaciones representen mejoras descriptivas o estilísticas. Florence genera descripciones más detalladas y ricas semánticamente, lo que es valorado positivamente por el evaluador basado en LLM. En cambio, LAVIS, aunque más conciso, parece quedar limitado en variedad y profundidad contextual.

En consecuencia, y pese a los resultados de las métricas automáticas, se considera que Florence demuestra una capacidad descriptiva más robusta y coherente con los objetivos del proyecto.

Evaluación de descripción de vídeo

En este caso, se recurrió al modelo Qwen-VL como evaluador LLM. A diferencia de las tareas anteriores, aquí se evaluaban fragmentos breves de vídeo junto con las descripciones generadas por dos modelos diferentes: LLaVA-Next Video y OpenBMB Video Captioning.

El proceso fue el mismo: para cada fragmento se solicitó al modelo evaluador que valorase la calidad de la descripción entre 0 y 10, comparando precisión, riqueza de detalle y adecuación contextual, calculándose a continuación la media de cada modelo. De nuevo, se recogen los resultados en la Tabla 4.5.

Tabla 4.5. Resultados de LLM para evaluación de algoritmos de descripción de vídeo

Modelo	**Puntuación**	
	Media	**σ**
LLaVA-Next-Video	8,35	0,51
OpenBMB	7,10	0,72

El evaluador otorgó puntuaciones más altas a las descripciones generadas por LLaVA-Next-Video, que fue el seleccionado como referencia dentro de la descripción de vídeos. Además, obtuvo una menor dispersión en las valoraciones, lo que refuerza su consistencia evaluativa.

Este resultado fue especialmente bien recibido, ya que en fases posteriores del proyecto se detectaron numerosos problemas de compatibilidad y ejecución con OpenBMB, que imposibilitaron realizar pruebas adicionales de manera estable.

Por tanto, se considera que LLaVA-Next-Video no solo ofrece mejores resultados en cuanto a calidad descriptiva, sino que, además, presenta una fiabilidad y accesibilidad claramente superiores. Esta elección también queda a la espera de su integración definitiva en el sistema, junto con el resto de modelos seleccionados.

4.2.4. Conclusión

Tras la evaluación llevada a cabo a través de métricas objetivas y modelos evaluadores LLM, se ha procedido a seleccionar los algoritmos que formarán parte del sistema final. En *Visual Question Answering*, el modelo ViLT ha demostrado un rendimiento consistentemente superior, tanto por métricas como por evaluación cualitativa. En descripción de cuadros, a pesar de que las métricas favorecen a LAVIS en algunos aspectos, la calidad de las descripciones de Florence ha sido avalada por el evaluador LLM con una ventaja clara. Finalmente, en descripción de vídeo, el modelo LLaVA-Next-Video no solo ha obtenido mejores puntuaciones, sino que ha mostrado una robustez técnica esencial para su despliegue práctico.

Una vez seleccionados estos modelos, en la Figura 4.15 se muestra la estructura del modelo cliente-servidor que engloba a los tres algoritmos, y que será con la que se trabaje a partir de ahora.

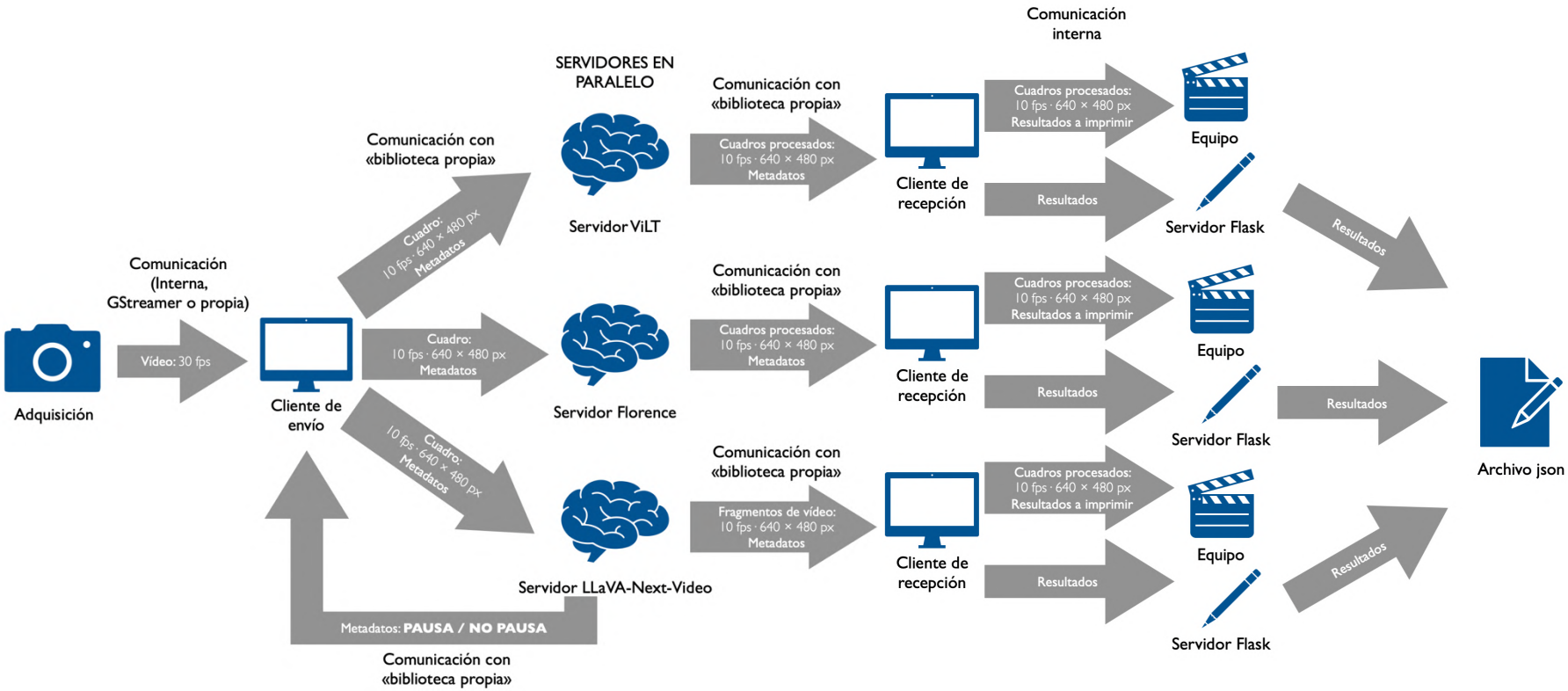


Figura 4.15. Esquema de la arquitectura cliente-servidor, con los tres modelos elegidos

4.3. Casos de uso

Tras haber construido una arquitectura robusta, perfectamente evaluada y validada tanto a nivel cualitativo como cuantitativo, es razonable hacerse la pregunta inevitable: ¿y ahora qué? Porque sí, se ha desarrollado un sistema complejo, con algoritmos de análisis de escenas que ofrecen descripciones extremadamente detalladas, con módulos de VQA que son capaces de razonar sobre el contenido visual, cuidadosamente afinados y puntuados por métricas tradicionales y por modelos de lenguaje evaluadores. Pero, ¿para qué sirve todo esto?

En última instancia, todo sistema inteligente debe tener una utilidad práctica. Y dado el marco en el que se ha desarrollado este proyecto, la robótica y la telepresencia inmersiva, resultaba especialmente importante aterrizar este desarrollo en usos concretos. No basta con demostrar que algo funciona, hay que demostrar también que sirve.

Inicialmente, se pensó en implementar un sistema sencillo en el propio cliente de recepción, que analizara las respuestas de los algoritmos buscando palabras clave como «fuego», «persona caída» o «arma», y activara alertas en caso necesario. Esta solución funcionaba bien en términos de rapidez y sencillez, pero resultaba excesivamente básica y, sobre todo, desaprovechaba por completo el inmenso caudal de información generado por los modelos de análisis.

Fue entonces cuando se recurrió a un enfoque más ambicioso y con más recorrido: aplicar el paradigma *sense-think-act* (sentir, pensar, actuar). Este enfoque, propio de la robótica, divide el comportamiento de un sistema inteligente en tres capas: percepción (*sense*), procesamiento (*think*) y actuación (*act*). Este modelo, ampliamente aceptado en el diseño de agentes cognitivos, no solo es aplicable a robots autónomos, sino que puede extrapolarse con éxito a múltiples contextos. En la Figura 4.16 se ha plasmado en un esquema la comunicación del sistema cliente-servidor con esta nueva capa de razonamiento y control, además de delimitar las capas correspondientes a *sense*, *think* y *act*.

