

UNIVERSIDAD DE CASTILLA-LA MANCHA

GRADO EN INGENIERÍA INFORMÁTICA

Ingeniería de Computadores

TRABAJO FIN DE GRADO

Sensorización y comunicación inalámbrica de objetos
manipulables en entornos de rehabilitación inmersivos
basados en Realidad Aumentada

Rafael Echevarría Aceña

Escuela
Superior
de Informática



Ciudad Real, julio de 2026



**UNIVERSIDAD DE CASTILLA-LA MANCHA
ESCUELA SUPERIOR DE INFORMÁTICA**

Tecnologías y sistemas de información

Ingeniería de Computadores

TRABAJO FIN DE GRADO

**Sensorización y comunicación inalámbrica de objetos
manipulables en entornos de rehabilitación inmersivos
basados en Realidad Aumentada**

Tutor/a: Javier Alonso Albusac Jiménez

julio, 2026

Sensorización y comunicación inalámbrica de objetos manipulables en entornos de rehabilitación inmersivos basados en Realidad Aumentada

© Rafael Echevarría Aceña, 2026

Este documento se distribuye con licencia CC BY-NC-SA 4.0. El texto completo de la licencia se puede obtener en <https://creativecommons.org/licenses/by-nc-sa/4.0/>. La copia y distribución de esta obra está permitida en todo el mundo, sin regalías y por cualquier medio, siempre que esta nota sea preservada. Se concede permiso para copiar y distribuir traducciones de este libro desde el español original a otro idioma, siempre que la traducción sea aprobada por el autor del libro y tanto el aviso de copyright como esta nota de permiso sean preservados en todas las copias.



Este texto ha sido preparado con la plantilla L^AT_EX para Trabajo Fin de Estudios en Ingeniería Informática para la UCLM publicada por [Jesús Salido](#) en el repositorio público Zenodo, DOI: [10.5281/zenodo.4561708](https://doi.org/10.5281/zenodo.4561708).

A los pacientes del Hospital Nacional de Paraplégicos, cuya realidad me ha enseñado que el fin último de la investigación y la tecnología debe ser, siempre, mejorar la vida de las personas.

Sensorización y comunicación inalámbrica de objetos manipulables en entornos de rehabilitación inmersivos basados en Realidad Aumentada

Rafael Echevarría Aceña
Ciudad Real, julio 2026

Resumen

La realidad mixta se ha convertido en una herramienta prometedora para la rehabilitación del miembro superior, al permitir combinar ejercicios terapéuticos, interacción inmersiva y objetos físicos manipulables. Sin embargo, los sistemas de seguimiento óptico integrados en visores comerciales, como Meta Quest 3, presentan una limitación crítica durante la interacción mano-objeto: cuando el paciente agarra el objeto, la mano puede ocultarlo parcial o totalmente, provocando pérdidas de seguimiento que afectan a la coherencia espacial y a la continuidad de la tarea.

Este Trabajo Fin de Grado propone el diseño y validación de una arquitectura de objetos inteligentes resistente a la oclusión, orientada a escenarios de rehabilitación con realidad mixta. El sistema integra un objeto físico instrumentado mediante un Seeed Studio XIAO nRF52840 Sense, una unidad de medición inercial de 6 ejes y sensores de presión FSR, de forma que el propio objeto aporta información sobre su movimiento y estado de agarre. Estos datos se transmiten mediante Bluetooth Low Energy y se fusionan con la detección visual basada en YOLOv8-Pose para mantener una estimación continua de la pose del cubo incluso cuando la visión falla.

La solución se ha integrado en un juego serio inspirado en el Box and Block Test y se ha evaluado mediante pruebas técnicas y pruebas preliminares con usuarios en el Hospital Nacional de Paraplégicos de Toledo. Los resultados muestran que la arquitectura mejora la continuidad de la interacción tangible, reduce los efectos de la oclusión visual y permite registrar datos cinemáticos y de interacción útiles para el análisis posterior de la sesión. Además, las pruebas con usuarios han permitido identificar limitaciones relevantes en condiciones clínicas reales, especialmente relacionadas con la profundidad del cubo durante agarres complejos como la pinza por tenodesis.

Palabras clave: realidad mixta, rehabilitación del miembro superior, objetos inteligentes, interacción tangible, fusión sensorial, seguimiento resistente a la oclusión, sensores inerciales, sensores de presión.

Sensor Integration and Wireless Communication for Manipulable Objects in Immersive Rehabilitation Environments Based on Augmented Reality

Rafael Echevarría Aceña
Ciudad Real, julio 2026

Abstract

Mixed reality has become a promising tool for upper-limb rehabilitation, as it enables the combination of therapeutic exercises, immersive interaction and manipulable physical objects. However, the optical tracking systems integrated into commercial headsets, such as Meta Quest 3, present a critical limitation during hand-object interaction: when the patient grasps the object, the hand may partially or completely occlude it, causing tracking losses that affect spatial coherence and task continuity.

This Bachelor Thesis proposes the design and validation of an occlusion-resilient smart-object architecture for mixed-reality rehabilitation scenarios. The system integrates a physical object instrumented with a Sseed Studio XIAO nRF52840 Sense, a 6-axis inertial measurement unit and FSR pressure sensors, so that the object itself provides information about its motion and grasping state. These data are transmitted via Bluetooth Low Energy and fused with visual detection based on YOLOv8-Pose to maintain a continuous estimation of the cube pose even when vision fails.

The solution has been integrated into a serious game inspired by the Box and Block Test and evaluated through technical tests and preliminary user trials at the Hospital Nacional de Paraplégicos de Toledo. The results show that the architecture improves the continuity of tangible interaction, reduces the effects of visual occlusion and enables the recording of kinematic and interaction data that are useful for subsequent session analysis. In addition, the user trials made it possible to identify relevant limitations under real clinical conditions, especially those related to cube depth during complex grasping strategies such as tenodesis grip.

Keywords: mixed reality, upper-limb rehabilitation, smart objects, tangible interaction, sensor fusion, occlusion-resilient tracking, inertial sensors, pressure sensors.

Agradecimientos

A mis padres, por darme una gran educación y por ser mi ejemplo de trabajo y dedicación. A mi hermana Irene, por motivarme a dar siempre lo mejor de mí. A Nuria, por estar a mi lado desde el principio, apoyándome en los momentos de frustración. Y, por supuesto, a mis amigos y compañeros, por compartir las largas noches de trabajo y hacer este recorrido mucho más ligero.

Quiero agradecer también a mi tutor su confianza para desarrollar este trabajo y su guía constante; su calidad docente y su apoyo han sido fundamentales para este proyecto. Asimismo, agradezco al grupo AIR su amabilidad y la oportunidad de conocer de cerca la labor que desempeñan.

Rafael Echevarría Aceña
Ciudad Real, 2026

Notación y acrónimos

NOTACIÓN

No se emplea una notación matemática específica que requiera una tabla independiente. Los símbolos y magnitudes técnicas utilizados en la memoria se definen en el propio texto cuando aparecen por primera vez.

LISTA DE ACRÓNIMOS

BBT	: Box and Block Test
BLE	: Bluetooth Low Energy
FSR	: Force Sensing Resistor
HNP	: Hospital Nacional de Paraplégicos
IA	: Inteligencia Artificial
IMU	: Inertial Measurement Unit / Unidad de Medición Inercial
IoT	: Internet of Things / Internet de las Cosas
RA	: Realidad Aumentada
RM	: Realidad Mixta
RTT	: Round-Trip Time
RV	: Realidad Virtual
SUS	: System Usability Scale
TFG	: Trabajo Fin de Grado
XR	: Extended Reality / Realidad Extendida
YOLO	: You Only Look Once

Índice general

Resumen	vii
Abstract	ix
Agradecimientos	xi
Notación y acrónimos	xiii
Índice de figuras	xvii
Índice de tablas	xix
Índice de listados	xxi
Índice de algoritmos	xxiii
1 Introducción	1
1.1 Contexto disciplinar y tecnológico	2
1.2 Justificación del proyecto	3
1.3 Estructura del documento	4
2 Objetivos	5
2.1 Objetivo general	5
2.2 Objetivos específicos	5
3 Estado del arte	7
3.1 Realidad Extendida en la rehabilitación del miembro superior	7
3.2 Limitaciones de los sistemas de interacción en XR para rehabilitación . . .	10
3.3 Wearables y sensorización del paciente en rehabilitación	11
3.4 Objetos inteligentes y entornos instrumentados en rehabilitación	14
3.5 Adaptación del espacio terapéutico a las capacidades del paciente	16
3.6 Síntesis del estado del arte y posicionamiento del trabajo	18
4 Metodología	21
4.1 Metodología elegida	21
4.2 Distribución del trabajo	21
4.3 Recursos empleados	28
4.4 Estimación de costes	29

5	Arquitectura del sistema	31
5.1	Retos y problemáticas	31
5.2	Visión general	32
5.3	Requisitos clínicos	34
5.4	Pipeline de datos extremo a extremo	34
5.5	Diseño de objetos inteligentes	35
5.6	Módulo de visión	44
5.7	Módulo de fusión	51
5.8	Módulo de juego serio	63
5.9	Módulo de registro de datos	65
6	Experimentación y resultados	67
6.1	Validación técnica del sistema	67
6.2	Pruebas con usuarios	70
7	Conclusiones	77
7.1	Conclusiones generales	77
7.2	Objetivos alcanzados	78
7.3	Justificación de competencias adquiridas	79
7.4	Limitaciones del sistema	79
7.5	Trabajo futuro	80
7.6	Uso de la Inteligencia Artificial	80
	Bibliografía	81
A	Recursos complementarios del proyecto	87
A.1	Repositorio de código fuente	87
A.2	Datos experimentales e informes derivados	87
A.3	Cuestionario de percepción subjetiva	88

Índice de figuras

1.1	Guante sensorizado del TFG de Mario Pérez	3
3.1	Diagrama de bosque (forest plot) que ilustra la diferencia de medias estandarizada (g de Hedges) de las intervenciones con realidad virtual sobre la función del miembro superior en pacientes con ictus crónico. El tamaño del efecto global (ES = 0,440) indica una mejora moderada y estadísticamente significativa en comparación con la rehabilitación convencional.	8
3.2	Diagrama de bosque (forest plot) de la escala Fugl-Meyer para el miembro superior (FMA-UE) categorizada por estadio de recuperación. Los resultados demuestran que la terapia VAMR (realidad virtual, aumentada y mixta) es significativamente superior al tratamiento estándar en la fase crónica (DM = 3,47; p = 0,03).	8
3.3	Análisis de meta-regresión de los parámetros de la intervención con RV. Arriba: correlación entre la duración de la intervención y la ganancia funcional, mostrando un aumento de la eficacia a medida que el tratamiento se acerca a las 8 semanas. Abajo: impacto de la frecuencia de las sesiones semanales en el tamaño del efecto.	9
3.4	Evolución tecnológica de los sistemas de rehabilitación del miembro superior en el HNP y el grupo AIR.	10
3.5	Sistemas de instrumentación del paciente mediante guantes en rehabilitación neurológica.	12
3.6	Sistema wBBT de Zhang et al. [37]: arquitectura, fusión de señales EMG e inerciales, y uso durante el Box and Block Test estándar.	13
3.7	Sistema de Postolache et al. [33]: arquitectura IoT para telemonitorización remota de rehabilitación y guante sensorizado asociado.	13
3.8	Sistema de detección de agarre mediante RSSI de Baccinelli et al. [4]: etiquetas RFID adheridas a objetos cotidianos y lector de muñeca para registro del contacto sin línea de visión.	14
3.9	Sistema de Lim et al. [25]: matriz de presión inalámbrica aplicada sobre un bloque del ARAT (izquierda) y mapa de distribución de presión resultante del agarre (derecha), que permite clasificar cinco patrones de agarre según la taxonomía GRASP con una precisión superior al 97 %.	15
3.10	Creación de volumen de movimiento adaptado al paciente [13]: el sistema estima el espacio de alcance funcional y reajusta automáticamente la disposición de los elementos interactivos.	17
4.1	Diagrama de Gantt del proyecto.	27
5.1	Arquitectura del sistema y flujo de datos extremo a extremo.	32

5.2	Representación virtual de los cubos del BBT superpuesta sobre el entorno físico real en el visor Meta Quest 3. El cubo de color apagado que se observa es el gemelo digital renderizado por el sistema de fusión, alineado con precisión sobre el cubo físico real que se encuentra debajo. La coincidencia espacial entre ambos es el resultado de la fusión visuo-inercial descrita en este capítulo.	36
5.3	Componentes del prototipo hardware: electrónica expuesta (a), disposición interna del sensor (b) y prototipo encapsulado final (c).	37
5.4	Interfaz de realidad mixta: retroalimentación en tiempo real y datos de los sensores recibidos por BLE, incluyendo valores de fuerza, orientación, movimiento inercial y nivel de batería.	43
5.5	Los tres artefactos principales de los sistemas de seguimiento exclusivamente visuales, resueltos mediante la validación táctil con FSR.	54
5.6	Configuración e inicio del juego serio de persecución temporal.	65
5.7	Mecánica del juego serio: colocación, acierto, penalización y transición entre rondas.	65
6.1	Comparación de estrategias de seguimiento, oclusión total e interfaz del juego serio.	71
6.2	Pruebas realizadas en el Hospital Nacional de Paraplégicos de Toledo.	71
6.3	Incidencia durante el agarre por tenodesis y resultado tras la corrección.	74
6.4	Resumen gráfico de la percepción subjetiva. Las dimensiones Likert se normalizan a 0–100; en cibermareo se representa tolerancia, no intensidad de síntomas.	75
6.5	Presentación del artículo en XRSalento2026.	76

Índice de tablas

3.1	Comparativa del presente trabajo con los principales enfoques del estado del arte. ✓: dimensión cubierta; ×: no cubierta; ~: cobertura parcial.	19
4.1	Resumen de las iteraciones del proyecto.	24
4.2	Materiales del prototipo	29
4.3	Resumen económico del proyecto	29
6.1	Latencia unidireccional estimada a partir de RTT ($N = 1\,121$).	68
6.2	Rendimiento de YOLOv8-Pose nano durante sesiones de juego real.	68
6.3	Error de registro visual con corrección de simetría.	69
6.4	Desglose por camino de calibración, ambos cubos agregados.	69
6.5	Comparación entre seguimiento visual y fusión híbrida.	70
6.6	Participantes y registros continuos disponibles.	71
6.7	Rendimiento funcional por participante.	72
6.8	Métricas cinemáticas por participante.	73
6.9	Indicadores de calidad del seguimiento por participante.	73
6.10	Resumen de la percepción subjetiva de los participantes.	75

Índice de listados

5.1	Conversión de cuaternión sensor → Unity en el callback BLE.	41
5.2	Resolución de nombre BLE por dirección MAC.	43
5.3	Letterboxing de la imagen de cámara a 640×640 preservando aspect ratio. .	45
5.4	Modo split-over-frames: ejecución distribuida del modelo YOLO.	46
5.5	Enrutamiento de detecciones por clase y color en el FusionSystemManager.	50
5.6	Interfaz de ShapePolicy: contrato para cada geometría.	52
5.7	Niveles de contacto del CubeContactAssessor.	53

Índice de algoritmos

5.1	Asociación de detecciones visuales	61
5.2	Fusión visuo-inercial y puente de oclusión	62

CAPÍTULO 1

Introducción

El presente trabajo se enmarca en la colaboración estratégica entre el grupo de investigación AIR (*Artificial Intelligence and Rendering*) de la Universidad de Castilla-La Mancha y el Hospital Nacional de Paraplégicos (HNP) de Toledo, con el objetivo de dar respuesta a las limitaciones observadas en la rehabilitación clínica de pacientes con daño neurológico.

El Hospital Nacional de Paraplégicos tiene como misión principal el tratamiento y acompañamiento de pacientes con patologías neurológicas e incapacitantes, tales como el ictus o las lesiones medulares [19]. Paralelamente, su labor investigadora permite aportar avances y soluciones innovadoras para la rehabilitación de estos usuarios. Clínicamente, estos pacientes se caracterizan por presentar discapacidades motoras totales o parciales; en el contexto de este trabajo, el foco se sitúa específicamente en las afecciones del miembro superior, cuya recuperación es esencial para la autonomía personal.

Dichos déficits suelen manifestarse a través de rangos de movimiento limitados, pérdida de la fuerza de prensión y posturas patológicas, como la posición de la mano en semiflexión. Tradicionalmente, la recuperación se basa en la ejecución de ejercicios repetitivos diseñados para reentrenar al paciente en la realización de las actividades de la vida diaria (AVD). La eficacia de estos procesos depende en gran medida del volumen de práctica, pero la naturaleza monótona de la terapia convencional suele derivar en una baja adherencia [22].

Por ello, considerando los factores psicológicos derivados del cambio repentino en las capacidades funcionales del individuo, se propone la integración de la gamificación y los juegos serios. Estas herramientas ofrecen la posibilidad de incrementar el rendimiento y el interés del paciente, permitiendo, además, una adaptación individualizada de los ejercicios según sus necesidades específicas [18].

Esta línea de trabajo constituye uno de los ejes principales del grupo de investigación AIR. El recorrido del grupo ha evolucionado en paralelo a los avances tecnológicos, transitando desde configuraciones semi-inmersivas basadas en sensores externos hasta el empleo de visores que permiten generar entornos inmersivos mediante distintas tecnologías de realidad extendida (XR). Estas se definen de la siguiente manera:

- **Realidad virtual (RV):** Creación de entornos completamente simulados que sustituyen la percepción del entorno físico por una inmersión digital total [20]. El paciente interactúa exclusivamente con objetos virtuales, lo que resulta especialmente adecuado en fases iniciales de la recuperación, cuando aún no existe capacidad de manipulación física.
- **Realidad aumentada (RA):** Superposición de capas de información digital sobre el entorno físico del usuario sin llegar a reemplazarlo [24]. Permite enriquecer la

percepción de los objetos reales con retroalimentación visual adicional.

- **Realidad mixta (RM):** Integración de elementos virtuales y reales de forma simultánea, permitiendo la interacción directa entre ambos mundos [24]. En el contexto de este trabajo, esta tecnología es especialmente relevante, ya que permite combinar objetos físicos manipulables con representaciones digitales superpuestas.

A pesar de sus ventajas, cada una de estas tecnologías presenta problemáticas inherentes a su naturaleza. En el caso de la realidad virtual pura, la ausencia de propiedades hápticas como tacto, peso y presión, constituye una limitación fundamental, tanto para los propios pacientes como para los profesionales de la rehabilitación. El uso de objetos puramente virtuales resulta artificial y limita la transferencia de los logros terapéuticos a la vida real. Esta carencia es especialmente relevante en fases avanzadas de la recuperación, cuando el paciente dispone ya de cierta capacidad motora residual y el entrenamiento requiere la manipulación de objetos con propiedades físicas reales para consolidar la mejoría funcional.

Una vez que la evolución del paciente permite trabajar con objetos físicos reales, surge el principal reto técnico de la realidad mixta: la oclusión. Cuando el paciente agarra un objeto físico durante la terapia, la mano cubre parcial o totalmente dicho objeto, impidiendo que las cámaras del visor mantengan su referencia visual. Este problema, unido a los falsos positivos en la detección del agarre (acentuados por la postura de semiflexión patológica) y a la persistencia de objetos virtuales tras la apertura de la mano, compromete la fiabilidad del sistema en las condiciones reales de uso clínico [16].

Recientemente, gracias a los trabajos de Herrera et al., se desarrolló un sistema capaz de adaptarse de forma personalizada al usuario mediante la creación de volúmenes de trabajo adaptados, permitiendo medir el rendimiento de cada paciente según su rango de movimiento específico [15, 13].

Como evolución lógica de esta línea de investigación, el presente trabajo propone equipar los objetos físicos de terapia con sensores integrados. Este enfoque permite trabajar con herramientas reales de la vida cotidiana, resolviendo los problemas de oclusión y la falta de fiabilidad en el agarre, al tiempo que restituye la validez ecológica de la terapia, entendida como la capacidad de que los logros obtenidos durante la rehabilitación en entorno clínico se trasladen de forma efectiva y directa a las actividades de la vida diaria del paciente.

1.1. CONTEXTO DISCIPLINAR Y TECNOLÓGICO

La integración de las tecnologías inmersivas en el ámbito clínico ha seguido un ciclo de maduración condicionado por la evolución del hardware comercial. Durante la última década, se ha producido una transición desde sistemas que requerían una infraestructura dedicada en sala, como cámaras de captura de movimiento externas o emisores de infrarrojos, hacia dispositivos autónomos con seguimiento óptico integrado [14]. Este cambio no solo ha reducido drásticamente los costes de implementación y mantenimiento, sino que ha permitido trasladar la terapia desde los laboratorios especializados a las salas comunes del hospital e, incluso, al propio domicilio del paciente, posibilitando una rehabilitación más ubicua, frecuente y autónoma.

No obstante, esta autonomía tecnológica introduce retos perceptivos y técnicos críticos. El rendimiento de estos sistemas depende de factores como la latencia, la precisión del registro espacial y la estabilidad de la imagen. Estudios previos sugieren que retrasos temporales superiores a ciertos umbrales pueden romper la sensación de presencia o incluso provocar malestar, mareos y fatiga ocular, fenómenos conocidos como *cybersickness* [2, 27]. En el

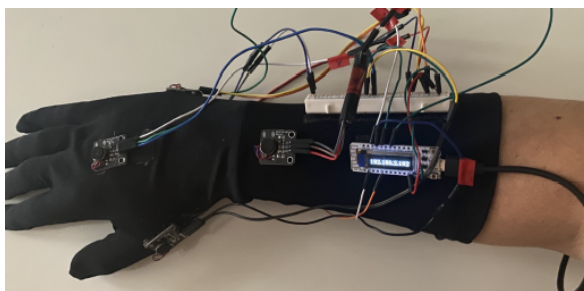


Figura 1.1: Guante sensorizado del TFG de Mario Pérez

contexto de la rehabilitación neurológica, donde el paciente puede presentar ya de por sí alteraciones sensoriales o de equilibrio, garantizar una respuesta fluida y síncrona del sistema es imperativo para que la experiencia sea terapéuticamente válida y segura.

En la actualidad, la instrumentación de la terapia para capturar datos cinemáticos objetivos sigue dos corrientes principales:

- **Instrumentación del usuario:** Se basa en el empleo de dispositivos *wearables* tales como guantes de datos, sensores inerciales corporales o interfaces electromiográficas [31]. Aunque ofrecen datos de alta fidelidad sobre la cinemática articular de forma independiente a la oclusión visual, presentan barreras de usabilidad significativas. En pacientes neurológicos con espasticidad severa, hipertonía o edemas, la colocación de estos dispositivos puede resultar lenta, incómoda o incluso contraindicada desde un punto de vista clínico [4].
- **Instrumentación del entorno (*Smart Objects*):** Consiste en dotar de inteligencia a los objetos que el paciente manipula habitualmente durante su terapia. Este enfoque, defendido en el presente trabajo, permite registrar la interacción de forma transparente para el usuario [4, 30]. Al integrar sensores de presión e inerciales directamente en el objeto terapéutico, se preserva la ergonomía original y se reduce el tiempo de preparación de la sesión clínica, lo que es vital en entornos hospitalarios con alta carga de trabajo.

El principal problema técnico surge al intentar combinar estos objetos físicos con visores de realidad mixta. Los algoritmos de visión por computador actuales suelen perder la referencia del objeto cuando este es cubierto por la mano del paciente durante un agarre firme [16]. Para solventar esto, la tendencia actual apunta hacia la fusión sensorial: combinar la información visual del visor con los datos inerciales transmitidos inalámbricamente desde el propio objeto inteligente, garantizando un seguimiento robusto y continuo [11].

1.2. JUSTIFICACIÓN DEL PROYECTO

La justificación de este proyecto radica en la necesidad de dotar de un realismo físico y una validez ecológica superior a la rehabilitación virtual. Durante las pruebas clínicas realizadas previamente con el sistema *Rehab-Immersive* en el HNP, se observó que la interacción con mandos estándar o mediante seguimiento de manos al aire no resultaba suficiente para pacientes con cierta capacidad motora residual [15, 14]. Para estos perfiles, el entrenamiento pierde eficacia si el sistema nervioso central no recibe la retroalimentación háptica coherente derivada de la textura, el peso y la inercia del objeto real.

Uno de los pilares de este trabajo es la replicación virtual y automatizada de pruebas estándar de destreza manual, como el *Box and Block Test* (BBT) [29], mediante el diseño y

desarrollo de objetos inteligentes combinados con técnicas de realidad mixta. Aunque existen versiones virtuales validadas que utilizan sensores como Leap Motion, la mayoría presentan limitaciones en la detección precisa del momento exacto del agarre bajo condiciones de manipulación real [3]. La sensorización de cada cubo físico del test mediante sensores de presión y acelerómetros reduce el error derivado de la oclusión visual y permite medir parámetros cinemáticos que no pueden obtenerse con un cronómetro manual, como la suavidad del movimiento, la fuerza de prensión o la aceleración durante el transporte del bloque [37, 21].

Por lo tanto, este trabajo se justifica bajo tres dimensiones fundamentales que refuerzan su relevancia:

1. **Dimensión Clínica:** Mejora la transferencia funcional de los ejercicios realizados en el hospital a las actividades de la vida diaria del paciente. Al entrenar con objetos físicos reales dentro de un entorno digital motivador, el cerebro procesa estímulos coherentes que favorecen la neuroplasticidad y la recuperación motora efectiva.
2. **Dimensión Tecnológica:** Resuelve el conflicto de la coherencia visuo-háptica en realidad mixta. Mediante el uso de sistemas embebidos, se garantiza que la representación virtual del objeto coincida con el objeto físico en la mano del paciente, incluso cuando las cámaras del visor no pueden verlo.
3. **Dimensión de Usabilidad y Sostenibilidad:** Propone un sistema transparente y de bajo coste. Al integrar la tecnología en la parte interior del objeto, se elimina el estigma de los dispositivos médicos complejos sobre el cuerpo del paciente. Además, el uso de hardware modular y accesible facilita que el paciente pueda disponer de un sistema de estas características en su propio domicilio, ampliando la rehabilitación más allá del entorno hospitalario e integrándola de forma continua en su rutina diaria.

1.3. ESTRUCTURA DEL DOCUMENTO

El presente documento se organiza en los capítulos que se detallan a continuación:

- **Objetivo:** Se definen las metas generales y los hitos específicos que guían el desarrollo de la solución propuesta.
- **Estado del arte:** Se realiza una revisión profunda de las tecnologías XR aplicadas a la salud, analizando desde los sistemas de seguimiento hasta las estrategias de gamificación actuales.
- **Metodología:** Se describen el proceso de diseño y las decisiones metodológicas tomadas para la integración del hardware y el software.
- **Arquitectura:** Se detalla el diseño del objeto inteligente, incluyendo la elección de sensores, el protocolo de comunicación inalámbrica, su integración con el visor Meta Quest 3 y el sistema de fusión ejecutado en las gafas del paciente.
- **Resultados:** Se presentan las pruebas de validación técnica, analizando la latencia, la precisión del seguimiento híbrido y la robustez ante oclusiones.
- **Conclusiones:** Se resumen las aportaciones del trabajo y se proponen líneas futuras para la implantación definitiva del sistema en el entorno hospitalario.

CAPÍTULO 2

Objetivos

Este capítulo presenta el objetivo principal del proyecto y su desglose en objetivos específicos agrupados por temática.

2.1. OBJETIVO GENERAL

El objetivo de este trabajo es diseñar, desarrollar e integrar un sistema de objetos sensorizados para la rehabilitación del miembro superior en entornos de realidad mixta. El problema técnico que motiva el proyecto es la oclusión visual: cuando el paciente agarra un objeto, su propia mano lo cubre y los sistemas de seguimiento óptico pierden la referencia precisamente en el momento más relevante del ejercicio terapéutico. Para resolverlo, el sistema combina seguimiento óptico e inercial, manteniendo la correspondencia entre el objeto físico y su representación virtual durante la interacción y registrando información cuantitativa de utilidad para la evaluación clínica.

En términos funcionales, el sistema debe:

- Permitir la manipulación natural de objetos terapéuticos físicos en ejercicios de rehabilitación del miembro superior.
- Mantener la correspondencia espacial entre el objeto físico y su representación virtual, incluyendo situaciones de oclusión manual.
- Detectar y cuantificar la fuerza de agarre ejercida por el paciente sobre el objeto.
- Registrar información cuantitativa sobre el movimiento y la interacción para su uso en evaluación funcional.
- Integrarse en una aplicación de rehabilitación basada en juegos serios con objetos físicos reales.

2.2. OBJETIVOS ESPECÍFICOS

A continuación se detallan los objetivos específicos que articulan la consecución del objetivo general, abarcando desde el análisis previo de tecnologías hasta la validación del sistema completo.

OBJ1. Análisis y selección de tecnologías para la sensorización de objetos terapéuticos

Con carácter previo al desarrollo, se llevará a cabo un estudio comparativo de las tecnologías disponibles para la instrumentación de objetos de rehabilitación. El análisis abarcará soluciones de captura de movimiento, medición de fuerza y comunicación inalámbrica, evaluando su

viabilidad para integrarse en objetos de pequeño formato sin comprometer sus propiedades terapéuticas ni ergonómicas.

OBJ2. Diseño e implementación del objeto terapéutico instrumentado

Este objetivo aborda el diseño y construcción del prototipo físico: un objeto de rehabilitación que incorpora sensores para capturar movimiento y fuerza de agarre, y que transmite esos datos de forma inalámbrica. El diseño debe preservar el peso, el agarre y las propiedades hápticas del objeto original, de modo que no interfiera con la ejecución natural del ejercicio terapéutico. Se incluye también el desarrollo del software embebido que gobierna el funcionamiento del dispositivo.

OBJ3. Desarrollo de la capa de comunicación e integración en el entorno de realidad mixta

Este objetivo cubre el desarrollo de la interfaz software encargada de recibir e interpretar en tiempo real los datos procedentes de los objetos sensorizados, asociando cada uno con su representación virtual correspondiente. El sistema debe ser capaz de gestionar esta asociación de forma robusta cuando hay varios objetos activos simultáneamente.

OBJ4. Implementación del algoritmo de seguimiento resistente a la oclusión

Este objetivo constituye el núcleo técnico del trabajo. Consiste en desarrollar el mecanismo de fusión que combina la información visual del entorno con los datos inerciales y táctiles del objeto, con el fin de mantener su posición y orientación correctas cuando la mano del paciente lo obstruye. Sin este componente, el sistema presentaría las mismas limitaciones que cualquier solución óptica convencional, fallando durante el agarre.

OBJ5. Desarrollo de una aplicación de rehabilitación y validación técnica del sistema

Este objetivo comprende dos aspectos complementarios. El primero es la integración del sistema en una aplicación de rehabilitación con juegos serios que permita ejecutar ejercicios funcionales del miembro superior con objetos físicos reales. El segundo es la validación técnica del sistema con carácter previo a su uso clínico, verificando la continuidad del seguimiento, la latencia de comunicación y la fiabilidad de los datos registrados.

CAPÍTULO 3

Estado del arte

3.1. REALIDAD EXTENDIDA EN LA REHABILITACIÓN DEL MIEMBRO SUPERIOR

Las patologías neurológicas de alta prevalencia, el ictus y la lesión medular principalmente, generan déficits motores persistentes cuya recuperación exige un volumen elevado de práctica repetitiva. El principal obstáculo no es el protocolo clínico en sí, sino mantener la adherencia terapéutica a largo plazo: la naturaleza monótona de los ejercicios convencionales provoca abandono y ralentiza la recuperación. Esta circunstancia impulsó, durante la última década, la incorporación de tecnologías de Realidad Extendida (XR) como herramienta tanto terapéutica como de evaluación. Dentro del espectro XR, RV proporciona un entorno digital controlado, mientras que la RM integra el mundo físico y permite interactuar simultáneamente con elementos digitales y objetos reales.

La RV fue la primera modalidad en alcanzar madurez clínica suficiente para su aplicación sistemática. Lee et al. [22] demostraron mejoras significativas en la función del miembro superior en pacientes con ictus crónico frente a terapia convencional. Leong et al. [24] confirmaron ventajas estadísticamente significativas en los grupos de mayor intensidad de tratamiento, y Karamians et al. [18] demostraron que las intervenciones basadas en RV y videojuegos son más efectivas que los métodos convencionales en la recuperación motora del miembro superior, con mayor beneficio cuando la intervención incorpora un componente de juego activo. La revisión Cochrane de Laver et al. [20], que sintetiza 72 ensayos controlados aleatorizados, aporta el matiz clínicamente más relevante: cuando la RV se emplea como terapia adjunta que incrementa el volumen total de práctica, produce mejoras estadísticamente significativas en la función del miembro superior y en las actividades de la vida diaria, situando el modelo de integración complementaria como la estrategia con mayor respaldo empírico. Esta revisión identifica además la mejora de la fidelidad de la interacción, en particular en tareas que requieren manipulación de objetos, como la principal línea de desarrollo pendiente.

El mecanismo principal a través del cual las plataformas XR consiguen mantener la motivación es el uso de *serious games*. Al envolver las tareas terapéuticas en un contexto interactivo, los entornos gamificados incrementan el compromiso del paciente y, según la literatura de neurociencia motora, potencian los mecanismos de plasticidad cortical al aumentar la saliencia de la tarea durante el aprendizaje motor [19, 32].

La trayectoria del grupo del Hospital Nacional de Paraplégicos (HNP) de Toledo ilustra de forma representativa cómo ha evolucionado el campo. Los primeros pasos se dieron en el proyecto europeo *Naviga* (2012), donde la Unidad de Biomecánica y Ayudas Técnicas llevó a cabo un estudio piloto con el guante CyberGlove®: un dispositivo que captura el

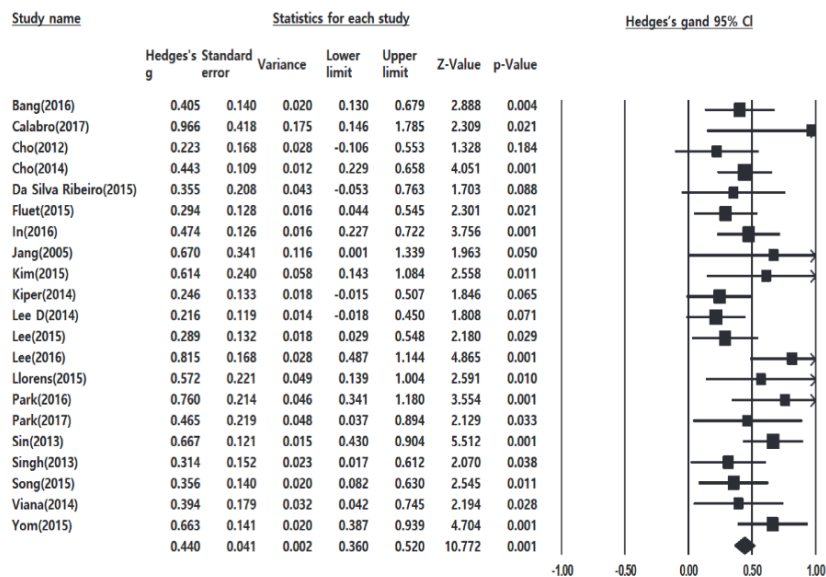


Figura 3.1: Diagrama de bosque (forest plot) que ilustra la diferencia de medias estandarizada (g de Hedges) de las intervenciones con realidad virtual sobre la función del miembro superior en pacientes con ictus crónico. El tamaño del efecto global (ES = 0,440) indica una mejora moderada y estadísticamente significativa en comparación con la rehabilitación convencional.

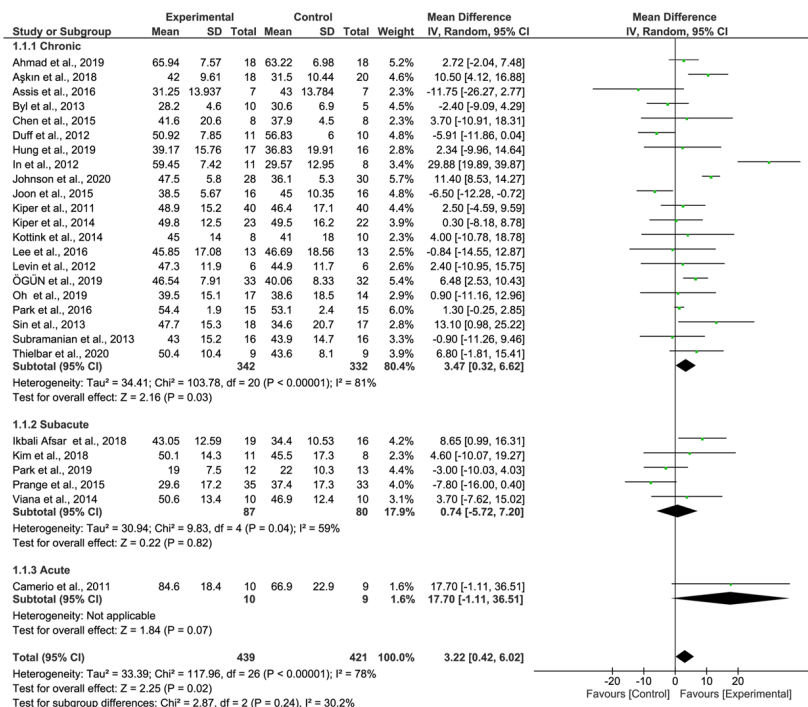


Figura 3.2: Diagrama de bosque (forest plot) de la escala Fugl-Meyer para el miembro superior (FMA-UE) categorizada por estadio de recuperación. Los resultados demuestran que la terapia VAMR (realidad virtual, aumentada y mixta) es significativamente superior al tratamiento estándar en la fase crónica (DM = 3,47; $p = 0,03$).

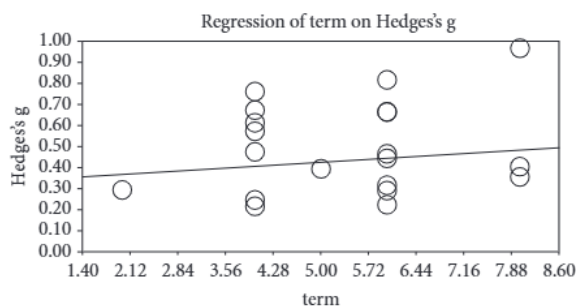


Figura 3.3: Análisis de meta-regresión de los parámetros de la intervención con RV. Arriba: correlación entre la duración de la intervención y la ganancia funcional, mostrando un aumento de la eficacia a medida que el tratamiento se acerca a las 8 semanas. Abajo: impacto de la frecuencia de las sesiones semanales en el tamaño del efecto.

movimiento de los dedos mediante sensores de flexión y lo representa en tiempo real, con retroalimentación vibrotáctil, para ejercicios de agarre, suelta y alcance de objetos [35].