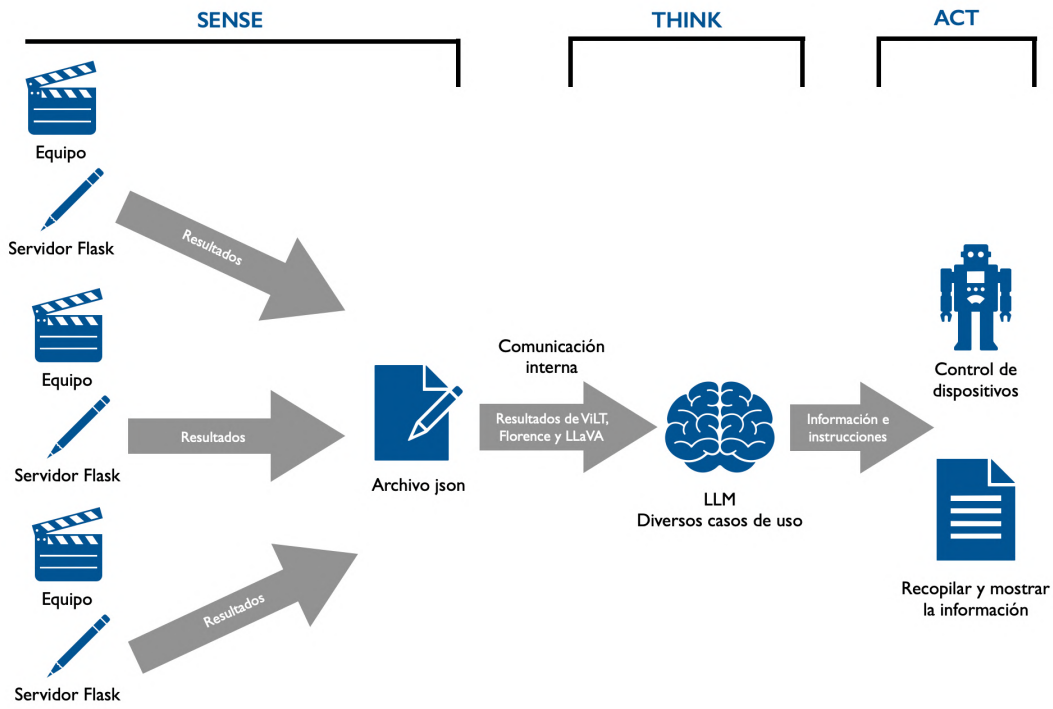


Figura 4.16. Esquema de la conexión de la capa de razonamiento con el sistema cliente-servidor

Así, en el presente proyecto, el bloque *sense* queda claramente representado por los algoritmos de análisis de escenas, que capturan y describen lo que ocurre en el entorno visual. La «magia» comienza en el *think*, que es donde se ha introducido una nueva capa de inteligencia artificial: un modelo de lenguaje de última generación que recibe como entrada los resultados de la percepción, los interpreta y extrae conclusiones. En teoría, debería ser una persona quien recibiera esta información y actuara en consecuencia. En la práctica, se delega ese razonamiento en una IA.

El modelo escogido para la capa de razonamiento ha sido Deepseek, por su rendimiento, rapidez, disponibilidad gratuita y porque ya se había utilizado en otros apartados del proyecto. Conviene señalar que, si bien ningún LLM «razona» en sentido estricto, Deepseek es capaz de simular razonamientos y análisis con una notable capacidad contextual. Se trata, por tanto, de una herramienta más que adecuada para extraer utilidad directa del contenido visual procesado.

Antes de explicar los diferentes casos de uso implementados, es conveniente revisar la arquitectura de esta parte del sistema. Los datos de entrada de este son, precisamente, los contenidos en el archivo json de salida en el que los diversos clientes de recepción escriben los resultados de los diversos algoritmos (véanse de nuevo las Figuras 4.14 y 4.15). En la versión definitiva, en que solo se emplean ViLT, Florence y LLaVA, dicho json tiene una estructura como la que se muestra en la Figura 4.17. Cabe mencionar, a modo de curiosidad, que no hay resultado de LLaVA para cada cuadro, sino únicamente al final del fragmento de vídeo analizado (en el ejemplo, el cuadro 24). En el *promt* que se introduce al LLM se le indica que ViLT solo arroja un término clave o dominante, que Florence genera una descripción detallada de la escena y que LLaVA añade contexto temporal.


```
      "frame": 23,
      "vilt": "phone",
      "florence": "The image shows a man sitting on a black office chair in an office
setting. He is holding a black tablet in his hands and appears to be playing a video game.
The man is wearing a white t-shirt, khaki pants, and red socks. He has a serious expression
on his face and is looking down at the tablet. On the right side of the image, there is a
woman sitting on the chair, wearing a green and white patterned dress. In the background,
there are other office chairs and a desk with a computer monitor. The floor is made of
concrete and there are windows on the left side.",
      "llava": null
    },
    {
      "frame": 24,
      "vilt": "phone",
      "florence": "The image shows two people sitting in an office setting. The person on
the left is sitting on a black office chair with wheels, while the person in the middle
is sitting in a green and white patterned chair. Both people are holding a black remote
control in their hands and appear to be playing a video game. In the background, there
are other office chairs and a desk with a computer monitor and other office supplies. The
floor is made of concrete and there is a large window on the right side of the image.",
      "llava": "In the video frames provided, we see a person standing in an office
environment. The individual is wearing a pair of dark pants and a pair of brown shoes.
They are holding a smartphone in their right hand, which they are using to interact with,
possibly playing a game or browsing the internet. The person's left hand is holding a
bottle of water, suggesting they are staying hydrated during their activity. The office
setting includes multiple chairs and desks, indicating a workspace or a meeting area. The
person appears to be engaged in a task on their phone, possibly multitasking between the
phone and the water bottle. The environment suggests a professional or work-related
setting."
    },
```


Figura 4.17. Ejemplo de salida del sistema cliente-servidor (capa *sense*)

Este programa se puede ejecutar en dos modos:

- En local, en que se leen los resultados de los modelos del archivo previamente guardado. Este modo está indicado especialmente para pruebas y depuraciones, especialmente a la hora de afinar la entrada de texto del LLM.
- En «vivo», en el que el programa va leyendo los datos del archivo conforme los clientes de recepción los van escribiendo en él, esto es, prácticamente a tiempo real (en la práctica hay un tiempo de retardo). Es el verdadero modo funcional del sistema, que permite la aplicación de los casos de uso que se introducen a continuación en tiempo real.

Tras esto, en función de la aplicación que se pretenda ejecutar, se habrá dado una entrada de texto determinada a Deepseek, que se encarga de leer el archivo con la información de cualquiera de los dos modos definidos antes, y de generar la respuesta pertinente, realizándose el *think* del sistema. En este punto, dado que el número total de escenas procesadas puede ser muy alto, especialmente en ejecución continua o cuando se trabaja con vídeos largos o robots en movimiento constante, se ha optado por dividir la entrada en bloques de tamaño fijo.

Esta decisión se debe, fundamentalmente, a las limitaciones de memoria y contexto del LLM, ya que los modelos como DeepSeek tienen una ventana de contexto limitada, por lo que no pueden analizar cientos de cuadros simultáneamente. Para evitar errores o respuestas truncadas, se agrupan en bloques pequeños. Otro motivo es el procesamiento en tiempo real o casi real, pues esta división por bloques permite procesar la información conforme va llegando, sin necesidad de esperar a que todo el vídeo o la sesión se complete.

El sistema almacena todos estos resultados, lo que permite su análisis posterior, o incluso construir una narrativa coherente. Además, en las aplicaciones en las que es necesario, se incluye un resumen final de los resultados de cada uno de estos bloques.

Una vez comprendida la estructura, se pasará a los diversos casos de uso que se han planteado.

4.3.1. Construcción de itinerarios y entornos en sistemas móviles

El primer caso de uso desarrollado se centra en la generación automática de un mapa o itinerario narrativo del entorno a partir de los datos visuales procesados por los algoritmos de análisis de escenas y el LLM. Esta utilidad se aplica directamente a escenarios en los que un robot móvil, equipado con cámaras, recorre un espacio y transmite continuamente cuadros o fragmentos de vídeo a la unidad de análisis.

El objetivo no es simplemente detectar objetos o describir escenas de forma aislada, sino permitir que un modelo de lenguaje como Deepseek mantenga un hilo conductor, identificando el recorrido completo del robot, los lugares atravesados, y los elementos relevantes encontrados en cada zona. Es decir, construir una traza del desplazamiento, con capacidad para identificar patrones, eventos clave o anomalías.

Este enfoque presenta aplicaciones reales y directas en diversos contextos:

- **Supervisión en tareas de inspección autónoma:** en una fábrica, almacén o laboratorio, el sistema puede registrar automáticamente por qué zonas ha pasado el robot, qué ha observado, y generar un informe textual que sustituye o complementa a los tradicionales informes técnicos. Esto reduce la necesidad de intervención humana y evita errores por omisión.

- **Apoyo a usuarios con movilidad reducida:** en contextos de telepresencia para personas con movilidad limitada, el sistema puede ofrecer un resumen continuo del entorno recorrido, indicando qué habitaciones se han visitado, qué personas o elementos se han detectado, si se ha pasado por una cocina, una sala de espera, etc.
- **Detección de desviaciones o eventos no previstos**: si el robot entra en una zona no autorizada, detecta personas donde no deberían estar, o se produce un cambio relevante respecto a anteriores recorridos (como una puerta abierta que siempre estaba cerrada), el sistema puede generar alertas en lenguaje natural entendibles por cualquier usuario.
- **Entrenamiento de sistemas de navegación cognitiva:** más allá del mapa físico, el itinerario que genera el modelo puede ser útil para desarrollar futuras capacidades de navegación basadas en comprensión contextual. Por ejemplo, el sistema puede inferir que ha salido de una zona segura para entrar en un área industrial peligrosa, o que ha pasado de una sala común a una habitación privada.

El funcionamiento es «sencillo», Deepseek, acumulando lasas descripciones recibidas, reconstruye el itinerario completo en formato narrativo, resaltando los lugares, objetos, personas y situaciones más destacadas, tal y como se ve en el ejemplo presentado en la Figura 4.18.

```
=== FINAL SUMMARY ===
1. Start in an unknown location, possibly in the middle of the building or inside
a workshop for the manufacturing facility. It interacts with various objects such
as computers, desks, and tables.

2. It moves further into the hallways, where it interacts with various
individuals. It often encounters individuals engaged in meetings, discussions,
or using computers.

3. It moves towards the meeting rooms, where it interacts with various tools and
equipment used in meetings. It often passes through the hallway to reach the
meeting rooms.

4. It moves towards the upstairs spaces, where it interacts with individuals in
the offices, observing people sitting on chairs and playing games.

5. Finally, the robot reaches the end and stops, possibly inspecting or
maintaining the robot itself.

The robot's journey suggests that the robot is exploring the environment,
observing people's activities, and learning about the building. It is guided by
the algorithms 'Vilt', 'Florence', and 'LLava' which interpret the surroundings
and identify the objects of interest. This journey is a complex one inside a
building, with the robot's movements and interactions being guided by its
understanding of the environment and interactions with objects and people.
```

Figura 4.18. Ejemplo de la salida del caso de uso de construcción de itinerario del robot

En definitiva, se trata de dar sentido al movimiento, convirtiendo una secuencia de cuadros en un registro comprensible, útil y automatizado, que además puede alimentar otras capas de decisión o control.

4.3.2. Control y navegación autónoma basada en percepción visual

En este segundo caso de uso se lleva un paso más allá la integración entre la percepción visual y la acción, dotando al sistema de una capacidad decisoria real basada en el análisis semántico del entorno.

A partir de las descripciones generadas por los algoritmos de análisis de escenas, se lanza una consulta al modelo Deepseek que, interpretando dicha información, propone una acción concreta que debería realizar el robot. Este modelo de funcionamiento se inspira directamente en el paradigma *Sense-Think-Act*. Se han contemplado dos modos de funcionamiento diferenciados:

- **Modo asistencial o de soporte**

  Aquí, el modelo no toma el control directo del robot, sino que emite recomendaciones en lenguaje natural. Por ejemplo:

  «Evite el área cercana a la estantería caída.»

  «La sala actual parece bloqueada, es recomendable regresar.»

  Este tipo de mensajes son ideales para contextos de teleoperación, donde una persona sigue recibiendo el control principal, pero se le ofrecen sugerencias útiles y contextualizadas en tiempo real. También es aplicable en entornos de colaboración humano-robot. Un ejemplo de esta salida es el que se muestra en la Figura 4.19.

```
- Scene: The robot is currently observing an indoor environment with a row of
black office chairs aligned in a neat arrangement. A large window in the
background allows natural light to reach the chairs, the overall atmosphere being
clean and professional.

- Observations: Based on the analysis, the robot can identify three major
elements: chairs (which includes the wheels), a person sitting on a chair, and a
desk with computer supplies. The person appears to be in an office setting and
is wearing traditional office attire. The background also indicates a small,
well-lit light source.

- Suggested action: The robot should proceed with caution. The person appears to
be walking towards or sitting down at the chairs, possibly due to a meeting or
communication. The robot should maintain a safe distance to avoid possible
contact. If the person is stationary and the chairs are occupied, the robot
should avoid the chairs and consider other pathways.
```

Figura 4.19. Ejemplo de la salida del caso de uso de control y navegación autónoma

- **Modo autónomo**

  En este caso, el modelo emite comandos estructurados (por ejemplo: avanzar, girar izquierda, retroceder) directamente integrables con sistemas como ROS u otros controladores de movimiento. Esto permite un funcionamiento completamente autónomo, en el que el robot decide por sí mismo hacia dónde dirigirse según lo que está viendo y comprendiendo.

Este sistema puede utilizarse en múltiples escenarios:

- Navegación autónoma en entornos complejos o desconocidos, donde no se dispone de mapas previos o se requiere una interpretación semántica de alto nivel.
- Asistencia a personas mayores o dependientes, guiando robots acompañantes o de reparto que deben evitar obstáculos o encontrar a una persona.
- Sistemas de patrullaje o seguridad, en los que se exige una reacción ante cambios no esperados (presencia humana, caída de objetos, bloqueo de rutas).

En resumen, esta utilidad convierte al sistema en una verdadera unidad de navegación autónoma o semiautónoma, con capacidad para interpretar y actuar sobre su entorno sin supervisión directa.

4.3.3. Generación de informes y resúmenes de actividades para sistemas de telepresencia

Este caso de uso traslada el poder del análisis semántico y la inteligencia generativa al ámbito de la telepresencia inmersiva, especialmente en situaciones donde una persona o un sistema desea mantenerse informado sobre lo que ha ocurrido durante un periodo de tiempo o en un entorno remoto.

La idea es sencilla, pero conveniente en este ámbito: tras observar una secuencia de vídeo, el modelo LLM es capaz de sintetizar en un breve informe qué ha ocurrido, recogiendo los aspectos más relevantes del análisis visual y convirtiéndolos en lenguaje natural comprensible. La utilidad de este tipo de resúmenes se manifiesta en diferentes contextos:

- **Actas automáticas de reuniones presenciales o virtuales:** en entornos laborales, educativos o técnicos, es posible generar un breve documento que indique:
  - Quiénes han estado presentes (si hay reconocimiento de personas).
  - Qué se ha hecho o tratado.
  - Cuáles han sido los cambios relevantes en el entorno.

  Esto facilita el seguimiento de proyectos, la documentación de tareas o el registro de incidencias en entornos monitorizados. Un ejemplo del resultado de esta aplicación es la de la Figura 4.20.

```
=== FINAL SUMMARY ===

The discussions were about a wide range of topics. There were many interchanges
of ideas, notes, and discussions on a notebook taken by one participant. Several
participants were also focusing their attention on a laptop screen showing a
presentation or projecting content.

The meeting ended with a cake serving, suggesting a dessert for dessert. The
group seemed to be in a positive, professional environment, with casual attire
and focus on the task at hand. This suggests the participants had a good working
or professional environment to engage in discussions.

To summarize, the meeting was productive with clear responsibilities and
expectations among participants, likely focused on an important project or
research task. Participants seemed to be engaged in meaningful discussions,
contributing to a positive professional environment.
```

Figura 4.20. Ejemplo de la salida del caso de uso de generación de informes y resúmenes

- **Supervisión asistida de pacientes o personas mayores:** aplicado al contexto médico o asistencial, el sistema podría generar breves notas indicando el comportamiento de la persona en cuestión, incluso detectando hechos anómalos, por ejemplo: «El paciente se ha mantenido en la cocina durante los últimos 30 minutos, pero no se ha sentado a comer, se recomienda visitarlo.»

  Este tipo de resúmenes automatizados, son de gran ayuda para cuidadores, familiares o personal médico, ya que permiten un seguimiento respetuoso, no invasivo y eficiente de la actividad cotidiana.

- **Análisis post-misión o auditorías visuales:** en aplicaciones industriales, de mantenimiento, vigilancia o inspección remota, se puede generar un informe que describa aspectos como qué

zonas han sido inspeccionadas, qué elementos han sido detectados o manipulados, y, de nuevo, si se han identificado posibles riesgos o anomalías.

Esto convierte el sistema en una especie de testigo digital, que no solo ve, sino que también comprende y deja constancia de lo que ha visto.