Poco después, la colaboración del HNP con Indra produjo TOYRA, una plataforma que combinaba el Kinect de Microsoft con sensores inerciales para telerrehabilitación domiciliaria del miembro superior [12], evaluada posteriormente mediante ensayo clínico aleatorizado en personas con tetraplejia [10]. A partir de esa base, el grupo desarrolló RehabHand, una plataforma de RV de bajo coste construida sobre el Leap Motion Controller (LMC). Su estudio piloto demostró buena adherencia, ausencia de efectos adversos y alta satisfacción del usuario [8].

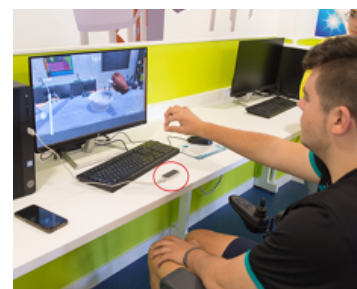
El paso siguiente fue el proyecto Rehab-Immersive, desarrollado entre 2022 y 2025 en colaboración con el grupo AIR de la Universidad de Castilla-La Mancha. Este proyecto adoptó RV completamente inmersiva mediante *Head-Mounted Displays* (HMDs), concretamente las gafas Meta Quest 2 y 3. Herrera et al. describieron su arquitectura software modular, con capacidad de registro cinemático, personalización del entorno según las limitaciones de cada paciente y retroalimentación en tiempo real [15]; trabajos posteriores demostraron mejoras significativas en usabilidad y satisfacción [14, 13]. El presente TFG se inscribe directamente en esta línea, extendiendo el ecosistema Rehab-Immersive hacia la realidad mixta con objetos físicos instrumentados.



(a) Guante CyberGlove en el proyecto Naviga [35].



(b) Plataforma TOYRA con Kinect e inerciales [12].



(c) Leap Motion Controller [8].



(d) Framework Rehab-Immersive con Meta Quest [15].

Figura 3.4: Evolución tecnológica de los sistemas de rehabilitación del miembro superior en el HNP y el grupo AIR.

3.2. LIMITACIONES DE LOS SISTEMAS DE INTERACCIÓN EN XR PARA REHABILITACIÓN

En rehabilitación del miembro superior, el objetivo terapéutico no se limita a desplazar la mano en el espacio, sino a reproducir de forma fiable los componentes que determinan la ejecución de actividades de la vida diaria: alcance, prehensión, manipulación fina, modulación de fuerza y soltado controlado. Leong et al. [24] señalaron la existencia de una brecha de fidelidad entre la interacción que ofrecen los dispositivos XR comerciales y las demandas motoras ecológicas requeridas en terapia ocupacional, especialmente en tareas de destreza manual y agarre funcional.

3.2.1. Oclusión en el seguimiento óptico

Los visores modernos de RV y RM emplean sistemas de seguimiento *inside-out* basados en visión por computador. Cuando se trabaja con objetos físicos en RM, la localización espacial depende de características visuales del objeto o de marcadores fiduciaros, lo que exige visibilidad directa entre las cámaras del dispositivo y el objeto. También deben tenerse en cuenta unas adecuadas condiciones de luminosidad y libertad en el espacio de trabajo para que estos se reconozcan adecuadamente. Por todo ello, durante la ejecución natural de una tarea de agarre, la propia mano envuelve el objeto y genera auto-oclusión, interrumpiendo el seguimiento.

La pérdida de seguimiento provoca discontinuidades en la estimación de la pose del objeto, degradando tanto la coherencia espacial entre el elemento real y su representación virtual como la calidad del registro cinemático. Estas inconsistencias afectan a métricas de evaluación funcional como la precisión espacial, la suavidad del movimiento o el tiempo de ejecución. Además, el desacoplamiento entre lo que el paciente siente y lo que observa en el entorno virtual reduce la sensación de presencia e inmersión, factores directamente relacionados con la motivación y la adherencia terapéutica.

3.2.2. Problemas de reconocimiento del agarre

Los algoritmos de *hand tracking* de los visores comerciales infieren el agarre a partir de configuraciones geométricas de los dedos o de colisiones virtuales aproximadas con el objeto. En población neurológica, las alteraciones del tono muscular, la espasticidad o la postura en

semiflexión pueden simular un cierre de mano sin que exista intención real de agarre; los sistemas basados exclusivamente en visión tienden en consecuencia a generar falsos positivos: adhesión prematura del objeto, agarres persistentes o liberaciones incorrectas. Estos errores introducen ambigüedad en variables terapéuticas como el número de repeticiones, el tiempo de manipulación o la calidad del gesto.

3.2.3. Ausencia de interacción tangible

En entornos de RV pura, los controladores genéricos o la manipulación en el aire impiden reproducir propiedades fundamentales de los objetos cotidianos como el peso, textura, fricción o inercia del objeto. La recuperación funcional del miembro superior incluye no solo la coordinación cinemática sino también la capacidad de generar y regular la fuerza de prensión, habilidad que no puede entrenarse sin interacción física real. El sistema neuromotor necesita ajustar el esfuerzo en función de las propiedades mecánicas del objeto; sin esa información háptica, la transferencia del aprendizaje a la vida cotidiana queda comprometida.

Estas tres limitaciones de oclusión, detección del agarre y ausencia de tangibilidad, han motivado dos líneas de investigación complementarias: instrumentar al paciente mediante wearables (Sección 3.3) e instrumentar los propios objetos del entorno terapéutico (Sección 3.4).

3.3. WEARABLES Y SENSORIZACIÓN DEL PACIENTE EN REHABILITACIÓN

Instrumentar al paciente mediante dispositivos vestibles permite capturar datos de alta fidelidad sobre el movimiento sin depender de la línea de visión con el objeto manipulado. Patel et al. [31] revisaron esta familia tecnológica, identificando tres grandes categorías: guantes instrumentados, sistemas inerciales corporales e interfaces de electromiografía, concluyendo que ninguna resuelve de forma integral el problema de la evaluación funcional continua en entornos ecológicamente válidos.

3.3.1. Guantes instrumentados

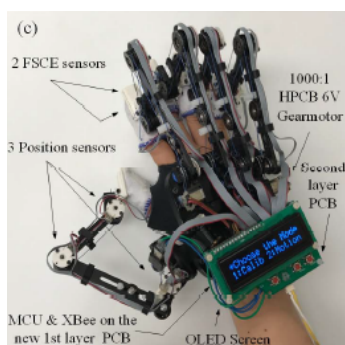
Los guantes de datos integran sensores de flexión sobre cada segmento digital, sensores de presión en las yemas y sensores inerciales sobre el dorso de la mano, proporcionando una representación tridimensional de la postura y las fuerzas de contacto durante el agarre. El guante CyberGlove®, empleado en el proyecto Naviga [35], es el referente histórico de esta aproximación, aunque su coste y la complejidad de calibración limitaron su adopción clínica.

Connelly et al. [7] desarrollaron un guante neumático capaz de asistir la extensión de los dedos durante tareas de alcance en RV, demostrando que es viable combinar asistencia física y retroalimentación visual inmersiva en rehabilitación post-ictus. Dimbwadyo-Terrer et al. [9] validaron clínicamente un guante de datos integrado en un entorno de RV inmersiva para lesión medular, con mejoras significativas tanto en función motora como en calidad de vida.

Un avance relevante fue el sistema iSAFER de Lee et al. [21], que incorpora sensores elastoméricos conductores de fuerza para detectar el deslizamiento del objeto durante el agarre. Modular la fuerza de prensión en función de las propiedades del objeto es una de las habilidades más difíciles de recuperar tras una lesión neurológica, y la detección del deslizamiento permite dar retroalimentación sobre la calidad del agarre más allá del simple registro de si el objeto fue cogido o soltado.



(a) Guante neumático de Connolly et al. [7] para asistencia a la extensión digital en RV.



(b) Sistema iSAFER [21]: sensores elastoméricos para detección de deslizamiento durante el agarre.

Figura 3.5: Sistemas de instrumentación del paciente mediante guantes en rehabilitación neurológica.

3.3.2. Sistemas inerciales corporales

Las unidades de medición inercial (IMU) colocadas sobre los segmentos de interés como muñeca, antebrazo o brazo permiten reconstruir la cinemática tridimensional mediante la integración de acelerómetro y giroscopo. Su ventaja principal es la independencia total de las condiciones de iluminación y la línea de visión, lo que las hace robustas en entornos clínicos variables y en telerrehabilitación domiciliaria.

Hunt et al. [16] demostraron que la fusión de señales inerciales con sistemas de realidad aumentada permite estimar trayectorias de movimiento de manera predictiva, compensando la latencia inherente a los sistemas ópticos. No obstante, los sistemas IMU presentan limitaciones clínicas relevantes: la deriva del giroscopo, especialmente en el eje de guiñada en configuraciones sin magnetómetro, acumula error si no se dispone de referencias absolutas externas. Además, la reconstrucción de la pose de la mano requiere modelos cinemáticos del esqueleto y protocolos de calibración anatómica que resultan problemáticos en pacientes con espasticidad, posturas compensatorias o asimetría funcional marcada [16].

3.3.3. Electromiografía de superficie y enfoques híbridos

La electromiografía de superficie (sEMG) añade información sobre la intención motora que los sistemas puramente cinemáticos no pueden capturar, especialmente valiosa en pacientes con movimientos muy reducidos o en fases tempranas de la recuperación, donde la intención puede existir sin que se traduzca en desplazamiento articular observable.

El sistema wBBT de Zhang et al. [37] combina señales del brazalete Myo —portado en el antebrazo del sujeto— con modelos CNN-LSTM para automatizar el conteo de bloques en el Box and Block Test con una precisión superior al 99 %, capturando simultáneamente parámetros cualitativos del movimiento. El brazalete adquiere ocho canales de EMG de superficie junto con acelerómetro, giroscopo y orientación, sin modificar en absoluto los bloques ni el material original del test. El inconveniente es que requiere infraestructura de procesamiento externo considerable y protocolos de colocación muy precisos, lo que limita su viabilidad en entornos clínicos sin personal técnico especializado.

Postolache et al. [33] exploran la integración de guantes inteligentes en arquitecturas IoT para telemonitorización, evidenciando el potencial de estos dispositivos en modelos de

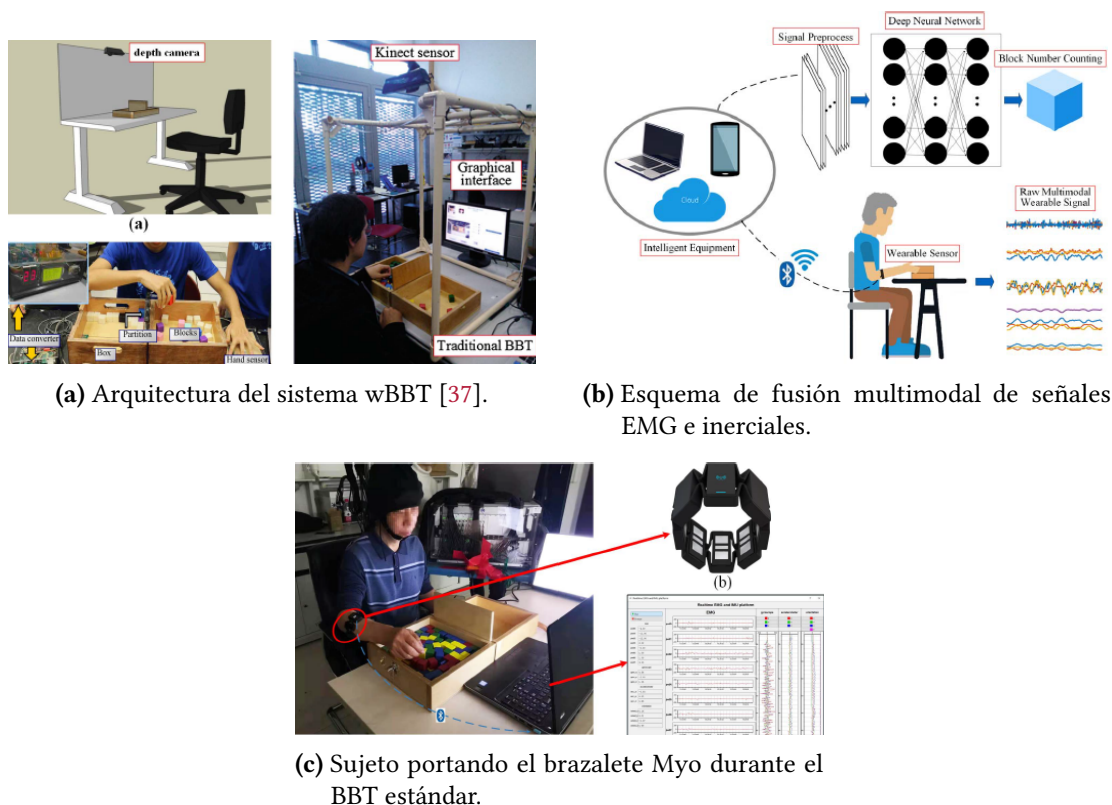


Figura 3.6: Sistema wBBT de Zhang et al. [37]: arquitectura, fusión de señales EMG e inerciales, y uso durante el Box and Block Test estándar.

atención distribuida.

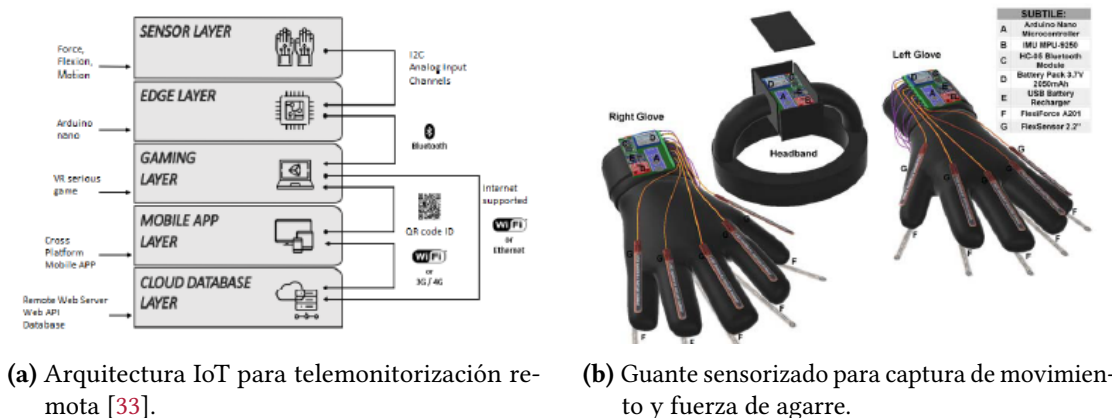


Figura 3.7: Sistema de Postolache et al. [33]: arquitectura IoT para telemonitorización remota de rehabilitación y guante sensorizado asociado.

3.3.4. Limitaciones estructurales del paradigma vestible

La colocación de guantes, brazaletes o electrodos sobre extremidades con espasticidad, hipertonía, edema post-ictus o contracturas puede resultar incómoda, dolorosa o clínicamente contraindicada. Cada paciente requiere un proceso de calibración individualizado que consume tiempo clínico, y la presencia de instrumentación visible sobre la mano puede alterar la percepción del paciente sobre su propia capacidad motora y modificar los patrones de agarre, comprometiendo la validez ecológica del ejercicio [4]. Estas limitaciones son especialmente

pronunciadas en los perfiles de mayor severidad, precisamente los que más necesitan una sensorización fiable.

3.4. OBJETOS INTELIGENTES Y ENTORNOS INSTRUMENTADOS EN REHABILITACIÓN

La alternativa es desplazar la instrumentación desde el cuerpo del paciente hacia los propios objetos terapéuticos, convirtiéndolos en nodos inteligentes capaces de registrar la interacción de forma autónoma. Este paradigma conocido en la literatura como *smart objects*, preserva la libertad de movimiento del usuario y la validez ecológica de la tarea al mantener intactas las propiedades hápticas y ergonómicas del objeto.

3.4.1. Fundamentos del paradigma de instrumentación del entorno

El concepto de objeto inteligente en rehabilitación se inscribe en el campo más amplio del Internet de las Cosas (IoT) aplicado a la salud. Baccinelli et al. [4] formularon una de las primeras taxonomías sistemáticas de los *smart objects*, distinguiendo entre los orientados a detección de contacto, los dedicados a captura cinemática y los que combinan ambas funcionalidades, e identificaron como reto principal el equilibrio entre la riqueza de la señal capturada y el impacto de la instrumentación sobre las propiedades hápticas del objeto.

Las primeras implementaciones emplearon etiquetas RFID adheridas a objetos cotidianos: midiendo la intensidad de señal recibida (RSSI) mediante un lector de muñeca se detecta el contacto sin línea de visión directa [4]. La señal RSSI, sin embargo, es unidimensional; no permite capturar orientación, velocidad angular ni fuerza de agarre, variables de mayor valor clínico en la evaluación funcional del miembro superior.

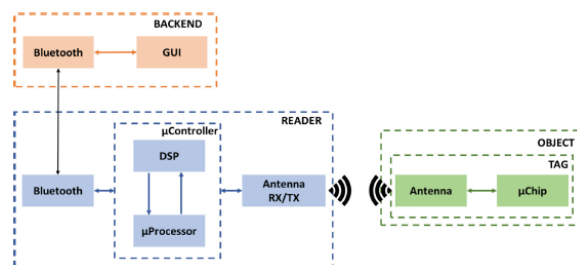


Figura 3.8: Sistema de detección de agarre mediante RSSI de Baccinelli et al. [4]: etiquetas RFID adheridas a objetos cotidianos y lector de muñeca para registro del contacto sin línea de visión.

Monge y Postolache [30] avanzaron al proponer sensores inerciales embebidos en objetos terapéuticos comunicados mediante BLE con un sistema de realidad aumentada, transmitiendo datos cinemáticos en tiempo real. El firmware que implementaron seguía una arquitectura *super-loop* sin sistema operativo en tiempo real, lo que implica una gestión ineficiente del procesador, consumo energético elevado en reposo y garantías insuficientes de determinismo temporal en la ejecución concurrente de las tareas de adquisición y comunicación.

3.4.2. Instrumentación del Box and Block Test

El Box and Block Test (BBT) es una de las pruebas estandarizadas de evaluación de la destreza manual unilateral más utilizadas en neurorrehabilitación [29]. Consiste en transportar el mayor número posible de bloques cúbicos de 2,5 cm de lado de un compartimento a otro en 60 segundos; sus resultados correlacionan con la capacidad funcional del miembro superior

en actividades de la vida diaria para una amplia variedad de diagnósticos neurológicos, lo que lo convierte en un candidato natural para la instrumentación inteligente.

Alvarez-Rodríguez et al. [3] realizaron una validación concurrente de una versión virtual del BBT en pacientes con trastornos neurológicos, demostrando que la correlación con la prueba física original justifica su uso como herramienta de evaluación clínica. Desarrollado en el HNP, este trabajo constituyó un primer paso hacia la integración del BBT en entornos XR, aunque sin incorporar interacción física real con objetos instrumentados.

En cuanto a la instrumentación física del propio material del test, Lee et al. [23] propusieron el sistema eBBT, que incorpora sensores de infrarrojos y fuerza en la partición de la caja para detectar el paso de cada bloque y cuantificar el esfuerzo compensatorio durante el traslado, sin requerir modificación de los bloques. Este enfoque de instrumentar el entorno en lugar del sujeto o los objetos anticipa el paradigma de *smart objects* descrito en la siguiente sección.