4.3.4. Asistente virtual y copiloto para la ejecución de tareas complejas

Una de las aplicaciones más transformadoras del sistema desarrollado es su capacidad para actuar como copiloto inteligente o asistente virtual durante la ejecución de tareas complejas que requieren seguimiento visual, análisis contextual y toma de decisiones en tiempo real. Este enfoque se basa en que el sistema comprenda, oriente e incluso evalúe al usuario mientras este realiza una actividad práctica.

Como ejemplo, se trató una clase de cocina. En fases iniciales se propuso una estrategia basada en la comparación directa entre dos vídeos: el del «profesor» y el del «alumno». Sin embargo, se descartó por tres motivos clave:

- La carga computacional excesiva, ya que analizar en paralelo dos flujos visuales implicaba una duplicación de procesos y memoria.
- La ambigüedad en las acciones, porque cuando ambos sujetos aparecían en la misma escena resultaba extremadamente difícil para el modelo discernir a quién pertenecía cada acción.
- La falta de sincronización, que por otra parte es inevitable, pero la ejecución del alumno rara vez se producía en tiempo real, lo que dificultaba la sincronización temporal de acciones.

Por tanto, se adoptó una estrategia mucho más robusta: usar como referencia un texto estructurado con los pasos correctos, y comparar la ejecución visual únicamente contra dicho estándar. En el caso de la clase de cocina es la «receta». Este enfoque permite una mayor escalabilidad (un mismo texto puede servir para múltiples usuarios), la posibilidad de una evaluación asincrónica (el sistema puede analizar los vídeos a posteriori) y amplía la flexibilidad para poder aplicarlo a otras tareas similares, como procedimientos médicos o protocolos industriales. Algunas aplicaciones son:

- **Cocina asistida por IA**

  Como se ha indicado, el sistema recibe como entrada el vídeo en tiempo real o pregrabado del usuario cocinando y una receta en texto estructurado, paso por paso.

  A partir de esta información, el LLM evalúa la ejecución, comparando lo que se ve con lo que se debería hacer, y genera comentarios como «Todas las instrucciones se han seguido» o «Has olvidado añadir el huevo», como puede verse en el ejemplo de la Figura 4.21.

```
=== STEP 3: Step 3: Place the egg whites into a clean mixing bowl ===
Step Evaluation:
- Is the step executed? Yes
- Observed actions: The chef is placing the egg whites into a clean mixing bowl.
- Missing or incorrect actions: None observed.
- Additional comments: The step is correctly executed. The egg whites are placed
into a clean mixing bowl.
```

Figura 4.21. Ejemplo de la salida del caso de uso de asistente y copiloto

Este enfoque genera un sistema de evaluación autónoma, que podría utilizarse tanto en contextos educativos (clases de cocina) como domésticos o incluso profesionales, permitiendo ahorrar recursos humanos y mejorar el aprendizaje autónomo, como se propone a continuación.

- **Generalización del caso: copiloto o asistente virtual**

  Más allá de la cocina, esta idea es extrapolable a cualquier proceso con pasos definidos: montaje de módulos, reparaciones, inspecciones visuales, formaciones técnicas, etc., abriendo la puerta a asistentes virtuales expertos en tiempo real.

  Se trata, en definitiva, de una forma innovadora de implementar un copiloto cognitivo que, gracias al análisis visual y a la referencia textual, actúa como un segundo par de ojos inteligente, capaz de emitir valoraciones, advertencias y sugerencias con una comprensión cercana al razonamiento humano.

Estos cuatro ejemplos demuestran la versatilidad de la arquitectura desarrollada. La combinación de sistemas de percepción visual y razonamiento natural permite transformar datos sin estructura en información organizada y operativa, acercando la robótica y la telepresencia a niveles de autonomía e inteligencia superiores.

Capítulo 5

CONCLUSIONES, LÍNEAS FUTURAS Y VALORACIÓN

5.1. Conclusiones generales

El desarrollo de este proyecto ha permitido validar una arquitectura funcional y robusta para el análisis de escenas mediante inteligencia artificial, orientada a su integración en contextos reales como la robótica móvil o la telepresencia inmersiva. A lo largo del trabajo se ha conseguido combinar tecnologías punteras en visión por computador, modelos multimodales y razonamiento generativo, en una solución completa capaz de interpretar el entorno, generar descripciones detalladas y actuar en consecuencia.

Más allá de la implementación técnica, el proyecto ha servido para explorar y cuestionar los límites actuales de los sistemas cognitivos artificiales, evaluando no solo su rendimiento en términos de precisión descriptiva, sino también su aplicabilidad práctica en casos de uso complejos y variados.

5.1.1. Funcionalidades implementadas con éxito

Entre las funcionalidades implementadas con éxito y que han cumplido satisfactoriamente con los objetivos iniciales, destacan:

- Estudio exhaustivo y fundamentado de distintos enfoques de análisis de escenas, incluyendo arquitecturas como ViLT, Florence o LLaVA, con especial atención a su comportamiento multimodal.
- Diseño e implementación de una arquitectura distribuida que permite la comunicación efectiva entre los clientes y los servidores de análisis, haciendo posible el procesamiento en tiempo real de las cuadros recibidas desde distintos entornos.
- Integración de un sistema de análisis de escenas con diferentes equipos de captación de imagen, entre los que destacan el robot zoomórfico Unitree Go2 y el dispositivo de telepresencia inmersiva Búho.
- Desarrollo de un módulo de razonamiento basado en LLM que transforma las descripciones visuales en acciones o interpretaciones más elevadas, aplicables en contextos como navegación robótica, generación de informes, evaluación de tareas, etc.

- Evaluación rigurosa del rendimiento de los modelos de análisis, combinando métricas tradicionales con *LLM-as-a-Judge*, lo que ha permitido una valoración más rica y contextual.
- Implementación de casos de uso reales y escalables, que prueban la utilidad del sistema en condiciones cercanas a escenarios operativos: navegación autónoma, recomendaciones asistidas, elaboración de resúmenes de actividad o seguimiento de tareas estructuradas.

En definitiva, el conjunto de funcionalidades implementadas en este proyecto supone un avance tangible hacia el fin último de muchos sistemas inteligentes: dotar de verdadera inteligencia operativa a las máquinas, permitiéndoles no solo percibir su entorno, sino también interpretarlo y actuar en consecuencia. La arquitectura construida, con su sistema modular y distribuido, demuestra que es posible integrar visión artificial avanzada con modelos de lenguaje para generar una capa de razonamiento autónomo, sin necesidad de supervisión constante.

Además, el sistema funciona con fiabilidad, es rápido en la generación de respuestas y lo suficientemente flexible como para adaptarse a distintos entornos o aplicaciones, tanto en robótica como en telepresencia. La combinación de modelos como ViLT, Florence y LLaVA, junto con un LLM final, produce una salida coherente, útil y contextualizada, que supera ampliamente el enfoque clásico de detección de objetos o simple generación de descripciones de cuadros o vídeos.

En resumen, el trabajo no se limita a validar piezas por separado, sino que construye un sistema funcional que realmente acerca a las máquinas a una comprensión más rica y práctica del entorno visual, abriendo la puerta a nuevas formas de autonomía, asistencia y toma de decisiones fundamentadas.

5.1.2. Limitaciones técnicas observadas

A pesar de los buenos resultados obtenidos, el trabajo se ha enfrentado a diversas limitaciones técnicas que han condicionado algunas decisiones y resultados:

- **Falta de madurez de la tecnología:** muchos de los modelos utilizados, especialmente en el ámbito del razonamiento multimodal, se encuentran aún en desarrollo activo, siendo evidente que es necesario que puedan aportar todavía «un poco más». Algunos, incluso, han presentado inconsistencias, escasa documentación o dificultades de integración, como ha ocurrido con OpenBMB.
- **Restricciones de *hardware*:** las pruebas se han llevado a cabo en entornos con GPU limitadas, lo que ha condicionado la carga de algunos modelos y ha forzado el uso de técnicas de cuantización y división por bloques. En ocasiones, se han producido errores por agotamiento de memoria.
- **Evaluación y *benchmarking*:** la falta de conjuntos de datos específicos para tareas como la descripción automática de cuadros o la evaluación razonada de escenas ha obligado a generar manualmente parte de los datos, o a adaptar recursos existentes que no siempre encajaban del todo.
- **Complejidad del flujo:** la arquitectura modular, aunque flexible, también introduce cierta complejidad técnica a la hora de sincronizar procesos, validar la coherencia de los datos transmitidos o mantener la integridad del flujo en tiempo real.
- **Limitaciones del razonamiento LLM:** aunque los modelos como Deepseek ofrecen una ilusión convincente de razonamiento, no dejan de ser generadores probabilísticos, lo que implica que sus respuestas pueden no ser siempre fiables, reproducibles o fácilmente verificables.