El trabajo más próximo al del presente TFG en cuanto a la instrumentación directa del objeto terapéutico es el de Lim et al. [25], que desarrollaron un módulo de matriz de presión inalámbrica adaptable a objetos cilíndricos y cúbicos, con hasta 1024 elementos sensores fabricados con Velostat —un material piezorresistivo de bajo coste— dispuestos en una estructura de filas y columnas ortogonales. El módulo se aplicó directamente sobre la superficie de un vaso cilíndrico y de un bloque de 7,5 cm empleado en el *Action Research Arm Test* (ARAT), cubriendo los objetos con la lámina sensora sin modificar su geometría ni sus propiedades táctiles. Los datos se transmiten de forma inalámbrica a una frecuencia superior a 30 Hz mediante ESP32, suficiente para capturar la dinámica del agarre en tiempo real. Mediante un algoritmo de *k-nearest neighbours* (KNN) entrenado sobre 50 repeticiones por clase, el sistema clasifica cinco patrones de agarre extraídos de la taxonomía GRASP, obteniendo precisiones del 98,0 % para el objeto cilíndrico y del 97,6 % para el cúbico. Una segunda aplicación del mismo módulo permitió estimar la pose bidimensional de objetos geométricos sobre la superficie sensora, con errores medios de traslación de 2,24 mm y de orientación de 2,33° en la configuración de mayor resolución (32×32 elementos).

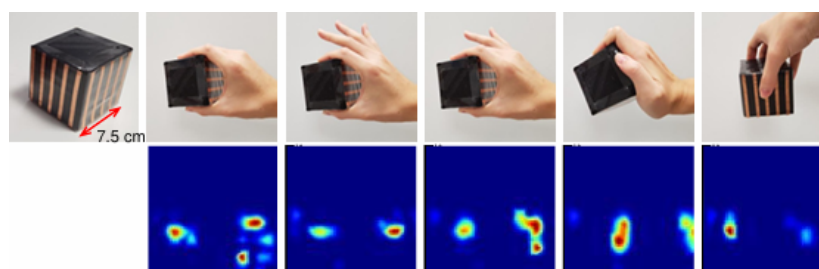


Figura 3.9: Sistema de Lim et al. [25]: matriz de presión inalámbrica aplicada sobre un bloque del ARAT (izquierda) y mapa de distribución de presión resultante del agarre (derecha), que permite clasificar cinco patrones de agarre según la taxonomía GRASP con una precisión superior al 97 %.

La propuesta de Lim et al. representa el estado del arte más avanzado en la instrumentación táctil directa del objeto en contexto rehabilitador. Sin embargo, presenta tres limitaciones que el presente trabajo aborda de forma explícita. En primer lugar, el módulo no integra datos inerciales: la posición y orientación tridimensional del objeto durante la manipulación no se registran, lo que impide construir una trayectoria cinemática completa del transporte. En segundo lugar, el sistema carece de integración con ningún entorno de realidad mixta; la información táctil se procesa en un ordenador externo y no existe representación virtual

sincronizada del objeto. En tercer lugar, la arquitectura hardware basada en ESP32 con Wi-Fi no está optimizada para el consumo energético requerido en sesiones clínicas de larga duración dentro de objetos de reducidas dimensiones como los cubos del BBT estándar de 2,5 cm. El presente trabajo adopta un planteamiento complementario: en lugar de cubrir la superficie del objeto con una densa malla sensora, integra un único sensor de fuerza resistivo (FSR) de geometría mínima junto a una IMU de seis ejes, lo que permite capturar simultáneamente la intención de agarre y la cinemática rotacional dentro del factor de forma del cubo, transmitiendo ambas señales mediante BLE a 100 Hz hacia el visor Meta Quest 3 para su fusión con el sistema de visión por computador.

3.4.3. Convergencia con entornos de realidad mixta

La convergencia entre objetos inteligentes y entornos XR ha dado lugar a sistemas que persiguen simultáneamente la sensorización de los objetos, su integración en entornos inmersivos y la preservación de las propiedades hápticas. D'Ippolito y Massaro [11] demostraron que la fusión de datos de múltiples sensores a distintas tasas de muestreo (*multi-rate sensor fusion*) es un requisito fundamental para mantener la coherencia espacial del entorno mixto durante la interacción física, y que combinar señales inerciales de alta frecuencia con detecciones ópticas de baja frecuencia permite compensar las limitaciones de latencia de los sistemas de visión por computador.

Para la identificación de objetos en escenarios de RM multiobjeto como el BBT, en el cual todos los objetos son cubos, Ahuja et al. [1] propusieron el sistema LightAnchors, que aprovecha los propios LED de los objetos instrumentados como identificadores ópticos activos. El enfoque evita marcadores fiduciarios externos, pero requiere LED visibles incorporados, lo que modifica la apariencia del objeto y puede alterar la naturaleza táctil de la interacción en contextos terapéuticos donde la transparencia del hardware es un requisito primordial.

Pillai et al. [32] abordan la gamificación de la rehabilitación del miembro superior en RM con HoloLens 2, pero operan exclusivamente con *hand tracking* óptico, sin sensorización física de los objetos ni mecanismos de robustez ante la oclusión manual. La integración de objetos físicos instrumentados con visores de RM autónomos, en particular con el ecosistema Meta Quest, no cuenta todavía con implementaciones de referencia consolidadas en la literatura de rehabilitación.

3.5. ADAPTACIÓN DEL ESPACIO TERAPÉUTICO A LAS CAPACIDADES DEL PACIENTE

La heterogeneidad de la población neurológica en términos de rango de movimiento articular, fuerza residual, nivel cognitivo y tolerancia a la fatiga exige que el entorno se adapte a las limitaciones funcionales específicas de cada paciente. Sin estos mecanismos de personalización, la misma sesión puede resultar trivial para un paciente en fase avanzada de recuperación e inalcanzable para otro con mayor afectación, comprometiendo la eficacia terapéutica en ambos casos.

3.5.1. Adaptación del espacio de trabajo al rango de movimiento

Los objetos y elementos interactivos de los entornos RV estándar se distribuyen siguiendo convenciones de diseño pensadas para usuarios sin limitaciones motoras, lo que puede hacer que determinadas posiciones resulten inaccesibles para pacientes con movilidad reducida o asimetría funcional. Herrera et al. [13] abordaron este problema en el proyecto Rehab-Immersive desarrollando un algoritmo de calibración automática del espacio de

trabajo periespinal. El sistema estima el volumen de alcance funcional mediante una fase de calibración inicial en la que el paciente realiza movimientos de máximo alcance en diferentes planos; a partir de esos datos, genera un volumen elipsoidal adaptado al rango de movimiento individual y reajusta automáticamente la posición, escala y disposición de los elementos interactivos, con mejoras significativas en usabilidad y percepción de agencia sobre el entorno.

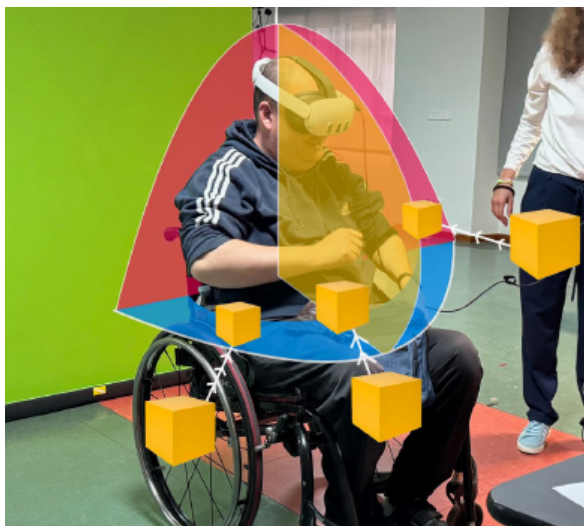


Figura 3.10: Creación de volumen de movimiento adaptado al paciente [13]: el sistema estima el espacio de alcance funcional y reajusta automáticamente la disposición de los elementos interactivos.

3.5.2. Ajuste progresivo de la dificultad

Los sistemas XR de rehabilitación avanzados incorporan ajuste dinámico de la dificultad durante la sesión, siguiendo los principios del aprendizaje motor progresivo y de la teoría del flujo [28]. El objetivo es mantener al paciente en una zona de actividad óptima donde el desafío estimula la reorganización cortical sin generar frustración ni compensaciones articulares que enmascaren el déficit motor real. El framework Rehab-Immersive implementa estos mecanismos ajustando parámetros como la velocidad de los estímulos, el tamaño y posición de los objetivos, el tiempo disponible o el nivel de asistencia virtual proporcionada [15]; Herrera et al. [14] demostraron que estos ajustes mejoran las métricas de usabilidad y reducen la varianza en el rendimiento entre sesiones.

La integración de objetos físicos instrumentados en este marco adaptativo introduce nuevas posibilidades: la fuerza de agarre registrada, la fluidez de la trayectoria o el tiempo de manipulación pueden alimentar el motor de adaptación en tiempo real, enriqueciendo la señal de rendimiento más allá de los indicadores puramente cinemáticos disponibles en los sistemas de RV.

3.5.3. Retroalimentación y telerrehabilitación

Los sistemas XR avanzados proporcionan información continua sobre la trayectoria, la velocidad y la suavidad del movimiento, lo que ha demostrado mejorar la tasa de aprendizaje motor frente al entrenamiento sin retroalimentación explícita [19]. La telerrehabilitación ha ganado relevancia como modalidad complementaria que amplía el acceso al tratamiento para pacientes con movilidad reducida o en áreas alejadas de centros especializados [33].

Incorporar objetos físicos instrumentados en estos modelos distribuidos enriquece el cuadro de seguimiento clínico con datos cuantitativos sobre la manipulación real del paciente como la información háptica y táctil de valor diagnóstico directo que los sistemas basados exclusivamente en RV no pueden capturar.

3.6. SÍNTESIS DEL ESTADO DEL ARTE Y POSICIONAMIENTO DEL TRABAJO

La evidencia clínica sobre la eficacia de las intervenciones XR en neurorrehabilitación es sólida. Las revisiones de Lee et al. [22], Leong et al. [24], Karamians et al. [18] y Laver et al. [20] convergen en que las intervenciones VAMR producen mejoras significativas en la función del miembro superior, con especial robustez cuando la tecnología se emplea como complemento que amplía el volumen total de práctica terapéutica. Lee et al. y Karamians et al. constatan una superioridad directa frente a la terapia convencional en parámetros motores del miembro superior. Laver et al. precisan que esa superioridad es máxima cuando la RV actúa como terapia adjunta: en esa condición obtienen mejoras estadísticamente significativas tanto en función del miembro superior como en actividades de la vida diaria, consolidando el modelo de integración complementaria como la estrategia con mayor respaldo empírico. Leong et al. refuerzan esta lectura señalando ventajas más pronunciadas en los grupos de mayor intensidad de tratamiento. La trayectoria del campo ha seguido una evolución clara: desde sistemas semi-inmersivos con sensores externos hacia visores autónomos completamente inmersivos, y desde la interacción en el aire hacia la manipulación de objetos físicos reales en entornos de realidad mixta.

Sin embargo, la brecha de fidelidad señalada por Leong et al. [24] persiste. El análisis de la literatura permite articularla en tres dimensiones que, hasta donde alcanza la revisión realizada, no han sido resueltas de forma integrada. La primera es la **oclusión visual durante la manipulación**: aunque D'Ippolito y Massaro [11] y Hunt et al. [16] señalan la fusión visuo-inercial como solución, su implementación cuantitativa en visores de RM autónomos con restricciones de procesamiento no ha sido validada en escenarios de rehabilitación. La segunda es la **detección fiable del agarre en pacientes neurológicos** mediante señales táctiles en lugar de heurísticas visuales: aunque Lim et al. [25] demuestran que una matriz de presión directamente aplicada sobre el objeto permite clasificar el patrón de agarre con alta precisión, esta información táctil no se integra con ningún sistema de seguimiento visual ni con un entorno de realidad mixta. La tercera es la **ausencia de arquitecturas integradas de objeto inteligente de bajo coste para RM**: los trabajos de Baccinelli et al. [4], Monge y Postolache [30] y Lim et al. [25] abordan componentes aislados del problema sin proporcionar una solución operable sobre un visor de RM autónomo que combine seguimiento inercial, validación táctil del agarre y detección visual en tiempo real.

3.6.1. Posicionamiento del presente trabajo

La Tabla 3.1 sintetiza el posicionamiento de esta propuesta frente a los trabajos más relevantes en cinco dimensiones: interacción tangible con objeto físico real (Tang.), robustez ante oclusión visual (Anti-ocl.), detección fiable del agarre por medios no visuales (Agarre), integración con un visor de realidad mixta comercial (RM) y viabilidad de bajo coste para uso clínico (Bajo coste).

Ninguno de los trabajos revisados cubre de forma simultánea las cinco dimensiones. Este TFG aspira a cubrir esa combinación de interacción tangible, robustez ante oclusión, validación táctil del agarre, integración con un visor de RM autónomo comercial (Meta Quest 3) y viabilidad de bajo coste, en un único sistema con validación experimental cuantitativa.

Tabla 3.1: Comparativa del presente trabajo con los principales enfoques del estado del arte. ✓: dimensión cubierta; ×: no cubierta; ~: cobertura parcial.

Trabajo	Tang.	Anti-ocl.	Agarre	MR	Bajo coste
Baccinelli et al. [4]	✓	×	~	×	✓
Monge y Postolache [30]	✓	×	×	×	✓
Lee et al. [21]	✓	×	✓	×	×
Zhang et al. [37]	✓	×	✓	×	×
Lim et al. [25]	✓	×	✓	×	✓
D’Ippolito y Massaro [11]	×	~	×	~	✓
Pillai et al. [32]	×	×	×	✓	✓
Herrera et al. [13]	×	×	×	✓	✓
Este trabajo	✓	✓	✓	✓	✓

Para el grupo AIR y el HNP, este trabajo continúa una trayectoria tecnológica que incluye el guante CyberGlove del proyecto Naviga [35], la telerrehabilitación de TOYRA [12], la plataforma RehabHand sobre Leap Motion Controller [8] y el framework Rehab-Immersive [15], llegando ahora a la realidad mixta con interacción tangible. La aportación del proyecto es técnica y metodológica: un sistema de seguimiento híbrido visuo-inercial, funcional con componentes de bajo coste sobre un visor comercial y replicable. Sobre esta base, el grupo podrá desarrollar otros objetos terapéuticos instrumentados y adaptados a distintas actividades de la vida diaria y perfiles clínicos.

CAPÍTULO 4

Metodología

En este capítulo se describen la metodología adoptada para el desarrollo del presente proyecto, la distribución del trabajo en paquetes y tareas, la planificación temporal seguida y los recursos empleados. Asimismo, se incluye una estimación de los costes asociados al proyecto.

4.1. METODOLOGÍA ELEGIDA

La metodología seguida es el **desarrollo iterativo e incremental**, un enfoque ágil en el que el ciclo de vida del proyecto se divide en una secuencia de iteraciones de corta duración. Cada iteración atraviesa las fases de análisis y diseño, implementación, pruebas y evaluación, obteniendo un incremento funcional que se integra progresivamente en el sistema global. Los resultados de cada iteración alimentan la planificación de la siguiente.

La elección responde a dos razones. En primer lugar, el carácter experimental de varios componentes hacía inviable una planificación rígida: la selección del hardware, la búsqueda de una solución BLE compatible con las Meta Quest 3, o la decisión de cambiar el sistema de calibración por QR por uno basado en *keypoints*, fueron decisiones cuya solución óptima solo pudo determinarse mediante exploración progresiva. En segundo lugar, el proyecto debía compatibilizarse con las responsabilidades académicas del autor; la organización en iteraciones facilitó concentrar las tareas de mayor carga técnica en los períodos de menor exigencia lectiva.

4.2. DISTRIBUCIÓN DEL TRABAJO

La carga de trabajo se ha organizado en siete paquetes de trabajo (PT), cada uno compuesto por un conjunto de tareas relacionadas:

1. Estudio inicial y requisitos
2. Hardware y firmware (IMU/BLE)
3. Módulo de visión por computador
4. Módulo de fusión sensorial
5. Juego serio y aplicación de realidad mixta
6. Validación y pruebas
7. Documentación y redacción de artículo científico

4.2.1. Paquetes de trabajo

PT1 — Estudio inicial y requisitos Agrupa las tareas de documentación y análisis del contexto clínico, realizadas en colaboración con el tutor.

T1.1 **Revisión del estado del arte.** Estudio de sistemas de rehabilitación basados en realidad mixta y objetos instrumentados.

T1.2 **Definición de requisitos del sistema.** Documentación de requisitos funcionales y no funcionales: parámetros de movimiento, restricciones ergonómicas y criterios de integración con el entorno inmersivo.

PT2 — Hardware y firmware (IMU/BLE) Abarca desde la selección y adquisición del hardware hasta el diseño de su integración física en el objeto terapéutico y el desarrollo del firmware embebido:

T2.1 **Evaluación y selección del hardware de sensorización.** Análisis comparativo de IMU y microcontroladores con BLE.

T2.2 **Adquisición y verificación de componentes.**

T2.3 **Diseño de la integración física.** Incorporación del microcontrolador y la IMU en el objeto terapéutico sin comprometer su ergonomía.

T2.4 **Desarrollo del firmware.** Implementación bajo Zephyr OS para lectura de la IMU, empaquetado de datos y transmisión vía BLE.

T2.5 **Identificación e integración de la solución BLE.** Búsqueda de una solución compatible con Unity y las Meta Quest 3 sin dispositivo intermediario.

T2.6 **Validación de la comunicación** extremo a extremo en tiempo real.

T2.7 **Incorporación del magnetómetro.** Evaluación de un magnetómetro externo para corregir la deriva en la estimación de orientación; finalmente descartado por su sensibilidad a interferencias electromagnéticas.

PT3 — Módulo de visión por computador

T3.1 **Selección del modelo de detección.** Evaluación de alternativas de inferencia y selección de YOLO nano por su equilibrio entre precisión y coste computacional.

T3.2 **Construcción del dataset y entrenamiento.** Grabación de vídeos con las propias gafas, etiquetado y entrenamiento del modelo.

T3.3 **Integración y evaluación.** Integración en Unity mediante Unity Sentis y evaluación ante distintas condiciones de iluminación, distancia y orientación.

T3.4 **Reescritura sobre las *AI Building Blocks*.** Reimplementación sobre la nueva capa de abstracción de Meta, manteniendo el modelo YOLO y Unity Sentis como motor de inferencia.

T3.5 **Sustitución del modelo YOLOv11-nano por YOLOv8-nano Pose.** Permite la calibración de la orientación mediante *keypoints*.

PT4 — Módulo de fusión sensorial Aborda el diseño e implementación del módulo que combina datos inerciales y detección visual para un seguimiento robusto del objeto incluso en condiciones de oclusión:

T4.1 **Diseño de la estrategia de fusión.** Selección del enfoque algorítmico y diseño de la arquitectura del módulo.

T4.2 **Implementación e integración** con los módulos de visión e inercial.

- T4.3 **Evaluación, revisión y validación.** Reformulación del enfoque a partir de los resultados de la primera versión y validación de la definitiva.
- T4.4 **Inclusión de controles de robustez y usabilidad.** Filtrado de falsos agarres, detecciones espurias, errores de color y posiciones no válidas.

PT5 — Juego serio y aplicación de realidad mixta

- T5.1 **Diseño del escenario de rehabilitación** y ejercicios funcionales, tomando como referencia las necesidades clínicas del PT1.
- T5.2 **Implementación de la lógica del juego serio**, incluyendo detección de eventos de manipulación y registro de métricas terapéuticas.
- T5.3 **Integración del módulo de fusión sensorial** como fuente de posición y orientación, cerrando la cadena completa del sistema.
- T5.4 **Revisión y ampliación del juego serio.** Ajuste de la dinámica de interacción, flujo de sesiones, registro de datos experimentales y elementos de control necesarios para su uso en pruebas reales.

PT6 — Validación y pruebas

- T6.1 **Diseño del protocolo de pruebas** para verificar continuidad del seguimiento, estabilidad BLE y coherencia espacial.
- T6.2 **Ejecución de pruebas y análisis de resultados.**
- T6.3 **Corrección de errores y verificación** del sistema.
- T6.4 **Pruebas en entorno clínico.** Realización de sesiones preliminares en el Hospital Nacional de Paraplégicos de Toledo para observar el comportamiento del sistema en un contexto real de uso.

PT7 — Documentación y redacción de artículo científico

- T7.1 **Redacción de la memoria del TFG.**
- T7.2 **Redacción y envío del artículo** al congreso XR Salento 2026.

4.2.2. Iteraciones

El proyecto se desarrolló en doce iteraciones entre julio de 2025 y la primera semana de junio de 2026. La secuencia estuvo determinada por las dependencias entre módulos: primero se abordaron la definición de requisitos, la selección del hardware, el firmware y la comunicación BLE; posteriormente, el módulo de visión y la fusión visuo-inercial; y, finalmente, la integración en el juego serio, la validación técnica, las pruebas preliminares en entorno clínico y la redacción de la memoria. Las revisiones principales se produjeron en enero, con la adaptación a las *AI Building Blocks* de Meta; en marzo, con la sustitución de la calibración por QR por *keypoints* y la revisión del algoritmo de fusión; en abril, con la ampliación del juego serio; en mayo, con las pruebas en el Hospital Nacional de Paraplégicos de Toledo; y en la primera semana de junio, con el cierre de la memoria previo a la presentación de resultados en el congreso XR Salento 2026.

Tabla 4.1: Resumen de las iteraciones del proyecto.

Iter.	Período y duración	Paquetes y tareas	Objetivos alcanzados
1	Julio 2025, 1 semana	PT1: T1.1, T1.2	Requisitos del sistema definidos y revisión inicial del estado del arte.
2	Septiembre 2025, 3 semanas	PT2: T2.1, T2.2, T2.3	Hardware seleccionado y adquirido. Entornos de desarrollo configurados. Diseño físico del objeto instrumentado definido.
3	Octubre 2025, 4 semanas	PT2: T2.4, T2.5, T2.6	Firmware operativo bajo Zephyr OS y comunicación BLE funcional extremo a extremo con Meta Quest 3.
4	Noviembre 2025, 3 semanas	PT3: T3.1, T3.2, T3.3; PT2: T2.6	Módulo de visión implementado y evaluado. Comunicación BLE validada con datos reales de movimiento.
5	Diciembre 2025, 3 semanas	PT1: T1.1; PT4: T4.1, T4.2; PT2: T2.7, T2.4	Primera versión del módulo de fusión implementada. Revisión de técnicas de fusión sensorial y magnetometría. Magnetómetro incorporado y firmware actualizado para su lectura.
6	Enero 2026, 4 semanas	PT1: T1.1; PT2: T2.7, T2.4; PT4: T4.1, T4.2, T4.3	Magnetómetro evaluado y descartado por sensibilidad a interferencias electromagnéticas. Firmware restaurado a la versión sin magnetómetro. Módulo de fusión sensorial revisado, implementado y validado.
7	Enero–Febrero 2026, 2 semanas	PT2: T2.3; PT3: T3.4; PT7: T7.2	Rediseño de la integración física para adaptarla a las <i>AI Building Blocks</i> . Módulo de visión reescrito sobre las herramientas de Meta. Inicio de la redacción del artículo científico XR Salento 2026.
8	Febrero 2026, 4 semanas	PT5: T5.1, T5.2, T5.3; PT6: T6.1, T6.2, T6.3; PT7: T7.2	Juego serio implementado e integrado con el módulo de fusión. Sistema validado técnicamente y errores corregidos. Artículo científico concluido y enviado al congreso XR Salento 2026.
9	Marzo 2026, 4 semanas	PT3: T3.5; PT4: T4.3, T4.4	Modelo de visión sustituido por YOLOv8-nano Pose. Calibración por <i>keypoints</i> operativa. Algoritmo de fusión revisado. Filtros y controles de usabilidad implementados frente a falsas detecciones, errores de color y posiciones no válidas.
10	Abril 2026, 4 semanas	PT5: T5.1, T5.2, T5.3, T5.4; PT6: T6.2, T6.3	Juego serio ampliado y adaptado para sesiones experimentales. Flujo de interacción, eventos, registro de datos en CSV y soporte multiobjeto revisados. Pruebas técnicas repetidas sobre la versión evolucionada del sistema.

Continúa en la página siguiente.

Iter.	Período y duración	Paquetes y tareas	Objetivos alcanzados
11	Mayo 2026, 1 semana	PT6: T6.1, T6.2, T6.4; PT5: T5.4	Protocolo de pruebas aplicado en el Hospital Nacional de Paraplégicos de Toledo. Sesiones preliminares ejecutadas y datos experimentales registrados. Ajustes menores derivados de la preparación y ejecución de las pruebas clínicas.
12	Junio 2026, 1 semana	PT1: T1.1; PT6: T6.2; PT7: T7.1	Estado del arte actualizado, resultados integrados en el análisis final y memoria del TFG redactada, revisada y concluida en la primera semana de junio de cara a su posterior tramitación y entrega formal en julio.

A continuación se resume la evolución del trabajo a lo largo de las iteraciones. La primera iteración constituyó la fase inicial del proyecto y se dedicó principalmente a la revisión del estado del arte y a la definición de requisitos en colaboración con el tutor. Esta etapa permitió delimitar el problema técnico central: la pérdida de seguimiento visual durante la manipulación de objetos físicos en entornos de realidad mixta.

Las iteraciones segunda y tercera se centraron en la selección del hardware, la configuración del entorno de desarrollo y la puesta en marcha de la comunicación entre el objeto instrumentado y el visor. En esta fase se configuraron Unity, Meta Quest 3 y Zephyr OS, se definió la integración física del objeto terapéutico y se desarrolló el firmware inicial. La comunicación BLE se validó extremo a extremo, adoptando finalmente un plugin de la Asset Store tras descartar una biblioteca no mantenida.

La cuarta iteración abordó el desarrollo del módulo de visión. Para ello se generó un conjunto de datos propio grabado con el visor, se entrenó un modelo YOLOv11-nano y se integró en Unity Sentis. Las pruebas se realizaron bajo distintas condiciones de iluminación, distancia y orientación, lo que permitió evaluar las primeras limitaciones del seguimiento visual.

Durante la quinta y sexta iteración se diseñó y revisó el módulo de fusión visuo-inercial. La primera versión permitió identificar limitaciones en la estimación de orientación, lo que motivó la incorporación experimental de un magnetómetro. Sin embargo, su evaluación mostró una sensibilidad elevada a interferencias electromagnéticas, por lo que fue descartado. A partir de ese punto, el sistema se reorientó hacia una solución basada en visión, IMU y validación táctil mediante FSR.

La séptima iteración supuso una reestructuración del módulo de visión sobre las *AI Building Blocks* de Meta, manteniendo YOLO y Unity Sentis como base del procesamiento visual. En esta fase también se incorporó una primera estrategia de calibración de *yaw* por QR y se inició la redacción del artículo científico para XR Salento 2026.

La octava iteración se dedicó al diseño e implementación del juego serio, a la integración del módulo de fusión en la dinámica de interacción y a la ejecución del protocolo de validación técnica. Esta fase permitió disponer de una primera versión funcional del sistema completo y culminó con el envío del artículo científico al congreso XR Salento 2026.

La novena iteración incorporó correcciones derivadas del uso del juego serio. Se substituyó la calibración mediante QR por una estrategia basada en *keypoints* con YOLOv8-nano Pose, se revisó el comportamiento del algoritmo de fusión durante oclusiones y se añadieron filtros para reducir falsas detecciones, errores de color y posiciones no válidas.

La décima iteración amplió el juego serio para su uso en sesiones experimentales. Se revisó el flujo de actividad, se añadieron eventos de interacción, se incorporó el registro de datos en CSV, se mejoró el soporte multiobjeto y se repitieron pruebas técnicas sobre la versión evolucionada del sistema.

La undécima iteración correspondió a las pruebas preliminares realizadas en el Hospital Nacional de Paraplégicos de Toledo. Estas pruebas permitieron comprobar la aplicación, el seguimiento, el registro de datos y el flujo del juego serio en un entorno clínico real. Además, proporcionaron información relevante para detectar limitaciones prácticas del sistema y realizar ajustes derivados del uso con pacientes.

Finalmente, la duodécima iteración se dedicó a la redacción y revisión final de la memoria, integrando los resultados técnicos y experimentales obtenidos durante el desarrollo y concluyendo el documento de forma definitiva antes de la asistencia al congreso.

4.2.3. Diagrama de Gantt

La Figura 4.1 resume la planificación temporal del proyecto: dependencias secuenciales entre módulos, solapamiento en enero entre la revisión del módulo de fusión y la adaptación del módulo de visión, revisión del algoritmo de fusión en marzo, ampliación del juego serio en abril, pruebas en el Hospital Nacional de Paraplégicos de Toledo en mayo y cierre de la memoria a comienzos de junio.

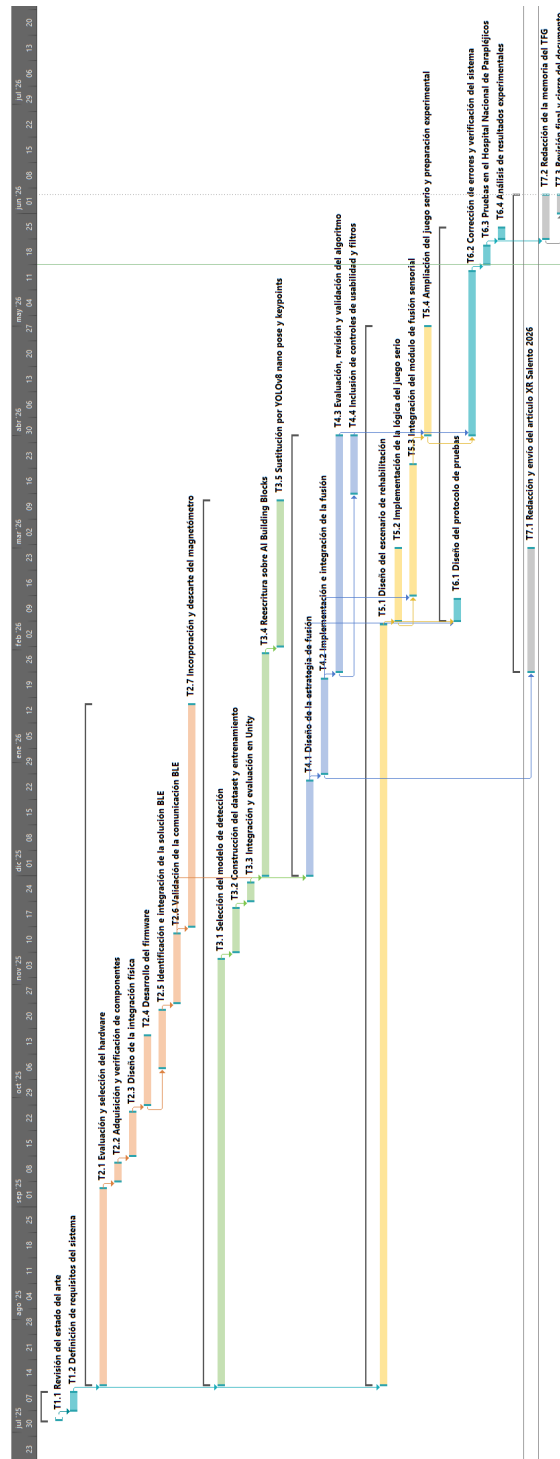


Figura 4.1: Diagrama de Gantt del proyecto.

4.3. RECURSOS EMPLEADOS

4.3.1. Hardware

- **Meta Quest 3.** Dispositivo de visión RV/RA/RM para la ejecución de la aplicación de realidad mixta.
- **Portátil MSI.** Equipo de desarrollo cedido por el grupo AIR para programación, compilación y despliegue.
- **Seeed Studio XIAO nRF52840 Sense.** Microcontrolador con BLE integrado e IMU incorporada, responsable de la adquisición y transmisión de datos inerciales.
- **Sensor FSR Teksan A401-25.** Sensor de fuerza resistivo para el registro de la fuerza aplicada sobre el objeto terapéutico.
- **Nordic Power Profiler Kit 2.** Analizador de consumo energético del sistema embebido.

4.3.2. Software

Sistemas operativos

- **Windows 11.** Entorno principal: compilación y despliegue en Meta Quest 3.
- **Ubuntu.** Desarrollo del firmware bajo Zephyr OS.

Entornos y herramientas

- **Unity.** Motor de desarrollo de la aplicación de realidad mixta.
- **Visual Studio.** IDE principal para scripting en C# y depuración.
- **Visual Studio Code.** Desarrollo del firmware embebido.
- **Meta Quest Link / Meta Developer Hub.** Despliegue, gestión y monitorización del dispositivo.
- **Git.** Control de versiones del firmware.
- **Overleaf.** Redacción de la memoria y el artículo en $\text{\LaTeX}$.
- **Draw.io.** Elaboración de esquemas e ilustraciones.

SDKs, plugins y librerías

- **Meta XR All-in-One SDK.** Framework oficial para experiencias inmersivas en Quest 3.
- **Mixed Reality Utility Kit.** Componentes de anclaje y comprensión de escena real.
- **Unity Sentis.** Motor de inferencia de redes neuronales en Unity.
- **AI Building Blocks (Meta).** Capa de abstracción para integración de modelos de inferencia en Quest 3.
- **BLE for iOS, tvOS and Android.** Plugin de la Asset Store para comunicación BLE con las Quest 3.

Frameworks y lenguajes

- **Zephyr OS.** Sistema operativo de tiempo real para el microcontrolador.
- **C#.** Lógica de la aplicación Unity.
- **C.** Firmware del microcontrolador.

- **LaTeX**. Redacción de la memoria y el artículo científico.

4.4. ESTIMACIÓN DE COSTES

La Tabla 4.2 detalla los componentes adquiridos para la fabricación de los tres prototipos del objeto terapéutico instrumentado, incluyendo el material de impresión 3D para las carcasas. La Tabla 4.3 recoge el desglose económico global del proyecto. Los costes de personal contemplan al autor del TFG con dedicación de cuatro horas diarias durante seis meses (583 €/mes, referencia Glassdoor para desarrollador junior a jornada parcial) y a un tutor académico con dedicación de una hora diaria (547 €/mes sobre una base de 35.000 €/año para ingeniero sénior).

Tabla 4.2: Materiales del prototipo

Componente	Cant.	Precio unit. (€)	Total (€)
Seed Studio XIAO nRF52840 Sense	3	16,40	49,20
Sensor FSR Teksan A401-25 mm	3	19,38	58,14
Batería LiPo 502020 3,7 V 200 mAh	3	12,00	36,00
Placa preperforada	1	10,00	10,00
Cables	1	9,00	9,00
Filamento PLA (10 g × 3 uds.)	1	0,60	0,60
Total prototipo			162,94

Tabla 4.3: Resumen económico del proyecto

Concepto	Coste (€)
Material directo del prototipo	162,94
Material indirecto	782,72
Material no incorporado	81,52
Desarrollador (autor del TFG)	3.498,00
Tutorización del proyecto	3.282,00
Energético	6,24
Coste total	7.813,42

Arquitectura del sistema

5.1. RETOS Y PROBLEMÁTICAS

Antes de describir la arquitectura del sistema, conviene recalcar las problemáticas técnicas que condicionan su diseño. Estas problemáticas justifican las decisiones tomadas a lo largo del capítulo y explican por qué la arquitectura requiere la combinación coordinada de múltiples fuentes de información.

- P1. Oclusión visual durante el agarre.** Cuando el paciente agarra un objeto terapéutico, su mano lo cubre parcial o totalmente, impidiendo que las cámaras del visor mantengan la referencia visual. Esta situación se produce precisamente en el momento más relevante del ejercicio, ya que es durante la manipulación activa cuando se generan los datos cinemáticos de mayor interés clínico. Los sistemas de seguimiento exclusivamente ópticos pierden el objeto en este instante, interrumpiendo la experiencia inmersiva y comprometiendo la fiabilidad de los datos registrados.
- P2. Pérdida de información rotacional sin referencia magnética.** Los sensores inerciales de 6 ejes proporcionan una estimación robusta de la inclinación (*pitch* y *roll*) gracias a la referencia gravitacional del acelerómetro. Sin embargo, carecen de referencia alguna para la rotación horizontal (*yaw*), que deriva inevitablemente con el tiempo. La inclusión de un magnetómetro, que en teoría aportaría esa referencia, se descartó tras la validación experimental por la inestabilidad que este provocaba. Es necesario, por tanto, un mecanismo alternativo de corrección periódica del *yaw* que no dependa del campo magnético terrestre. Esta limitación se asume como parte del compromiso entre coste, tamaño, consumo y robustez del prototipo.
- P3. Necesidad de fusión multimodal.** Ninguna fuente de información individual es suficiente por sí sola: la visión falla durante la oclusión, la IMU no puede proporcionar posición absoluta y su *yaw* deriva, y el sensor de presión solo indica si hay contacto, pero no dónde está el objeto. Solo la integración coordinada de las tres fuentes permite mantener una estimación continua y robusta de la pose del objeto durante toda la interacción.
- P4. Falsos positivos en la detección del agarre.** En pacientes con lesión medular o ictus, la postura de reposo en semiflexión de la mano puede ser interpretada erróneamente por los sistemas visuales como un agarre activo, generando falsas adhesiones del objeto virtual a la mano. De forma complementaria, el modelo de detección visual puede confundir el puño del paciente con un objeto terapéutico, produciendo falsas detecciones. Ambos artefactos comprometen la fiabilidad del sistema en condiciones reales de uso clínico.

P5. Asociación de identidad con múltiples objetos. Cuando coexisten varios objetos instrumentados en la escena, el sistema debe asociar de forma determinista cada detección visual con su sensor inercial correspondiente. Una asociación errónea haría que el gemelo digital de un objeto siguiese los movimientos del sensor de otro, invalidando completamente el seguimiento.

P6. Registro de datos cinemáticos para evaluación clínica. La práctica habitual en rehabilitación se limita a medir tiempos de ejecución o conteos de repeticiones. Instrumentar los objetos abre la posibilidad de registrar métricas sobre la *calidad* del movimiento, como la estabilidad del agarre, suavidad de la trayectoria, velocidad y aceleración durante el transporte. Estos datos son completamente inaccesibles mediante observación directa. El sistema debe ser capaz de registrar y persistir estos datos de forma estructurada para su consulta posterior por parte del equipo clínico.

Los requisitos derivados de estas problemáticas se detallan en la Sección 5.3, y la visión general de la solución propuesta se presenta a continuación.

5.2. VISIÓN GENERAL

El sistema desarrollado en este trabajo se articula en torno a cinco módulos funcionales que operan de forma concurrente durante una sesión de rehabilitación.