En resumen, estas limitaciones no restan valor al sistema propuesto, pero marcan claramente las áreas donde será necesario seguir trabajando, mejorar el soporte técnico o esperar avances de la comunidad científica para una adopción más amplia y estables.

5.2. Propuestas de mejora y líneas de futuro

A pesar del grado de desarrollo alcanzado, el sistema presentado ofrece múltiples vías de evolución y mejora. Algunas de estas líneas futuras están orientadas a consolidar el trabajo actual, mientras que otras apuntan a expandir significativamente su potencial en entornos reales más complejos y exigentes. A continuación, se detallan las principales:

- **Incorporación de nuevos modelos de análisis de escenas:** el sistema actual se apoya en una combinación sólida de modelos como ViLT, Florence, LLaVA y DeepSeek, pero el ritmo de evolución de la inteligencia artificial es vertiginoso. La inclusión de nuevos modelos más avanzados que surjan en el futuro puede enriquecer la comprensión visual, integrar mejor el contexto temporal, y acercarse aún más a una percepción profunda y estructurada.
- **Desarrollo de nuevos casos de uso con alto valor añadido:** aunque el sistema ya ha demostrado utilidad en escenarios como la navegación robótica, el seguimiento de tareas o la telepresencia guiada, su arquitectura permite generalizarlo a muchas más situaciones. Un caso especialmente prometedor es el del copiloto inteligente, que asista al usuario en tiempo real durante tareas complejas, adaptándose al contexto, corrigiendo errores y ofreciendo sugerencias proactivas. Este asistente podría evolucionar desde entornos estructurados (como recetas o mantenimiento industrial) hacia tareas abiertas, como apoyo en cirugía, docencia, o labores de campo.
- **Integración directa con sistemas de control como ROS:** la capacidad del sistema para generar órdenes de navegación es una base potente, pero su verdadero potencial se desbloquea al integrarse directamente con *middleware* como ROS Esta conexión permitiría que las recomendaciones generadas por el LLM se traduzcan automáticamente en comandos de bajo nivel para el robot, habilitando un control autónomo o semiautónomo basado en interpretación visual de alto nivel. Esto abriría la puerta a robots que «piensan y actúan» de forma reactiva, sin intervención humana constante.
- **Optimización del razonamiento LLM para entornos en tiempo real:** uno de los desafíos clave actuales es la latencia en el proceso de razonamiento. Aunque los modelos generan decisiones razonables, el tiempo necesario para procesar bloques de cuadros aún impide una verdadera actuación instantánea. Futuras mejoras deberían centrarse en reducir este tiempo, ya sea mediante modelos más ligeros, optimización por *hardware* o técnicas de razonamiento incremental continuo, en las que el LLM mantenga memoria activa de lo ya visto y solo procese nuevos elementos.
- **Incorporación de una memoria a largo plazo contextual:** en aplicaciones prolongadas, puede ser útil dotar al sistema de una memoria contextual, que retenga información sobre lo que ya ha ocurrido: habitaciones atravesadas, tareas realizadas, objetos detectados previamente, etc. Esto no solo permitiría generar una narrativa coherente del entorno, sino que podría alimentar procesos de toma de decisiones más ricos, históricos y personalizados.
- **Exploración de entornos colaborativos humano-robot:** finalmente, una dirección ambiciosa pero alineada con el espíritu del trabajo es la de diseñar interacciones colaborativas,

donde humano e IA compartan el proceso de análisis y decisión. Esto podría incluir, por ejemplo, asistentes en logística que guíen robots en almacenes, operarios que corrijan o completen inferencias del sistema, o plataformas educativas en las que el sistema evalúe automáticamente el progreso de un alumno y ofrezca realimentación adaptada.

5.3. Valoración personal del trabajo y análisis crítico

El desarrollo de este proyecto ha supuesto un desafío técnico y conceptual considerable, pero también una oportunidad enriquecedora para explorar a fondo el potencial actual de la inteligencia artificial en entornos robóticos y de telepresencia. Una de las principales satisfacciones ha sido haber conseguido integrar diversas herramientas, modelos y tecnologías en un sistema funcional y coherente, superando la fragmentación inicial que suponía trabajar con tantos enfoques distintos. La variedad de programas empleados, cada uno con su propia lógica, formato y nivel de especialización, exigía una arquitectura capaz de unificarlos sin perder fiabilidad ni utilidad real. En este sentido, lograr una solución que armoniza esta diversidad es, sin duda, uno de los logros clave del proyecto.

Además, ha resultado especialmente valioso el ejercicio de analizar cómo distintos algoritmos de visión y modelos de lenguaje pueden colaborar entre sí, con mejores o peores resultados, pero siempre desde un enfoque pragmático. Esta sinergia, más allá de las métricas individuales, ha permitido construir una percepción enriquecida del entorno, útil no solo para generar descripciones, sino también para alimentar procesos de decisión posteriores. En consecuencia, se ha conseguido ir más allá del tratamiento puramente teórico o descriptivo de la información visual, y se ha establecido una conexión real entre la percepción y la acción, entre lo que la máquina «ve» y lo que puede llegar a «hacer».

Otro aspecto destacable ha sido comprobar que el sistema, en la mayoría de los casos, se comporta con un grado de consistencia razonable, incluso cuando las entradas eran incompletas, ambiguas o demasiado técnicas. El hecho de que los modelos LLM pudieran interpretar correctamente los análisis de escenas y transformarlos en decisiones útiles o resúmenes comprensibles es una señal alentadora de la fiabilidad del enfoque seguido. Si bien no se trata de una inteligencia real, sí puede afirmarse que la integración alcanzada se acerca a lo que sería un comportamiento cognitivo simple, funcional y adaptable.

Ahora bien, no todo ha sido perfecto. Algunos elementos podrían haberse abordado con mayor profundidad o ambición. Por ejemplo, el sistema de evaluación automática de resultados podría haber sido más robusto y sistemático, incorporando comparaciones entre múltiples ejecuciones para medir consistencia. Del mismo modo, algunas decisiones arquitectónicas, especialmente en las primeras fases, podrían haberse tomado de forma más estructurada, lo que habría evitado ciertas redundancias o procesos poco eficientes. También quedó limitada, en parte, la capacidad del sistema para operar en tiempo real de forma completamente fluida, algo que futuras versiones deberían afrontar con mejoras tanto en *hardware* como en optimización de los modelos.

En conjunto, el trabajo realizado ha supuesto un paso relevante en la línea de aproximar las capacidades de la inteligencia artificial a su fin último: dotar o, al menos, ayudar a dotar de inteligencia práctica a las máquinas. Un objetivo aún lejano, pero en el que cada integración funcional, cada razonamiento automatizado con sentido, y cada pequeña decisión tomada a partir de lo que «se ve» representa un avance real.

APÉNDICE A:

ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE

El presente proyecto, centrado en el análisis de escenas mediante inteligencia artificial y su integración en entornos robóticos reales, encuentra una vinculación clara y justificada con varios de los Objetivos de Desarrollo Sostenible impulsados por las Naciones Unidas. Aunque se trata de una investigación de base tecnológica, su planteamiento, desarrollo y proyección futura se alinean con una visión más amplia de sostenibilidad, accesibilidad, eficiencia e innovación responsable.

ODS 9 – Industria, Innovación e Infraestructura

Este trabajo contribuye directamente al fomento de la innovación tecnológica y al desarrollo de infraestructuras digitales resilientes. La arquitectura distribuida cliente-servidor que conecta dispositivos físicos (como el robot zoomórfico) con servidores avanzados de análisis de cuadros y vídeo permite demostrar cómo es posible construir sistemas inteligentes adaptables y escalables. Asimismo, la evaluación de modelos mediante técnicas de *LLM-as-a-Judge* y el uso combinado de distintas arquitecturas multimodales representan un claro avance en la frontera tecnológica, promoviendo una innovación orientada a la utilidad práctica.

ODS 11 – Ciudades y Comunidades Sostenibles

Uno de los propósitos implícitos de este proyecto es facilitar la presencia y operación de sistemas autónomos en espacios compartidos, tanto urbanos como privados. El hecho de que el robot pueda comprender su entorno a través de descripciones visuales detalladas y generar decisiones de navegación en lenguaje natural, abre la puerta a aplicaciones en asistencia a personas, guiado inteligente, patrullaje o inspección de entornos complejos. Todo ello contribuye a hacer más seguras, inclusivas y eficientes las ciudades del futuro, especialmente en lo relativo a movilidad asistida y automatización responsable.