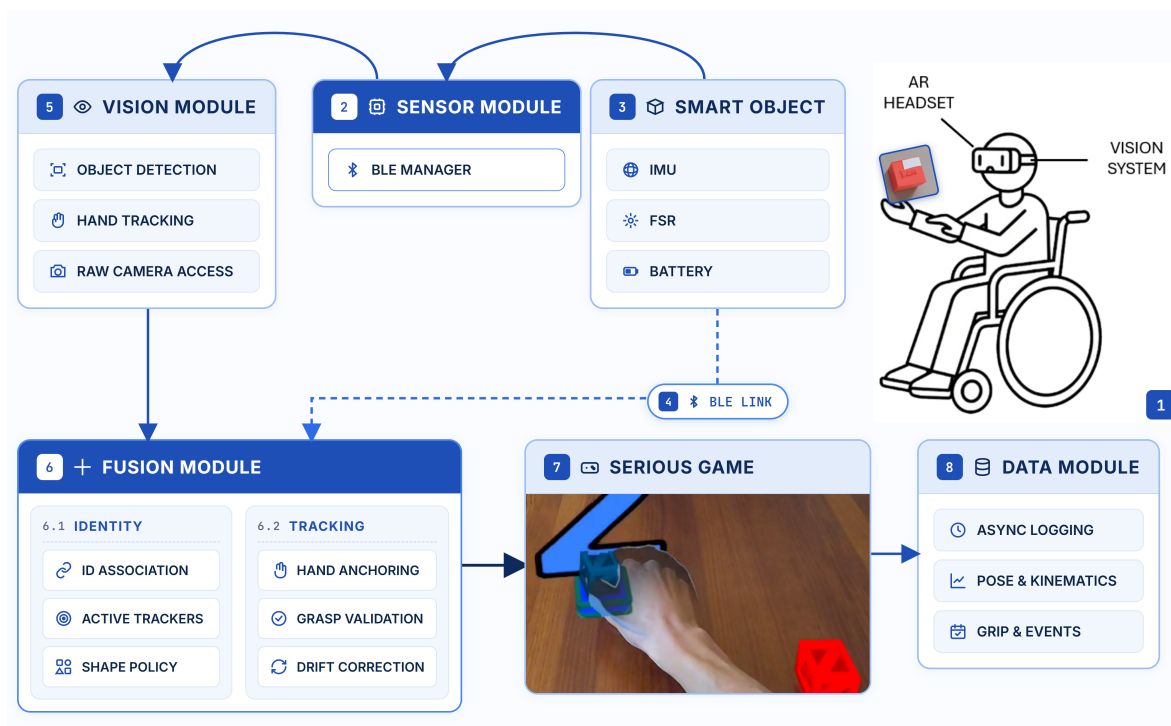


Figura 5.1: Arquitectura del sistema y flujo de datos extremo a extremo.

- El **objeto sensorizado**, responsable de la adquisición de datos inerciales, aceleración lineal cruda y presión embebidos en el objeto terapéutico, de la fusión local de orientación mediante el filtro de Madgwick y de la transmisión inalámbrica al visor.
- El **módulo de visión**, encargado de la detección óptica del objeto mediante un modelo YOLOv8-Pose, de la proyección de las detecciones 2D a coordenadas 3D mediante el mapa de profundidad del visor, y del seguimiento de las manos del paciente.

- El **módulo de fusión**, que integra ambas fuentes de información para mantener una estimación robusta de la posición 6-DoF del objeto en el espacio de coordenadas del visor. Además de gestionar la calibración de orientación y otros controles necesarios para lograr la suficiente precisión. Este módulo incluye una arquitectura extensible (ShapePolicy) que permite incorporar nuevos tipos de objetos de la vida diaria.
- El **módulo de juego serio**, que consume la pose resultante para ofrecer retroalimentación gamificada al paciente y estructurar la sesión terapéutica. La mecánica principal es un juego de persecución que requiere la manipulación activa de los objetos bajo presión temporal progresiva.
- El **módulo de registro de datos**, que obtiene de forma continua y asíncrona la información cinemática, sensorial y de eventos de cada sesión, registrándola en archivos CSV estructurados, incluyendo la segmentación automática de las interacciones del paciente con los objetos y la clasificación del tipo de suelta. Este módulo permite al equipo clínico consultar *a posteriori* métricas objetivas sobre la calidad del movimiento del paciente.

El flujo de información parte del paciente y termina en la capa de aplicación. El paciente manipula el objeto terapéutico físico, que lleva integrado el hardware de sensorización. Por un lado, la unidad de medida inercial (IMU) registra de forma continua la orientación del objeto, fusionando ambas medidas mediante el filtro de Madgwick para producir un cuaternión que codifica la rotación tridimensional. Además, el firmware incorpora en la telemetría una medida cruda de aceleración lineal, utilizada por el módulo de fusión para identificar movimiento del objeto incluso cuando la variación angular es reducida. Por otro lado, un sensor de fuerza resistivo (FSR) adherido a la superficie del cubo monitoriza la presión que ejerce la mano del paciente, permitiendo discriminar cuándo existe un agarre confirmado. Toda esa información se transmite de forma inalámbrica al visor *Meta Quest 3* mediante *Bluetooth Low Energy* (BLE).

Al mismo tiempo, el módulo de visión analiza los fotogramas de la cámara *passthrough*¹ del visor. Sobre esos fotogramas, un modelo YOLOv8-Pose nano devuelve la identidad del objeto, su posición bidimensional en la imagen y hasta cuatro *keypoints* correspondientes a los vértices de la cara superior del cubo. Esas coordenadas 2D se proyectan a 3D mediante el mapa de profundidad que proporciona el visor, una textura en la que cada píxel codifica la distancia al objeto más cercano en esa dirección, y se clasifican por color para vincular cada detección visual a su sensor BLE correspondiente.

El módulo de fusión toma esas dos fuentes visual e inercial y las combina para mantener una estimación coherente de la pose del gemelo digital asociado. El gemelo digital es una réplica virtual del objeto físico cuya posición y orientación se actualizan en tiempo real para que coincida con las del cubo real. El resultado de la fusión es especialmente relevante durante los periodos de oclusión, cuando la mano cubre el objeto y la cámara deja de verlo. En ese intervalo, la IMU mantiene la orientación y aporta evidencia de movimiento, mientras que la traslación se propaga mediante el vínculo con la mano y se corrige cuando vuelve la referencia visual. La pose resultante se entrega al juego serio, que la usa para ofrecer retroalimentación inmersiva al paciente.

¹El término *passthrough* se refiere a la funcionalidad que permite al visor de realidad mixta capturar el entorno físico con sus cámaras y mostrarlo al usuario como fondo, sobre el que se superponen los elementos virtuales.

5.3. REQUISITOS CLÍNICOS

El punto de partida del diseño es el contexto clínico en el que el sistema va a operar. La recuperación motora del miembro superior se trabaja habitualmente a través de tareas funcionales que implican la manipulación física de objetos reales. El ejemplo de referencia en neurorrehabilitación es el *Box and Block Test* (BBT) [29], una prueba estandarizada para la evaluación de la destreza manual unilateral. En este protocolo, el paciente alcanza, transporta y suelta bloques cúbicos de forma repetida, poniendo en juego toda la cadena cinética: desde la estabilidad proximal en hombro y codo hasta la orientación de muñeca y el control fino del agarre.

En la práctica clínica habitual, la evaluación de estas tareas se limita al tiempo de ejecución o al número de repeticiones completadas. Instrumentar los propios objetos abre la posibilidad de capturar métricas sobre la *calidad* del movimiento como estabilidad del agarre o suavidad de la trayectoria que son completamente inaccesibles mediante la observación directa, evitando además la instrumentalización del paciente con sensores corporales que puedan resultar incómodos o intimidantes.

De este contexto se derivan cuatro requisitos que han marcado las decisiones de diseño a lo largo del proyecto:

1. **Telemetría en tiempo real y fusión multimodal.** El objeto debe transmitir datos de forma continua y con baja latencia, de modo que la información inercial pueda combinarse con la visual para mantener la coherencia espacial incluso cuando el objeto queda completamente oculto por la mano. Esto descarta soluciones con latencias altas como WiFi o almacenamiento local con descarga posterior.
2. **Cuantificación del agarre.** Dado que en manos paréticas la intención de agarre puede manifestarse antes de que el objeto sea levantado, el sistema debe ser capaz de detectarla mediante presión. Esto permite además registrar la evolución de la fuerza y la modulación motora fina a lo largo del tratamiento.
3. **Ergonomía y transparencia.** La electrónica embebida y la batería deben tener una huella mínima para no alterar el centro de gravedad ni las propiedades hápticas del objeto, y la autonomía debe ser suficiente para cubrir sesiones clínicas completas sin interrupciones por recarga. El paciente no debería percibir diferencia significativa entre manipular un cubo instrumentado y un cubo convencional.
4. **Registro y persistencia de datos clínicos.** El sistema debe almacenar de forma automática las métricas cinemáticas y de agarre de cada sesión para que el terapeuta pueda consultarlas *a posteriori*. Esto incluye la posición y orientación del objeto a lo largo del tiempo, la fuerza de agarre, la cinemática de la mano del paciente y los eventos del juego serio. Sin este registro, la información generada durante la sesión se perdería al finalizarla, desaprovechando el potencial de los objetos sensorizados como herramienta de evaluación objetiva.

5.4. PIPELINE DE DATOS EXTREMO A EXTREMO

La arquitectura completa queda recogida en la Figura 5.1, cuyos bloques se corresponden con las etapas del *pipeline*:

- (1) El paciente interactúa con el objeto terapéutico sensorizado.
- (2) El sistema embebido fusiona los datos inerciales mediante el filtro de Madgwick, conserva la aceleración lineal cruda como indicador de movimiento y lee el sensor de

presión.

- (3) Los datos se transmiten al visor mediante BLE.
- (4) El visor Meta Quest 3 recibe la telemetría de orientación, aceleración lineal cruda y presión a través del componente BLEManager.
- (5) Simultáneamente, el agente YoloPoseAgent ejecuta la inferencia del modelo YOLO sobre los fotogramas de la cámara *passthrough*, y el visor realiza el seguimiento nativo de las manos del paciente.
- (6) Los datos se integran en dos etapas coordinadas:
 - (6.1) Las detecciones visuales se vinculan de forma determinista a los sensores BLE correspondientes mediante una asociación jerárquica implementada en el FusionSystemManager. En primer lugar se utiliza la clase visual y la identidad de color extraída de la textura de la cámara. Si esta correspondencia no es suficiente por oclusión de la mano o pérdida temporal del color, se emplea un respaldo cinemático basado en la proximidad de la mano y en el único *tracker* compatible que informa de movimiento inercial. El enrutamiento soporta tanto objetos diferenciados por color (cubos) como objetos de clase única (taza) mediante la comparación de las clases visible y ocluida de la ShapePolicy.
 - (6.2) Cada FusionTracker individual combina la posición visual con la orientación inercial, gestiona la calibración de yaw mediante los *keypoints*, evalúa el nivel de contacto mano-cubo a través del CubeContactAssessor, y decide si el objeto debe anclarse localmente a la mano por FSR o propagarse mediante el puente inercial de oclusión basado en el desplazamiento de la mano.
- (7) La pose resultante es consumida por el juego serio, que proporciona retroalimentación gamificada al paciente mediante una mecánica de persecución temporal.
- (8) En paralelo a todo el flujo, el módulo de registro persiste de forma asíncrona los datos sensoriales, las detecciones visuales, los resultados de la fusión, la cinemática del paciente, el estado completo de cada cubo y los eventos del juego serio en archivos CSV estructurados. Además, segmenta automáticamente las interacciones del paciente para facilitar el análisis posterior.

Esta descomposición modular persigue dos objetivos fundamentales. El primero es la **independencia funcional**: cada módulo puede ser desarrollado, validado y actualizado de forma independiente, lo que facilita el mantenimiento y la extensibilidad del sistema. Por ejemplo, si en el futuro se desea reemplazar el modelo YOLO por una arquitectura más reciente, basta con modificar el agente de inferencia sin tocar el módulo de fusión ni el firmware. El segundo objetivo es la **tolerancia a fallos parciales**: si el canal visual falla temporalmente por oclusión, el módulo de fusión puede operar exclusivamente con datos inerciales hasta que la visión se recupere, sin interrumpir la experiencia del paciente.

5.5. DISEÑO DE OBJETOS INTELIGENTES

5.5.1. Selección de plataforma hardware

El diseño hardware del objeto instrumentado está condicionado por cuatro requisitos propios de los dispositivos de rehabilitación embebidos: eficiencia energética, robustez mecánica, comunicación en tiempo real y minimización del factor de forma. El núcleo hardware seleccionado es el módulo **Seeed Studio XIAO nRF52840 Sense**, un Sistema en Módulo (SoM) basado en el SoC Nordic Semiconductor nRF52840.

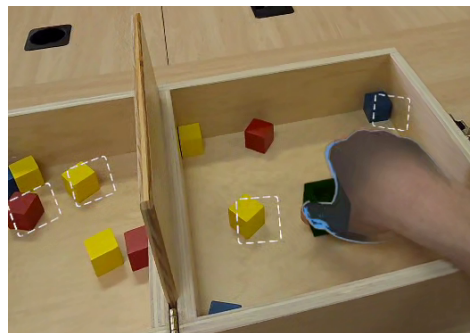


Figura 5.2: Representación virtual de los cubos del BBT superpuesta sobre el entorno físico real en el visor Meta Quest 3. El cubo de color apagado que se observa es el gemelo digital renderizado por el sistema de fusión, alineado con precisión sobre el cubo físico real que se encuentra debajo. La coincidencia espacial entre ambos es el resultado de la fusión visuo-inercial descrita en este capítulo.

La elección de esta plataforma frente a alternativas compactas como el Arduino Nano 33 BLE o la familia ESP32-C3 responde a la necesidad de integrar el mayor número de funciones posible en el menor espacio. Otras plataformas obligan a añadir módulos inerciales externos mediante cableado I²C o SPI, lo que aumenta el volumen del conjunto e introduce puntos de fallo mecánico como soldaduras, conectores o cables que, en un contexto de uso continuado con pacientes que pueden tener un control motor irregular, resultan problemáticos.

El módulo XIAO integra en una huella de tan solo $21 \times 17,8 \times 5$ mm los siguientes componentes relevantes para este proyecto:

- Procesador ARM Cortex-M4 a 64 MHz con unidad de coma flotante (FPU), necesaria para ejecutar las operaciones trigonométricas del filtro de Madgwick por hardware en lugar de emularlas por software.
- IMU STMicroelectronics **LSM6DSL** de 6 ejes (acelerómetro + giroscopio), conectada internamente por I²C.
- Radio BLE 5.0 con antena integrada.
- Circuito de gestión de carga para baterías LiPo.
- Conversor analógico-digital (ADC) SAADC de 12 bits con múltiples canales, utilizado para leer el sensor de presión y monitorizar el voltaje de la batería.

El prototipo actual se ha construido sobre una placa presoldada, sin necesidad de diseñar un PCB personalizado. Esta decisión, justificada por el carácter exploratorio del trabajo, permite además apreciar con claridad cuál es el tamaño mínimo real del sistema: el conjunto completo de placa, sensor FSR adherido a la superficie del objeto y batería puede reducirse tanto como permita la capacidad de la celda elegida.

Para el prototipo funcional, los cubos han sido rediseñados y fabricados mediante impresión 3D con el objetivo de alojar la electrónica sin modificar las propiedades hápticas ni la geometría de agarre, respetando la ergonomía del Box and Block Test.

5.5.2. Sensores y configuración de adquisición

5.5.2.1. Unidad de medida inercial (IMU)

La IMU LSM6DSL integrada en el SoM combina dos sensores complementarios:

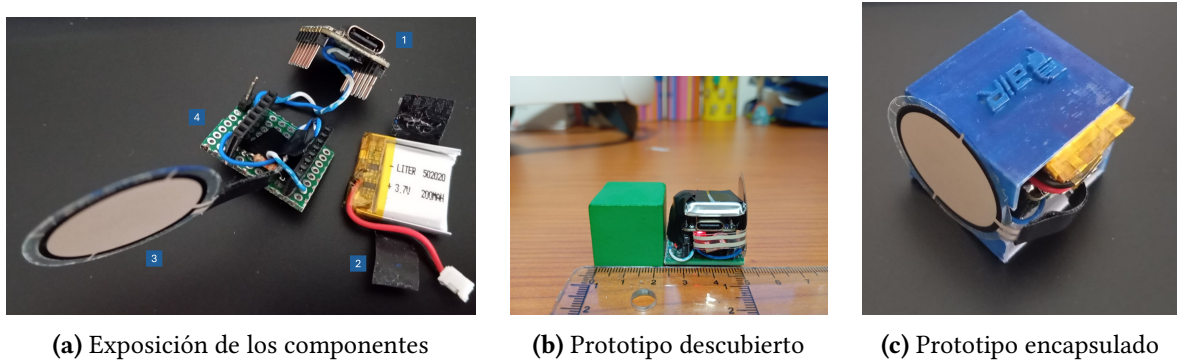


Figura 5.3: Componentes del prototipo hardware: electrónica expuesta (a), disposición interna del sensor (b) y prototipo encapsulado final (c).

- El **acelerómetro** mide la aceleración lineal a la que está sometido el dispositivo en tres ejes. Cuando el objeto está en reposo, la única aceleración presente es la de la gravedad, lo que permite al filtro de orientación determinar la vertical del mundo y corregir la inclinación del sensor (*pitch* y *roll*). Su limitación es que durante el movimiento la aceleración propia se mezcla con la gravitacional, haciendo la medida ruidosa.
- El **giroscopio** mide la velocidad angular del dispositivo en tres ejes. Al integrarse en el tiempo produce una estimación de la rotación acumulada. Su ventaja es que responde de forma inmediata a los giros, pero su limitación es la **deriva**: pequeños errores de medida se acumulan con cada integración y provocan que la orientación estimada se desvíe lentamente de la real.

La IMU se configura a una frecuencia de muestreo (ODR, *Output Data Rate*) de 104 Hz, que ofrece un buen compromiso entre resolución temporal y consumo energético. Durante el proceso de diseño se evaluó una configuración de 9 ejes mediante la adición de un magnetómetro externo. La idea era que el magnetómetro, al medir el campo magnético terrestre, proporcionase una referencia absoluta de la orientación horizontal (*yaw*) que ni el acelerómetro ni el giroscopio pueden estabilizar por sí solos. Sin embargo, la validación experimental reveló que los magnetómetros introducen inestabilidades significativas debidas a las interferencias magnéticas del entorno, más aún en un prototipo de tamaño tan reducido. En consecuencia, se adoptó la configuración robusta de 6 ejes, aceptando la deriva de *yaw* inherente y corrigiéndola periódicamente, y de forma transparente al usuario, mediante el algoritmo de calibración basado en *keypoints* descrito en la Sección 5.7.7, así como un algoritmo para simular el norte y poder intercambiar los movimientos del cubo en función del mismo.

5.5.2.2. Sensor de fuerza resistivo (FSR)

Para la detección del agarre, el prototipo integra un sensor de fuerza resistivo **FSR**. Este tipo de sensor es una lámina delgada cuya resistencia eléctrica disminuye al aplicarle presión: sin presión presenta una resistencia muy alta (megaohmios), y al presionarlo baja hasta los kilohmios. Para convertir esa variación de resistencia en una señal de voltaje legible por el ADC, el sensor se conecta en una configuración de **divisor de tensión** (*pull-down*) con una resistencia fija de 30 k Ω conectada a tierra. Cuando el paciente presiona el sensor, su resistencia disminuye, el voltaje en el punto medio del divisor sube y el ADC lo registra como un valor más alto. La resistencia de 30 k Ω se eligió específicamente para maximizar la sensibilidad en el rango inferior de fuerzas, lo que permite detectar la intención de agarre

incluso antes de que la presión sea suficiente para levantar el objeto. Esto resulta clínicamente valioso en pacientes con control motor reducido cuya fuerza de prensión es muy baja.

La lectura del FSR se realiza mediante el ADC SAADC del nRF52840 a una resolución de 12 bits (4096 niveles) con una **ganancia de 1**. La ganancia del ADC determina qué rango de voltaje se mapea al rango completo de la conversión: una ganancia mayor reduce el rango de entrada pero aumenta la sensibilidad a variaciones pequeñas. Con la referencia interna de voltaje del nRF52840 y ganancia 1, el rango de entrada efectivo es de aproximadamente 0–0,6 V, lo que proporciona una resolución de $\approx 0,3$ mV por nivel y resulta suficiente para discriminar presiones leves.

La interfaz ADC del microcontrolador soporta un *array* de sensores escalable. El número de sensores de presión depende de la geometría del objeto: herramientas direccionales como cubiertos pueden requerir un único sensor, mientras que objetos simétricos como los cubos del BBT pueden integrar varios sensores en distintas caras para detectar el agarre independientemente de la orientación con la que el paciente tome el objeto.

5.5.3. Firmware: filtro de orientación y calibración

5.5.3.1. Sistema operativo y configuración de compilación

El firmware se ha desarrollado sobre **Zephyr RTOS**, un sistema operativo en tiempo real de código abierto mantenido por la Linux Foundation y ampliamente utilizado en el ecosistema Nordic Semiconductor. Se descartó el ecosistema Arduino por su dependencia de capas de abstracción hardware de alto nivel (HAL) que ocultan el control directo sobre los registros del microcontrolador y dificultan garantizar el determinismo temporal.

Zephyr expone los *drivers* de los sensores y la pila BLE a bajo nivel, lo que permite ajustar las frecuencias de muestreo, las prioridades de las interrupciones y los modos de bajo consumo de forma explícita. La configuración de compilación del proyecto refleja las decisiones orientadas al ahorro energético: se desactivan todos los subsistemas de *logging*, la consola serie, la UART y la salida por `printf`. Estos subsistemas, útiles durante el desarrollo, consumen corriente de forma constante al mantener activos los periféricos de comunicación serie, y en producción no aportan ninguna funcionalidad al usuario. En contrapartida, se habilita la unidad de coma flotante para que las operaciones trigonométricas del filtro de Madgwick (raíces cuadradas, divisiones, productos vectoriales) se ejecuten por hardware en el coprocesador del Cortex-M4, reduciendo drásticamente el tiempo de CPU necesario por cada iteración del filtro.

5.5.3.2. Calibración de sensores en reposo

Todos los sensores inerciales presentan pequeños errores sistemáticos que varían de una unidad a otra y con la temperatura: el giroscopio puede reportar una velocidad angular distinta de cero aunque el dispositivo esté perfectamente quieto (*bias*), y el acelerómetro puede presentar un *offset* respecto al valor esperado de la gravedad. Si estos errores no se compensan, se acumulan rápidamente durante la integración del filtro y degradan la estimación de orientación.

Para corregirlos, el firmware ejecuta al arrancar una rutina de calibración automática. El procedimiento es el siguiente: se activa la IMU a 104 Hz, se esperan 2 segundos para que los sensores se estabilicen, y a continuación se toman 200 muestras consecutivas (separadas 5 ms entre sí). De esas 200 muestras se calcula la media aritmética de los tres ejes del giroscopio, que se almacena como *bias* estático ($\mathbf{b}_g$), y la media del acelerómetro, a la que se le resta la

componente gravitacional esperada ($9,80665 \text{ m/s}^2$ en el eje vertical) para obtener el *offset* ($\mathbf{b}_a$). Durante la operación normal, cada lectura de los sensores se corrige restándole estos valores:

$$\boldsymbol{\omega}_{corr} = \boldsymbol{\omega}_{raw} - \mathbf{b}_g, \quad \mathbf{a}_{corr} = \mathbf{a}_{raw} - \mathbf{b}_a \quad (5.1)$$

De forma análoga, el sensor FSR se calibra al arrancar tomando 50 muestras del ADC en reposo (sin presión aplicada) y almacenando su media como *offset* de cero. Todas las lecturas posteriores se referencian a este valor, compensando así la tensión residual del circuito divisor de tensión.

5.5.3.3. Filtro de Madgwick

La fusión de las medidas de acelerómetro y giroscopio para producir una estimación de la orientación se realiza mediante el **filtro de Madgwick** [26], un algoritmo ampliamente utilizado en aplicaciones de seguimiento de movimiento por su bajo coste computacional y su buena precisión.

La orientación del dispositivo se representa internamente como un **cuaternión unitario** $\mathbf{q} = [q_0, q_1, q_2, q_3]$. Los cuaterniones son la representación estándar de rotaciones en gráficos por computador y sistemas de navegación inercial porque, a diferencia de los ángulos de Euler, no sufren el problema del *gimbal lock* (bloqueo de cardán), en el que ciertas orientaciones causan una pérdida de grado de libertad que impide representar rotaciones arbitrarias. Un cuaternión unitario codifica de forma compacta un eje de rotación y un ángulo en tan solo 4 números, y la composición de rotaciones se calcula mediante la multiplicación de cuaterniones, una operación con apenas 16 multiplicaciones y 12 sumas.

El funcionamiento del filtro puede resumirse en dos pasos que se ejecutan en cada iteración:

1. **Predicción con el giroscopio.** A partir de la velocidad angular medida $\boldsymbol{\omega}$ y el intervalo de tiempo Δt transcurrido desde la última muestra, se calcula cuánto ha rotado el dispositivo desde la estimación anterior y se actualiza el cuaternión:

$$\dot{\mathbf{q}}_{\omega} = \frac{1}{2} \mathbf{q}_{t-1} \otimes [0, \omega_x, \omega_y, \omega_z] \quad (5.2)$$

donde $\otimes$ denota la multiplicación de cuaterniones.

2. **Corrección con el acelerómetro.** Si el objeto no está en caída libre, el acelerómetro mide la dirección de la gravedad. El filtro calcula la diferencia entre la dirección de gravedad que predice la orientación actual y la que realmente mide el sensor, y aplica un paso de **descenso por gradiente** para reducir esa discrepancia. Este paso se pondera por el parámetro β :

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_{\omega} - \beta \nabla f(\mathbf{q}, \mathbf{a}) \quad (5.3)$$

donde f es la función de error entre la gravedad predicha y la medida, y ∇f su gradiente respecto al cuaternión.

El cuaternión resultante se normaliza para mantener la restricción de unitario y se almacena como la estimación actual de orientación. El parámetro β controla el compromiso entre los dos sensores: un valor alto da más peso al acelerómetro, lo que reduce la deriva pero suaviza los movimientos rápidos; un valor bajo da más peso al giroscopio, lo que mejora la respuesta dinámica pero permite la acumulación de deriva. El valor de $\beta = 0,1$ utilizado

en el firmware se determinó experimentalmente como adecuado para las velocidades de manipulación observadas durante las pruebas. Este valor resultó estable para los distintos perfiles de paciente evaluados, ya que las velocidades angulares durante las tareas de alcanzar-agarrar-transportar del BBT se encuentran dentro de un rango relativamente acotado. No obstante, en un escenario de despliegue clínico con población más heterogénea, podría ser deseable exponer β como parámetro configurable desde la interfaz del terapeuta, permitiendo ajustarlo en función de las características motoras del paciente (por ejemplo, un valor más alto para pacientes con temblor severo que requieran mayor filtrado del giroscopio).

Es importante señalar que, al utilizar solo 6 ejes (sin magnetómetro), el filtro de Madgwick puede corregir *pitch* y *roll* gracias a la referencia de la gravedad, pero **no tiene referencia alguna para el yaw**, que es la rotación horizontal del objeto (girar sobre sí mismo visto desde arriba). Esto significa que el *yaw* deriva inevitablemente con el tiempo. La corrección de esta deriva se delega al módulo de fusión, que la realiza periódicamente comparando la orientación del sensor con la referencia visual proporcionada por los *keypoints* del modelo YOLO (Sección 5.7.7).

5.5.3.4. Bucle principal del firmware

El bucle principal del firmware se ejecuta de forma continua mientras el microcontrolador está encendido. Cada iteración realiza las siguientes operaciones, condicionadas al estado actual del dispositivo:

En el estado **IDLE** (sin conexión BLE), el bucle simplemente duerme durante 1 segundo entre iteraciones. La IMU está desactivada y el consumo se mantiene en torno a 2 mA, dedicado casi exclusivamente a la emisión periódica de paquetes de publicidad BLE (*advertising*) para que el visor pueda descubrir el dispositivo.

En el estado **CONNECTED**, cada iteración dura aproximadamente 10 ms (`k_sleep(K_MSEC(10))`), lo que establece un bucle a ≈ 100 Hz. En cada iteración se realizan las siguientes tareas:

1. Se lee el valor del FSR mediante el ADC.
2. Se obtiene una muestra del acelerómetro y del giroscopio.
3. Se restan los *offsets* de calibración.
4. Se ejecuta una iteración del filtro de Madgwick con el Δt real transcurrido desde la iteración anterior, calculado mediante el reloj del kernel de Zephyr. Para evitar inestabilidades numéricas, el Δt se limita al rango [1ms, 100ms].
5. Se normaliza el cuaternión resultante y se empaqueta junto con la aceleración lineal cruda en la estructura de datos que se enviará por BLE.
6. Se envían las notificaciones BLE de forma **decimada**: el paquete IMU, formado por cuaternión y aceleración lineal cruda, se notifica cada 3 iteraciones (≈ 33 Hz), y el valor del FSR cada 10 iteraciones (≈ 10 Hz). Esta decimación reduce la carga sobre la pila BLE y el consumo de radio sin sacrificar información útil, ya que la tasa de 33 Hz es suficiente para el módulo de fusión y los cambios de presión del agarre son mucho más lentos que los cambios de orientación.
7. Cada 10 segundos se lee el voltaje de la batería (con ganancia ADC de 1/6, que cubre el rango completo de 3,3–4,2 V de la celda LiPo sin saturar) y se publica el nivel de carga como porcentaje.

El siguiente fragmento muestra la conversión del cuaternión recibido por BLE desde el sistema de coordenadas diestro del sensor al sistema zurdo de Unity, tal como se implementa en el componente BLEManager del visor:

Listado 5.1: Conversión de cuaternión sensor → Unity en el callback BLE.

```

1  // Cuaternion recibido del sensor (right-handed)
2  float w = BitConverter.ToSingle(data, 0);
3  float x = BitConverter.ToSingle(data, 4);
4  float y = BitConverter.ToSingle(data, 8);
5  float z = BitConverter.ToSingle(data, 12);
6
7  // Valor adicional cuando el firmware envia el paquete extendido de 20
   bytes
8  device.imuLinearMotion = data.Length >= 20
9      ? Mathf.Max(0f, BitConverter.ToSingle(data, 16))
10     : 0f;
11
12  // Conversion a sistema zurdo de Unity:
13  // negar Y y Z para cambiar de convención
14  device.orientation = new Quaternion(-y, -z, x, ←
   ↪ w).normalized;

```

5.5.4. Gestión energética

La gestión energética resulta de la combinación de la máquina de estados descrita que desactiva la IMU cuando no hay conexión, con la desactivación de los periféricos de depuración en la configuración de compilación. El resultado es un consumo medio en torno a 2–2,5 mA que proporciona una **autonomía operativa superior a 75 horas** con la batería de 200 mAh utilizada en los prototipos de prueba, y de más de 38 horas con la batería reducida de 100 mAh. En ambos casos, la duración es muy superior a la de cualquier sesión clínica (típicamente 30–60 minutos), lo que elimina una de las principales fricciones en la adopción de dispositivos embebidos en entornos hospitalarios: la necesidad de cargar los dispositivos entre usos.

Al producirse una desconexión BLE, el firmware ejecuta un **reinicio en frío** (*cold reboot*) del microcontrolador completo en lugar de volver al estado IDLE. Esta decisión de diseño garantiza que cada sesión arranca desde un estado limpio: el filtro de Madgwick se reinicializa, los *offsets* de calibración se recalculan, y cualquier acumulación de error residual se elimina. El coste de esta aproximación es un tiempo de arranque de ≈ 3 segundos (2 segundos de calibración más el reinicio del stack BLE), que resulta imperceptible en un contexto clínico.

5.5.5. Comunicación BLE

5.5.5.1. Conceptos fundamentales de BLE y GATT

Bluetooth Low Energy (BLE) es un estándar de comunicación inalámbrica diseñado para dispositivos de bajo consumo. A diferencia del Bluetooth clásico, BLE transmite pequeñas cantidades de datos en ráfagas cortas, lo que reduce drásticamente el consumo de radio. En el modelo BLE, un **periférico** expone sus datos mediante un perfil **GATT** (*Generic Attribute Profile*), que organiza la información en una jerarquía de **servicios** y **características**. Un servicio agrupa un conjunto de datos relacionados (por ejemplo, datos inerciales), y cada característica es un dato individual (por ejemplo, el cuaternión de orientación). El **central** (el visor Meta Quest 3) se conecta al periférico, descubre sus servicios y características, y puede

leer datos puntualmente o suscribirse a **notificaciones** para recibirlos de forma automática cada vez que cambian.

5.5.5.2. Perfil GATT del objeto sensorizado

El firmware define dos servicios GATT:

- Un **servicio personalizado** identificado por un UUID de 128 bits, con dos características:
 - **Paquete IMU extendido.** 20 bytes que codifican los cuatro componentes del cuaternión (q_0, q_1, q_2, q_3) como flotantes IEEE 754 de simple precisión en orden $w-x-y-z$, seguidos de un quinto flotante con la aceleración lineal cruda empleada como indicador de movimiento. Se notifica a ≈ 33 Hz. El receptor mantiene compatibilidad con paquetes antiguos de 16 bytes, en cuyo caso el valor de aceleración se considera nulo.
 - **Fuerza FSR.** 4 bytes que codifican el valor entero con signo de 32 bits del sensor de presión, ya corregido con el *offset* de calibración. Se notifica a ≈ 10 Hz.
- El **servicio estándar de batería** (UUID 0x180F del BT SIG), que publica el nivel de carga como un único byte de porcentaje (0–100 %). Al ser un servicio estandarizado, el visor puede leerlo de forma genérica sin conocer el perfil personalizado.

Cada cubo físico se anuncia por BLE con un nombre que contiene su color (CubeRed, CubeBlue, CubeGreen), lo que permite al visor filtrar los periféricos durante el escaneo y asociar cada dispositivo a un color de forma automática. Conviene señalar que esta estrategia de identidad basada en color presenta una **limitación de escalabilidad**: en el *Box and Block Test* estándar se utilizan hasta 150 bloques con colores repetidos, lo que imposibilita la discriminación individual por tonalidad. Para un despliegue clínico completo sería necesario explorar mecanismos de identificación complementarios, como marcadores visuales únicos por objeto o preidentificación por movimiento. Esta limitación se recoge como línea de trabajo futuro en las conclusiones del documento.

5.5.5.3. Recepción en el visor: componente BLEManager

En el lado del visor, el componente BLEManager implementado en Unity gestiona el ciclo completo de conexión BLE. Al iniciar, solicita los permisos de Android necesarios (BLUETOOTH_SCAN, BLUETOOTH_CONNECT), inicializa la radio BLE en modo central y comienza el escaneo buscando periféricos cuyo nombre coincida con una lista de nombres objetivo. Cuando encuentra un dispositivo, pausa el escaneo (algunas pilas BLE de Android no permiten escanear y conectar simultáneamente), establece la conexión GATT, descubre los servicios y las características disponibles, y se suscribe a las notificaciones de las tres características de interés. Si la conexión se pierde, intenta reconectarse automáticamente.

El flujo de comunicación BLE se resume en el siguiente esquema: (1) el cubo emite paquetes de publicidad (*advertising*) con su nombre de color; (2) el visor escanea, filtra por nombre y conecta; (3) se suscribe a las notificaciones del paquete IMU extendido (≈ 33 Hz), FSR (≈ 10 Hz) y batería; (4) cada notificación actualiza la estructura DeviceData del dispositivo correspondiente; (5) el FusionSystemManager escucha el evento OnConnected e instancia el *tracker*.

El siguiente fragmento ilustra la resolución de nombre por dirección MAC, un mecanismo necesario porque Android cachea el nombre BLE del chip y, tras reiniciar la aplicación en el

visor surgen confusiones entre los distintos dispositivos:

Listado 5.2: Resolución de nombre BLE por dirección MAC.

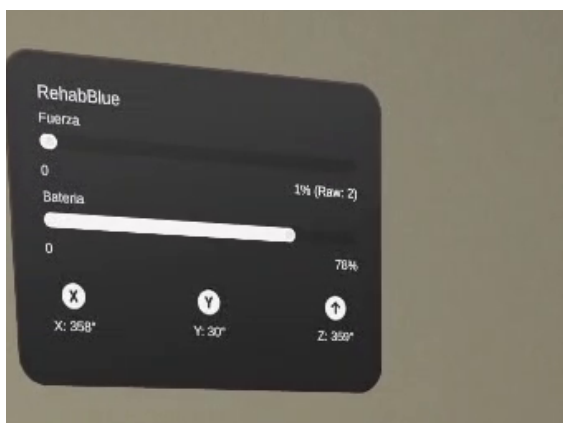
```

1  private static readonly Dictionary<string, string> ←
    ↪ macDeviceMap = new()
2  {
3      { "E4:32:5A:81:6E:64", "MugGreen" },
4      { "FE:13:2A:ED:BC:78", "CubeBlue" },
5      { "D4:91:B7:47:30:BD", "CubeRed" },
6  };
7
8  private string ResolveNameByMac(string address, string ←
    ↪ deviceName)
9  {
10     if (enableMacLookup &&
11         macDeviceMap.TryGetValue(address.ToUpper(), out ←
            ↪ string mapped))
12         return mapped;
13     return deviceName;
14 }

```

Al recibir una notificación de la característica del cuaternión, el BLEManager realiza una **conversión de sistema de coordenadas**. El sensor produce cuaterniones en un sistema de coordenadas diestro (*right-handed*), que es el estándar en navegación inercial, pero Unity utiliza un sistema zurdo (*left-handed*). La conversión consiste en negar dos de los cuatro componentes del cuaternión, y se ejecuta en el propio *callback* de notificación para que el resto del sistema trabaje directamente en el espacio de Unity.

Para cada dispositivo conectado, el BLEManager mantiene una estructura DeviceData que almacena la orientación más reciente, la aceleración lineal cruda, la longitud del paquete IMU recibido, la fuerza de agarre y el nivel de batería. El componente expone un evento estático OnConnected que se dispara cada vez que un periférico se conecta con éxito; el FusionSystemManager escucha este evento para instanciar el *tracker* correspondiente, como se describe en la Sección 5.7.1.



(a) Panel de telemetría BLE en tiempo real.

Figura 5.4: Interfaz de realidad mixta: retroalimentación en tiempo real y datos de los sensores recibidos por BLE, incluyendo valores de fuerza, orientación, movimiento inercial y nivel de batería.