ODS 4 – Educación de Calidad

Desde una perspectiva académica y formativa, este proyecto supone una contribución directa a la educación técnica avanzada. No solo ha requerido la integración de conocimientos de visión por computador, IA, arquitectura de sistemas y robótica, sino que su desarrollo ha estado guiado por una filosofía de aprendizaje activo, resolución de problemas reales y exploración crítica de tecnologías emergentes. La documentación detallada, los experimentos realizados y las reflexiones extraídas pueden servir de base para futuras investigaciones, contribuyendo a la difusión de buenas prácticas en el uso ético y eficaz de la inteligencia artificial.

ODS 12 – Producción y Consumo Responsables

El proyecto promueve un uso optimizado de los recursos computacionales. Mediante el diseño por bloques y el envío inteligente de información visual, se minimiza el procesamiento redundante y se reduce la carga sobre los dispositivos. La reutilización de modelos preentrenados, el uso de dispositivos ya disponibles, y la atención al equilibrio entre coste computacional y rendimiento, se alinean con un enfoque de sostenibilidad técnica que evita el derroche energético y favorece una adopción más consciente y responsable de la IA en entornos reales.

ODS 17 – Alianzas para Lograr los Objetivos

Finalmente, este trabajo representa una forma concreta de colaboración técnica entre herramientas diversas, comunidades open source y servicios interconectados. La combinación de modelos distintos (Florence, ViLT, LLaVA, Deepseek), su evaluación comparada, y la integración con *frameworks* externos como ROS o servidores Flask, son ejemplo de cómo es posible construir soluciones más potentes y flexibles a partir de la cooperación entre agentes tecnológicos. Esta actitud de apertura y cooperación es clave para afrontar los desafíos complejos y transversales que plantean los ODS.

LISTA DE REFERENCIAS

[1] A. M. Turing, "Computing Machinery and Intelligence", *Mind*, vol. 59, no. 236, pp. 433-460, oct. 1950, doi: 10.1007/s11245-013-9182-y [Online]. Disponible en: https://phil415.pbworks.com/f/TuringComputing.pdf

[2] J. McCarthy, M. L. Minsky, N. Rochester y C. E. Shannon, "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955", en *AIMag*, vol. 27, no. 4, p. 12. [Online]. Disponible en: https://doi.org/10.1609/aimag.v27i4.1904

[3] J. Haugeland, *Artificial Intelligence: The Very Idea*, Cambridge, MA, EEUU: MIT Press, 1985.

[4] J. Weizenbaum, "ELIZA : a computer program for the study of natural language communication between man and machine", *Communications of the ACM*, vol. 9, no. 1, pp. 36-45, 1966 [Online]. Disponible en: https://dl.acm.org/doi/10.1145/365153.365168

[5] L. Floridi, "AI and Its New Winter: from Myths to Realities", Philos. Technol., vol. 33, pp. 1-3, 2020. [Online]. Disponible en: https://link.springer.com/article/10.1007/s13347-020-00396-6

[6] D. Silver et al., "Mastering the game of Go with deep neural networks and tree search", *Nature*, vol. 529, no. 7587, pp. 484-489, ene. 2016, doi: 10.1038/nature16961. [Online]. Disponible en: https://www.nature.com/articles/nature16961#citeas

[7] N. Ahmed, M. Wahed y N. C. Thompson, "The growing influence of industry in AI research", Science, vol. 379, no. 6635, pp. 884-886, mar. 2023, doi: 10.1126/science.ade2420. [Online]. Disponible en: https://www.science.org/doi/10.1126/science.ade2420

[8] "Estrategia de Inteligencia Artificial 2024", Ministerio para la Transformación Digital y de la Función Pública, Madrid, España, 2024. [Online]. Disponible en: https://portal.mineco.gob.es/es-es/digitalizacionIA/Documents/Estrategia_IA_2024.pdf

[9] D. Man y R. Olchawa, "The Possibilities of Using BCI Technology in Biomedical Engineering", en *Biomedical Engineering and Neuroscience,* W. P. Hunek y S. Paszkiel, Opole, Eds., Polonia: Springer, pp. 30-37. [Online]. Disponible en: https://www.researchgate.net/publication/322958198_The_Possibilities_of_Using_BCI_Technology_in_Biomedical_Engineering

[10] A. Khodaskar y S. Ladhake, "Semantic Image Analysis for Intelligent Image Retrieval", *Procedia Computer Science*, vol. 48, pp. 192-197, 2015, doi: 10.1016/j.procs.2015.04.169. [Online]. Disponible en: https://www.sciencedirect.com/journal/procedia-computer-science/vol/48/suppl/C

[11] "Ingeniería". Diccionario de la lengua española. Visitado: 11 mar. 2025. [Online]. Disponible en: https://dle.rae.es/ingenier%C3%ADa?m=form

[12] "What is Engineering?" IEEE. Visitado: 11 mar. 2025. [Online]. Disponible en: https://ewh.ieee.org/cmte/pa/UCF/Engineering.html

[13] E. Chang, Y. Lee, M. Billinghurst y B. Yoo, "Efficient VR-AR communication method using virtual replicas in XR remote collaboration", *International Journal of Human-Computer Studies*, vol.

190, n. 103304, oct. 2024, doi: 10.1016/j.ijhcs.2024.103304. [Online]. Disponible en: https://www.sciencedirect.com/science/article/pii/S1071581924000880

[14] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier y L. Van Gool, "Online Multiperson Tracking-by-Detection from a Single, Uncalibrated Camera", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1820-1833, sept. 2011, doi: 10.1109/TPAMI.2010.232. [Online]. Disponible en: https://ieeexplore.ieee.org/document/5674059

[15] M. Billinghurst, A. Clark y G. Lee. "A survey of augmented reality", *Foundations and Trends in Human-Computer Interaction*, vol. 8, pp. 73-272, ene. 2015, doi: 10.1561/1100000049. [Online]. Disponible en: https://www.researchgate.net/publication/277637661_A_Survey_of_Augmented_Reality

[16] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy y A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 1 abr. 2018, doi: 10.1109/TPAMI.2017.2699184. [Online]. Disponible en: https://ieeexplore.ieee.org/document/7913730

[17] L. Ferreira, L. Pereira, y R. Silva, "Immersive Mobile Telepresence Systems: A Systematic Literature Review", *Journal of Mobile Multimedia*, vol. 15, no.3, pp. 255-270, may. 2020, doi: 10.13052/jmm1550-4646.1535. [Online]. Disponible en: https://journals.riverpublishers.com/index.php/JMM/article/view/375/1939

[18] M. Utiel-Moreno et al., "Immersive Telepresence for Hybrid Meetings: Gamified Tele-assistance, Museums, and Technical Support", en *2024 16th International Conference on Quality of Multimedia Experience (QoMEX)*, Karlshamn, Suecia, 2024, pp. 175-178, doi: 10.1109/QoMEX61742.2024.10598299. [Online]. Disponible en: https://ieeexplore.ieee.org/document/10598299

[19] A. Jiménez-Moreno et al., "Evaluation of Segmentation Algorithms for Embodiment Improvement in an XR Application", Association for Computing Machinery, en *Proceedings of the 17th International Workshop on IMmersive Mixed and Virtual Environment Systems (MMVE '25)*, New York, NY, EEUU, 2025, pp. 36-39, doi: 10.1145/3712677.3720463. [Online]. Disponible en: https://dl.acm.org/doi/abs/10.1145/3712677.3720463

[20] J. V. Hurtado y A. Valada, "Semantic scene segmentation for robotics", en *Deep Learning for Robot Perception and Cognition*, A. Iosifidis y A. Tefas, Eds., Cambridge, MA, EEUU: Academic Press, 2022, cap. 12, pp. 279-311. [Online]. Disponible en: https://www.sciencedirect.com/science/article/abs/pii/B9780323857871000178

[21] "A Multi-Task Neural Architecture for On-Device Scene Analysis". Apple. Visitado: 23 may. 2025. [Online]. Disponible en: https://machinelearning.apple.com/research/on-device-scene-analysis

[22] C. Che, H. Zheng, Z.Huang, W. Jiang y B. Liu, "Intelligent Robotic Control System Based on Computer Vision Technology", en *Proceedings of the 6th International Conference on Computing and Data Science*, may. 2024, pp. 150-155, doi: 10.54254/2755-2721/64/20241373. [Online]. Disponible en: https://www.researchgate.net/publication/380597985_Intelligent_robotic_control_system_based_on_computer_vision_technology

[23] Y. LeCun, Y. Bengio y G. Hinton, "Deep learning", *Nature*, vol. 521, pp. 436-444, may. 2015, doi: 10.1038/nature14539. [Online]. Disponible en: https://www.nature.com/articles/nature14539

[24] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, D. Mané, "Concrete Problems in AI Safety", 2016, *arXiv:1606.06565*. [Online]. https://doi.org/10.48550/arXiv.1606.06565

[25] "Máster Universitario en Ingeniería Industrial + Máster Universitario en Administración de Empresas". Universidad Pontificia de Comillas. Visitado: 11 mar. 2025. [Online] . Disponible en: https://www.comillas.edu/postgrados/master-universitario-en-ingenieria-industrial-y-master-universitario-en-administracion-de-empresas-mba/

[26] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*. Cambrige, MA, EEUU: MIT Press, 2016. [Online]. Disponible en: https://www.deeplearningbook.org/

[27] M. Vega León, "Hagámoslo fácil: Deep Learning y Redes convolucionales", *Spain Business School Blog*, 13 oct. 2020. [Online]. Disponible en: https://blog.spainbs.com/2020/10/891/hagamoslo-facil-deep-learning-y-redes-convolucionales

[28] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", en *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, EEUU, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90. [Online]. Disponible en: https://ieeexplore.ieee.org/document/7780459

[29] A. Krizhevsky, I. Sutskever y G. E. Hinton, "ImageNet classification with deep convolutional neural networks", *Neural Information Processing Systems*, vol. 25, ene. 2012, doi: 10.1145/3065386. [Online]. Disponible en: https://www.researchgate.net/publication/267960550_ImageNet_Classification_with_Deep_Convolutional_Neural_Networks

[30] F. Doshi-Velez y B. Kim, "Towards A Rigorous Science of Interpretable Machine Learning", 2017, *arXiv:1702.08608*. [Online]. Disponible en: https://arxiv.org/abs/1702.08608

[31] Y. Lecun, L. Bottou, Y. Bengio y P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, nov. 1998, doi: 10.1109/5.726791. [Online]. Disponible en: https://ieeexplore.ieee.org/document/726791

[32] V. Dumoulin y F. Visin, "A guide to convolution arithmetic for deep learning", 2016, *arXiv:1603.07285*. [Online]. Disponible en: https://arxiv.org/abs/1603.07285

[33] "Building a Convolutional Neural Network", *The Click Reader*. [Online]. Disponible en: https://www.theclickreader.com/building-a-convolutional-neural-network/

[34] W. Rawat y Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review", Neural Computation, vol. 29, pp. 2352-2449, sept. 2017, doi: 10.1162/NECO_a_00990. [Online]. Disponible en: https://www.researchgate.net/publication/317496930_Deep_Convolutional_Neural_

[35] A. Vaswani et al., "Attention Is All You Need", en *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, EEUU, 2017. [Online]. Disponible en: https://arxiv.org/pdf/1706.03762

[36] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale", presentado en ICLR 2021, The Ninth International Conference on Learning

Representations, Viena, Austria, 3-7 may. 2021. [Online]. Disponible en: https://openreview.net/pdf?id=YicbFdNTTy

[37] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles y J. Herve, "Training data-efficient image transformers & distillation through attention", en *Proceedings of the 38th International Conference on Machine Learning*, 18-24 jul. 2021, pp. 10347-10357. [Online]. Disponible en: https://proceedings.mlr.press/v139/touvron21a/touvron21a.pdf

[38] A. Radford et al., "Learning Transferable Visual Models From Natural Language Supervision", en *Proceedings of the 38th International Conference on Machine Learning*, 18-24 jul. 2021. [Online]. Disponible en: https://proceedings.mlr.press/v139/radford21a/radford21a.pdf

[39] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database", en *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Miami, FL, EEUU, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848. [Online]. Disponible en: https://ieeexplore.ieee.org/document/5206848

[40] T. Y. Lin et al., "Microsoft COCO: Common Objects in Context", en *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele y T. Tuytelaars, Eds., Berlín, Alemania: Springer, 2014, pp. 740-755. [Online]. Disponible en: https://link.springer.com/chapter/10.1007/978-3-319-10602-1_48#citeas

[41] K. He, H. Fan, Y. Wu, S. Xie y R. Girshick, "Momentum Contrast for Unsupervised Visual Representation Learning", en *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, EEUU, 2020, pp. 9729-9738, doi: 10.1109/CVPR42600.2020.00975. [Online]. Disponible en: https://ieeexplore.ieee.org/document/9157636

[42] R. Bommasani et al., "On the Opportunities and Risks of Foundation Models", 2021, *arXiv:2108.07258*. [Online]. Disponible en: https://arxiv.org/pdf/2108.07258

[43] J. Li, D. Li, C. Xiong y S. Hoi, "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation", en *Proceedings of the 39th International Conference on Machine Learning*, Baltimore, MD, EEUU, 17-23 jul., 2022. [Online]. Disponible en: https://arxiv.org/abs/2201.12086

[44] M. Savva et al., "Habitat: A Platform for Embodied AI Research", 2019, *arXiv:1904.0120*. [Online]. Disponible en: https://arxiv.org/pdf/1904.01201

[45] T. Brown et al., "Language Models are Few-Shot Learners", en *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan y H. Lin, Eds., 6-12 dec. 2020, pp. 1877-1901. [Online]. Disponible en: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

[46] A. Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways", *Journal of Machine Learning Research*, vol. 24, pp. 1-113, 2023, doi: 10.48550/arXiv.2204.02311. [Online]. Disponible en: https://arxiv.org/pdf/2204.02311

[47] H. Touvron et al., "LLaMA 2: Open Foundation and Fine-Tuned Chat Models", 2023, *arXiv:2307.09288*. [Online]. Disponible en: https://arxiv.org/pdf/2307.09288

[48] S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models", 2023, *arXiv:2210.03629*. [Online]. Disponible en: https://arxiv.org/pdf/2210.03629

[49] J. B. Alayrac et al., "Flamingo: a visual language model for few-shot learning", en *Proceedings of the 36th International Conference on Neural Information Processing Systems (NIPS '22)*, Red Hook, NY, EEUU, 2022, Art.1723, pp. 23716-23736. [Online]. Disponible en: https://dl.acm.org/doi/10.5555/3600270.3601993

[50] J. Kaplan et al., "Scaling Laws for Neural Language Models", 2020, *arXiv:2001.08361*. [Online]. Disponible en: https://arxiv.org/pdf/2001.08361

[51] L. Yuan et al. "Florence: A New Foundation Model for Computer Vision", *arXiv:2111.11432*. [Online]. Disponible en: https://arxiv.org/pdf/2111.11432

[52] N. P. Jouppi et al., "In-Datacenter Performance Analysis of a Tensor Processing Unit", en *The 44th International Symposium on Computer Architecture (ISCA '17)*, Toronro, ON, Canadá, 24-28 jun. 2017, pp. 1-12, doi: 10.1145/3079856.3080246. [Online]. Disponible en: https://dl.acm.org/doi/pdf/10.1145/3079856.3080246

[53] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", en *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, EEUU, 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91. [Online]. Disponible en: https://ieeexplore.ieee.org/document/7780460

[54] K. He, G. Gkioxari, P. Dollár y R. Girshick, "Mask R-CNN", en *2017 IEEE International Conference on Computer Vision (ICCV)*, Venecia, Italia, 22-29 oct. 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322. [Online]. Disponible en: https://openaccess.thecvf.com/content_ICCV_2017/papers/He_Mask_R-CNN_ICCV_2017_paper.pdf

[55] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff y H. Adam, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation", en *Computer Vision - ECCV 2018. Lecture Notes in Computer Science*, V. Ferrari, M. Hebert, C. Sminchisescu y Y. Weiss, Eds., Munich, Alemania, 8-14 sept. 2018, pp. 833-851, doi: 10.1007/978-3-030-01234-2_49. [Online]. Disponible en: https://link.springer.com/chapter/10.1007/978-3-030-01234-2_49

[56] W. Kim, B. Son y I. Kim, "ViLT: Vision-and-Language Transformer Without Convolution or Region Supervision", en *Proceedings of the 38th International Conference on Machine Learning*, 18-24 jul. 2021, doi: 10.48550/arXiv.2102.03334. [Online]. Disponible en: https://arxiv.org/pdf/2102.03334

[57] D. Li et al., "LAVIS: A One-stop Library for Language-Vision Intelligence", en Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), D. Bollegala, R. Huang y A. Ritter, Eds., Toronto, Canadá, jul. 2023, pp. 31-41, doi: 10.18653/v1/2023.acl-demo.3. [Online]. Disponible en: https://aclanthology.org/2023.acl-demo.3.pdf

[58] F. Li et al., "LLaVA-NeXT-Interleave: Tackling Multi-image, Video, and 3D in Large Multimodal Models", 2024, *arXiv.2407.07895*.

[59] OpenBMB, "OpenBMB: Big Models for Everyone", *Medium*, 18 nov. 2022. Visitado: 16 jun. 2025. [Online]. Disponible en: https://medium.com/@openbmb/openbmb-big-models-for-everyone-f5a812c4ad43

[60] O. González-Chávez, G. Ruiz, D. Moctezuma y T. Ramirez-del Real, "Are metrics measuring what they should? An evaluation of Image Captioning task metrics", *Signal Processing: Image*

Communication, vol. 120, ene. 2024, Art. no. 117071, doi: 10.1016/j.image.2023.117071. [Online]. Disponible en: https://arxiv.org/pdf/2207.01733

[61] H. Sharma y A. S. Jalal, "A survey of methods, datasets and evaluation metrics for visual question answering", *Image and Vision Computing*, vol. 116, dec. 2021, Art. no. 104327, doi: 10.1016/j.imavis.2021.104327. [Online]. Disponible en: https://doi.org/10.1016/j.imavis.2021.104327

[62] D. Vernon, G. Metta y G. Sandini, "A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents", *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 2, pp 151-180, 2007, doi: 10.1109/TEVC.2006.890274. [Online]. Disponible en: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9e14bdc1e47f5a71d6e2b3c92a7501b83a0a658e

[63] R. C. Arkin, *Behavior-Based Robotics*. Cambridge, MA, EEUU: MIT Press, 1998. [Online]. Disponible en: http://www.coep.ufrj.br/~ramon/COE-841/robotics/book%201998%20-%20Behavior-Based%20Robotics%20-%20Arkin.pdf

[64] J. E. Laird, "Requirements for Cognitive Architectures", en *The Soar Cognitive Architecture*, Cambridge, MA, EEUU: MIT Press, 2012, cap. 2, doi: 10.7551/mitpress/7688.003.0004. [Online]. Disponible en: https://doi.org/10.7551/mitpress/7688.003.0004

[65] D. Vernon, *Artificial cognitive systems: A primer*, Cambrige, MA, EEUU: MIT Press, 2014. [Online]. Disponible en: http://vernon.eu/publications/14_Vernon_Artificial_Cognitive_Systems_Preamble.pdf

[66] R. Krishnaet al., "Visual Genome: Connecting language and vision using crowdsourced dense image annotations", *International Journal of Computer Vision*, vol. 123, no.1, pp. 32-73, 2017, doi: 10.1007/s11263-016-0981-7. [Online]. Disponible en: https://link.springer.com/article/10.1007/s11263-016-0981-7

[67] B. Kehoe, S. Patil, P. Abbeel y K. Goldberg, "A Survey of Research on Cloud Robotics and Automation", *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 398-409, Abr. 2015, doi: 10.1109/TASE.2014.2376492. [Online]. Disponible en: https://goldberg.berkeley.edu/pubs/T-ASE-Cloud-RA-Survey-Paper-Final-2015.pdf

[68] J. Wang et al., "Large language models for robotics: Opportunities, challenges, and perspectives", *Journal of Automation and Intelligence*, vol. 4, no. 1, pp. 52-64, 2025, doi: 10.1016/j.jai.2024.12.003. [Online]. Disponible en: https://www.sciencedirect.com/science/article/pii/S2949855424000613?via%3Dihub

[69] W. Chen, S. Hu, R. Talak y L. Carlone, "Leveraging Large (Visual) Language Models for Robot 3D Scene Understanding", 2022, *arXiv:2209.05629*. [Online]. Disponible en: https://arxiv.org/pdf/2209.05629

[70] M. Afrin, J. Jin, A. Rahman, A. Rahman, J. Wan y E. Hossain, "Resource Allocation and Service Provisioning in Multi-Agent Cloud Robotics: A Comprehensive Survey", *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 842-870, 2021, doi: 10.1109/COMST.2021.3061435. [Online]. Disponible en: https://arxiv.org/pdf/2104.14270

[71] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez y T. Funkhouser, "Learning Synergies Between Pushing and Grasping with Self-Supervised Deep Reinforcement Learning" en *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, España, 2018, pp. 4238-4245,

doi: 10.1109/IROS.2018.8593986. [Online]. Disponible en: https://arxiv.org/pdf/1803.09956

[72] A. Steed, y M. Oliveira, *Networked Graphics: Building Networked Games and Virtual Environments*, EEUU: Morgan Kaufmann, 2009.

[73] C. Flavián, S. Ibáñez-Sánchez y C. Orús, "The impact of virtual, augmented and mixed reality technologies on the customer experience", *Journal of Business Research*, vol. 100, pp. 502-510, jul. 2019, doi: 10.1016/j.jbusres.2018.10.050. [Online]. Disponible en: https://zaguan.unizar.es/record/84331/files/texto_completo.pdf

[74] T. Baltrušaitis, C. Ahuja and L. P. Morency, "Multimodal Machine Learning: A Survey and Taxonomy", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423-443, 1 feb. 2019, doi: 10.1109/TPAMI.2018.2798607. [Online]. Disponible en: https://people.ict.usc.edu/~gratch/CSCI534/Readings/Baltrusaitis-MMML-survey.pdf

[75] B. Mittelstadt, P. Allo, M. Taddeo, S. Wachter y L. Floridi, "The ethics of algorithms: Mapping the debate", *Big Data & Society*, vol. 3, no. 2, feb. 2017, doi: 10.1177/2053951716679679. [Online]. Disponible en: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2909885

[76] A. Munteanu. "Top 5 reasons to use Ubuntu for your AI/ML projects". Ubuntu, jun. 2024. Visitado: 16 jun. 2025. [Online]. Disponible en: https://ubuntu.com/blog/ubuntu-ai-ml-projects

[77] M. Iyam, "The Importance of Python in Artificial Intelligence", *Medium*, 23 oct. 2024. Visitado: 16 jun. 2025. [Online]. Disponible en: https://michael-lyamm.medium.com/the-importance-of-python-in-artificial-intelligence-341c7af1fb94

[78] "venv — Creation of virtual environments". Python documentation. Visitado: 16 jun. 2025. [Online]. Disponible en: https://docs.python.org/3/library/venv.html#module-venv

[79] "Managing environments". Conda documentation. Visitado: 16 jun. 2025. [Online]. Disponible en: https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

[80] "What is Docker?". Docker documentation. Visitado: 16 jun. 2025. [Online]. Disponible en: https://docs.docker.com/get-started/overview/

[81] A. Paszke et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library", en *Advances in Neural Information Processing Systems 32 (NeurIPS 2019)*, H. Wallach et al., Eds., Vancouver, Canadá, dic. 2019, doi: 10.48550/arXiv.1912.01703. [Online]. Disponible en: https://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[82] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing" en *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, oct. 2020, pp. 38-45, doi: 10.18653/v1/2020.emnlp-demos.6. [Online]. Disponible en: https://aclanthology.org/2020.emnlp-demos.6/

[83] G. Bradski, "The OpenCV Library", *Dr. Dobb's Journal of Software Tools*, vol. 120, 2000, pp. 122-125.

[84] P. Kumari, "Florence-2: Vision Model Shaping the Future of AI Understanding", *Labeller*, 21 nov. 2023. [Online]. Disponible en: https://www.labellerr.com/blog/florence-2-vision-model-by-microsoft/

[85] H. Lu et al., "DeepSeek-VL: Towards Real-World Vision-Language Understanding", 2024, *arXiv:2403.05525v2*.

[86] J. Bai et al., "Qwen-VL: A Multimodal Foundation Model with Video Understanding Capabilities", 2023, *arXiv:2308.12966*.

[87] "About Go2". Unitree Documentation Center. Visitado: 16 jun. 2025. [Online]. Disponible en: https://support.unitree.com/home/en/developer

[88] R. Kachach et al., "The Owl: Immersive Telepresence Communication for Hybrid Conferences", en *2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, Bari, Italia, 2021, pp. 451-452, doi: 10.1109/ISMAR-Adjunct54149.2021.00104. [Online]. Disponible en: https://ieeexplore.ieee.org/document/9585789

[89] R. Kachach et al., "A Multi-Peer, Low Cost Immersive Communication System for Pandemic Times", en *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, Lisboa, Portugal, 2021, pp. 691-692, doi: 10.1109/VRW52623.2021.00228. [Online]. Disponible en: https://ieeexplore.ieee.org/document/9419314

[90] M. Orduna, P. Pérez, J. Gutiérrez y N. García, "Methodology to Assess Quality, Presence, Empathy, Attitude, and Attention in 360-degree Videos for Immersive Communications", *IEEE Transactions on Affective Computing*, vol. 14, no. 3, pp. 2375-2388, 1 July-Sept. 2023, doi: 10.1109/TAFFC.2022.3149162. [Online]. Disponible en: 3.2.https://ieeexplore.ieee.org/abstract/document/9713751

[91] "GStreamer. Open source multimedia framework". GStreamer. Visitado: 3 jul. 2025. [Online]. Disponible en: https://gstreamer.freedesktop.org/features/