5.6. MÓDULO DE VISIÓN

5.6.1. Modelo YOLO-Pose y clases de detección

El núcleo del sistema de visión es un modelo **YOLOv8-nano Pose** entrenado específicamente para los cubos del BBT. YOLO (*You Only Look Once*) es una familia de modelos de detección de objetos basados en redes neuronales convolucionales, produciendo simultáneamente las *bounding boxes* y las puntuaciones de confianza. La variante *Pose* añade la capacidad de detectar *keypoints* (puntos de interés predefinidos) dentro de cada objeto, que en nuestro caso corresponden a los cuatro vértices en la cara superior del cubo.

A diferencia de los marcadores fiduciarios tradicionales como ArUco o los códigos QR, que requieren que el patrón sea visible de forma íntegra y a una distancia específica para ser reconocidos, el modelo YOLO reconoce los objetos a partir de mapas de características aprendidos durante el entrenamiento. Esto permite una detección robusta incluso cuando la mano del usuario ocluye parcialmente el objeto, situación que se produce constantemente durante las tareas de alcanzar y agarrar del BBT.

El modelo distingue tres clases de detección:

- **CuboNormal** (clase 0): un cubo sin ninguna marca especial. El modelo devuelve únicamente la *bounding box* y la puntuación de confianza. Este tipo de detección proporciona información posicional pero no angular.
- **CuboSensor** (clase 1): un cubo que lleva la pegatina visible en una de sus caras. Además de la *bounding box*, el modelo devuelve 4 *keypoints* con sus coordenadas en píxeles y sus puntuaciones de visibilidad. Estos *keypoints* son fundamentales para la calibración de la orientación, como se detalla más adelante.
- **Mug** (clase 2): una taza a modo de demostración de la escalabilidad del sistema, el cual permite añadir nuevos objetos sin apenas cambios. Debido a su naturaleza de demostración no se ha implementado el sistema de calibración y por tanto una segunda clase.

El enfoque basado en aprendizaje profundo desacopla la lógica de reconocimiento de la arquitectura del sistema: la incorporación de nuevos objetos terapéuticos como vasos, cepillos o cubiertos requiere únicamente reentrenar el modelo con nuevas instancias anotadas, sin modificar el código de seguimiento. Para facilitar esta extensibilidad, la lógica geométrica dependiente de la forma del objeto se ha abstraído en una jerarquía de *ShapePolicy* (*ShapePolicy* → *CubeShapePolicy*), descrita en detalle en la Sección 5.7.2.

La dificultad adicional que introduce el modelo *pose* frente a un detector convencional es la necesidad de anotar los *keypoints* en el *dataset* de entrenamiento, un proceso más laborioso que anotar solo *bounding boxes*.

5.6.1.1. Justificación del uso de clases múltiples

Para poder obtener la orientación del cubo en el espacio (6-dof) y permitir la calibración del yaw, es necesario que el objeto no sea totalmente simétrico para el modelo. Por ello, se pega una pegatina rectangular blanca (alto contraste, fácilmente reconocible) de tal forma que podemos obtener también el norte. En algunos casos, esta pegatina no será visible por oclusión parcial o total, pero el sistema debe seguir reconociendo el objeto.

5.6.1.2. Ventajas de la detección de puntos de interés frente a marcadores QR

La adopción de los *keypoints* fue el resultado de un proceso iterativo. Inicialmente se utilizaron marcadores QR pegados a la cara superior del cubo para obtener la referencia angular de *yaw*. Sin embargo, la detección del QR exigía que el usuario mirase el cubo desde arriba, es decir, que la cámara del visor estuviese prácticamente perpendicular a la cara del marcador. En la práctica esto fallaba según la altura del paciente o la posición del cubo en la mesa: si el ángulo de visión era demasiado oblicuo, el QR no se detectaba y la calibración no se ejecutaba. El paciente debía inclinarse a propósito para que el visor viese el marcador, rompiendo la naturalidad de la interacción. Los modelos *pose* de YOLO resuelven este problema porque los *keypoints*, al ser puntos individuales, se detectan desde ángulos de observación mucho más amplios que un patrón bidimensional completo.

5.6.2. Pipeline de inferencia: YoloPoseAgent

El componente *YoloPoseAgent* gestiona el ciclo completo de inferencia sobre los fotogramas de la cámara *passthrough*. Ha sido desarrollado a partir del script de procesamiento del SDK de Meta, que no resultaba válido por no incluir soporte para *keypoints*. El proceso consta de tres fases:

Preprocesado (*letterboxing*). El modelo YOLO espera como entrada una imagen cuadrada de 640×640 píxeles, pero la cámara del Quest 3 produce fotogramas con una relación de aspecto diferente. Para adaptar la imagen sin distorsionarla, se aplica la técnica de *letterboxing*: la imagen se redimensiona manteniendo su relación de aspecto original hasta que cabe dentro del cuadrado de 640×640 , y los márgenes sobrantes se rellenan con negro. Los parámetros de esta transformación como factor de escala y desplazamientos horizontal/vertical, se almacenan porque serán necesarios después para convertir las coordenadas de salida del modelo de vuelta a las coordenadas reales de la imagen original.

El siguiente fragmento muestra la implementación del *letterboxing*:

Listado 5.3: Letterboxing de la imagen de cámara a 640×640 preservando aspect ratio.

```

1  private void PrepareInputRT(Texture src)
2  {
3      if (_captureRT == null || _captureRT.width != ImageSize)
4      {
5          if (_captureRT != null) _captureRT.Release();
6          _captureRT = new RenderTexture(ImageSize, ←
              ← ImageSize, 0,
7                                          RenderTextureFormat.ARGB32);
8      }
9
10     float srcW = src.width, srcH = src.height;
11     _lbScale = Mathf.Min(ImageSize / srcW, ImageSize / ←
        ← srcH);
12     float newW = srcW * _lbScale, newH = srcH * _lbScale;
13     _lbPadX = (ImageSize - newW) * 0.5f;
14     _lbPadY = (ImageSize - newH) * 0.5f;
15
16     if (Mathf.Abs(srcW - srcH) < 2f)
17     {
18         Graphics.Blit(src, _captureRT);
19         _lbScale = ImageSize / srcW;

```

```

20     _lbPadX = 0f; _lbPadY = 0f;
21 }
22 else
23 {
24     RenderTexture prev = RenderTexture.active;
25     RenderTexture.active = _captureRT;
26     GL.Clear(true, true, Color.black);
27     GL.PushMatrix();
28     GL.LoadPixelMatrix(0, ImageSize, ImageSize, 0);
29     Graphics.DrawTexture(new Rect(_lbPadX, _lbPadY, ↵
        ↵ newW, newH), src);
30     GL.PopMatrix();
31     RenderTexture.active = prev;
32 }
33 }

```

La imagen preparada se convierte a un tensor de forma $[1, 3, 640, 640]$ (un *batch* de 1 imagen, 3 canales de color, 640×640 píxeles) que es el formato de entrada esperado por la red neuronal.

Ejecución del modelo. El modelo YOLO está compuesto internamente por decenas de capas convolucionales que se ejecutan secuencialmente. En un dispositivo móvil como el Quest 3, ejecutar todas las capas en un solo fotograma provoca un pico de tiempo de procesamiento (*frame-time spike*) que puede hacer caer la tasa de refresco por debajo de los 72 Hz necesarios para una experiencia de realidad mixta cómoda. Para evitar este problema, el sistema ofrece dos modos de ejecución:

- **Modo inmediato:** todas las capas del modelo se ejecutan de golpe en un solo fotograma. Produce la menor latencia de detección pero los picos de tiempo de *frame* son perceptibles.
- **Modo repartido** (*split-over-frames*, por defecto): la función `ScheduleIterable()` del motor de inferencia de Unity distribuye la ejecución a lo largo de varios fotogramas, avanzando un número configurable de capas por *frame* (50 por defecto). La tasa de refresco se mantiene estable a costa de que cada detección tarda varios fotogramas en completarse, lo que aumenta la latencia. Este compromiso es aceptable porque la posición del objeto cambia relativamente poco entre fotogramas consecutivos a la frecuencia de refresco del visor.

El siguiente fragmento muestra la coroutine del modo repartido:

Listado 5.4: Modo split-over-frames: ejecución distribuida del modelo YOLO.

```

1 private IEnumerator RunInferenceSplit()
2 {
3     _busy = true;
4     float t0 = Time.realtimeSinceStartup;
5
6     var src = _cam.GetTexture();
7     if (src == null) { _busy = false; yield break; }
8     PrepareInputRT(src);
9
10    var inputTensor = new Tensor<float>(
11        new TensorShape(1, 3, ImageSize, ImageSize));
12    TextureConverter.ToTensor(_captureRT, inputTensor,
13        new TextureTransform());

```

```

14
15     var schedule = _worker.ScheduleIterable(inputTensor);
16     int layerCount = 0;
17     while (schedule.MoveNext())
18     {
19         layerCount++;
20         if (layerCount % layersPerFrame == 0)
21             yield return null; // Esperar al siguiente frame
22     }
23
24     var outputTensor = _worker.PeekOutput() as ↩
↪ Tensor<float>;
25     float[] data = outputTensor.DownloadToArray();
26     ProcessOutput(data, outputTensor.shape[2], src);
27
28     inputTensor.Dispose();
29     _lastInferenceMs = (Time.realtimeSinceStartup - t0) * ↩
↪ 1000f;
30     _busy = false;
31 }

```

Postprocesado y NMS. El modelo produce como salida un tensor de forma $[1, 6 + 4 \times 3, N_{anchors}]$, donde $N_{anchors}$ es el número de posibles detecciones que el modelo evalúa. Cada columna de ese tensor codifica una detección candidata:

- Filas 0–3: coordenadas del centro y dimensiones de la *bounding box* (c_x, c_y, w, h), en el espacio *letterboxed* de 640×640 .
- Filas 4–5: *logits* de las dos clases (CuboNormal y CuboSensor), que se convierten a probabilidades aplicando la función sigmoide.
- Filas 6–17: coordenadas (x, y) y visibilidad de cada uno de los 4 *keypoints*, 3 valores por *keypoint*.

El agente filtra las candidatas cuya confianza máxima entre ambas clases es inferior a un umbral configurable, revierte la transformación de *letterbox* sobre todas las coordenadas para recuperar las posiciones en píxeles reales, y ejecuta una **supresión de no-máximos** (NMS, *Non-Maximum Suppression*) greedy. La NMS funciona de forma sencilla: se ordenan todas las detecciones por puntuación de mayor a menor, se selecciona la mejor, se eliminan todas las que se solapan excesivamente con ella (medido mediante la *Intersección sobre Unión*, IoU), y se repite el proceso con las restantes. El resultado es una lista limpia de detecciones sin duplicados. Se implementó una NMS propia en lugar del NMSCompute estándar del SDK de Meta porque este último no soporta el postprocesado de *keypoints*.

5.6.3. Proyección 2D a 3D mediante mapa de profundidad

El componente ObjectDetectionVisualizerV2 recibe las detecciones 2D del agente YOLO y las proyecta a coordenadas 3D del mundo. El visor Meta Quest 3 proporciona un **mapa de profundidad**: una textura en la que cada píxel almacena la distancia (en metros) desde la cámara hasta el primer objeto que encuentra en esa dirección. Combinando la posición en píxeles de una detección con la profundidad correspondiente del mapa, es posible calcular la posición tridimensional del objeto en el espacio del mundo.

El proceso de proyección para el centro de la *bounding box* funciona de la siguiente manera: se convierte la posición en píxeles a coordenadas normalizadas del *viewport* de la

cámara, se genera un rayo desde la posición de la cámara *passthrough* en la dirección de ese píxel, se transforma la dirección del rayo al espacio de coordenadas del mapa de profundidad mediante la matriz de vista-proyección, se muestrea la profundidad en la posición resultante, y finalmente se desplaza el origen del rayo por esa distancia para obtener el punto 3D del mundo. Adicionalmente, se proyectan los bordes de la *bounding box* a la misma profundidad para estimar la escala 3D del objeto detectado.

Estrategia de profundidad híbrida para keypoints. La proyección de los *keypoints* requiere un tratamiento especial que merece explicación detallada. Los vértices de la pegatina se sitúan cerca de los bordes del cubo, y en los bordes de un objeto el mapa de profundidad presenta un artefacto conocido como *edge bleeding*: al muestrear la profundidad en el borde de un objeto, el píxel puede captar parcialmente la profundidad de la superficie que hay detrás (en nuestro caso, la mesa), devolviendo un valor intermedio o directamente la profundidad de la mesa. Esto haría que el *keypoint* se proyecte a una posición incorrecta, típicamente por detrás y por debajo de donde debería estar.

Para resolver este problema se implementó una **estrategia de profundidad híbrida** en dos pasos:

1. Se muestrea la profundidad individual de cada *keypoint* en su posición en píxeles.
2. Se compara esa profundidad con la del centro de la *bounding box* (que, al estar en el centro del objeto, raramente sufre *edge bleeding*). Si la diferencia es menor que un umbral de tolerancia (8 cm, que cubre la diagonal del cubo de 2,5 cm más un margen de ruido), se acepta la profundidad individual: el *keypoint* conserva su propia profundidad, lo que permite capturar la variación real cuando el cubo está inclinado. Si la diferencia supera el umbral, se descarta y se asigna al *keypoint* la misma profundidad que el centro (*shared depth*), asumiendo que el valor individual estaba contaminado por *edge bleeding*.

5.6.4. Clasificación de color

La identidad de cada cubo se determina mediante un muestreo de color en la textura de la cámara *passthrough*. El procedimiento es el siguiente: se toma el píxel central de la *bounding box* mediante interpolación bilineal (para suavizar el ruido del sensor de la cámara), se convierte del espacio RGB al espacio HSV (*Hue-Saturation-Value*), y se clasifica por su componente de *hue* (tono) en una de cuatro identidades discretas:

- *Rojo*: $hue \geq 330$ o $hue \leq 30$ (el rojo envuelve el inicio del círculo cromático).
- *Verde*: $80 \leq hue \leq 160$.
- *Azul*: $170 \leq hue \leq 260$.
- *Amarillo*: $40 \leq hue \leq 70$.

Los píxeles acromáticos (saturación $< 0,25$) o muy oscuros (valor $< 0,2$) se descartan como *Unknown* para evitar falsas clasificaciones en zonas de sombra. El espacio HSV se eligió porque separa la información de color (*hue*) de la luminosidad (*value*), lo que hace la clasificación más robusta ante cambios de iluminación. Los rangos de tono se definieron empíricamente bajo las condiciones de iluminación del *passthrough* del Quest 3 y resultaron estables en los entornos de prueba utilizados.

5.6.5. Validación geométrica

Cuando los 4 *keypoints* son válidos y están proyectados en 3D, se comprueba si las distancias entre los pares de vértices superior e inferior son coherentes con el tamaño real conocido del cubo. Si la pegatina tiene una longitud de arista de, por ejemplo, 1,5 cm, las distancias entre los *keypoints* del borde superior y del borde inferior deberían estar próximas a ese valor. Si difieren más de una tolerancia configurable, la detección se marca como geoméricamente inválida.

Este *flag* de validación (*IsGeometricValid*) no bloquea la emisión de detecciones, las cuales se emiten siempre que haya 3 o más *keypoints* válidos, sino que se propaga al módulo de fusión como una **señal de calidad**. El módulo de fusión utiliza esta señal para decidir qué camino de calibración aplicar: la calibración post-suelta, que es la más exigente, requiere 4 *keypoints* válidos y validación geométrica positiva, mientras que la calibración continua puede funcionar con estándares más relajados.

5.6.6. Integración con Meta AI Building Blocks

La integración del modelo YOLO con el entorno de desarrollo del Meta Quest 3 evolucionó a lo largo del proyecto. En la primera fase, los modelos se integraban mediante **Unity Sentis**, partiendo del ejemplo *MultiObjectDetection* del repositorio de muestras de Oculus, que era la única opción disponible antes de que Meta publicara sus herramientas nativas de IA. Cuando apareció el *framework* **AI Building Blocks**, la arquitectura se migró para aprovechar su mayor optimización y modularidad.

Durante esa migración se detectó un **error de precisión** en el componente estándar *ObjectDetectionVisualizer* del SDK: los cálculos de la matriz de proyección de profundidad introducían una deriva espacial que desalineaba las cajas de detección 3D respecto al centro físico del objeto. Para el sistema de fusión esa desviación era inaceptable, de modo que se integró una implementación corregida adaptada del repositorio de demos de AI Building Blocks [36], que resuelve el *raycasting* viewport-to-world con mayor exactitud.

Finalmente, con la evolución a un modelo YOLOv8-Pose se volvió a usar Unity Sentis, esta vez de forma pura y personalizada.

5.6.7. Problema de asociación de identidad

Uno de los retos centrales en sistemas visuo-inerciales con múltiples objetos es el **problema de asociación de datos**: cuando la cámara detecta varios cubos y el sistema recibe datos BLE de varios sensores, saber qué sensor corresponde a qué detección visual es un reto. Fallar la asociación haría que el gemelo digital de un cubo siguiese los movimientos del sensor de otro, invalidando completamente el seguimiento.

Trabajos previos han abordado este problema mediante marcadores LED activos que emiten identificadores ópticos [1] o mediante la correlación entre el movimiento visual y los datos inerciales para la asignación inicial de identidad [11]. Aunque eficaces en sus respectivos contextos, estos enfoques pueden resultar invasivos o romper el flujo natural de la tarea terapéutica.

Durante el desarrollo se exploró la posibilidad de utilizar la **intensidad de señal recibida** (RSSI) de las conexiones BLE como criterio de proximidad: la idea era asignar cada detección visual al dispositivo cuya señal fuera más intensa, bajo el supuesto de que el objeto más cercano al visor sería el detectado. Sin embargo, la validación experimental descartó esta alternativa. En el escenario del BBT, los cubos se encuentran a distancias similares entre sí

y respecto al usuario (20–60 cm), y el RSSI de BLE presenta una varianza suficientemente alta a esas distancias cortas como para que las lecturas de distintos cubos se solapen y la discriminación sea inviable.

La solución adoptada fue un **análisis pasivo de identidad de color**. Cada cubo lleva un color diferenciado y ese color queda mapeado de forma fija a un dispositivo BLE concreto durante la configuración de la sesión, mediante la tabla de *bindings* del FusionSystemManager. Por ejemplo: el cubo rojo se anuncia por BLE como CubeRed, y en la tabla de mapeos el color *Rojo* está vinculado al filtro de nombre Red". Cuando la cámara detecta un cubo y la clasificación de color devuelve *Rojo*, el sistema sabe sin ambigüedad a qué sensor asociarlo.

El enrutamiento de detecciones se ha generalizado para soportar tanto objetos diferenciados por color (cubos) como objetos de clase única (taza). Cuando la ShapePolicy del *tracker* define la misma clase para estado visible y ocluido (VisibleClass == OccludedClass), el color se ignora en el enrutamiento, ya que solo puede existir un objeto de esa clase:

Listado 5.5: Enrutamiento de detecciones por clase y color en el FusionSystemManager.

```

1  foreach (FusionTracker tracker in _activeTrackers)
2  {
3      if (tracker == null || tracker.shape == null) continue;
4      bool classMatch = (det.Class == ↵
5          ↵ tracker.shape.VisibleClass ||
6              det.Class == ↵
7              ↵ tracker.shape.OccludedClass);
8      if (!classMatch) continue;
9
10     // Clase unica (taza): ignorar color; solo hay uno de este tipo.
11     // Clases duales (cubo): exigir coincidencia de color.
12     bool singleClass =
13         (tracker.shape.VisibleClass == ↵
14             ↵ tracker.shape.OccludedClass);
15     bool colorMatch = singleClass ||
16         (tracker.myColorIdentity == det.ColorCategory);
17     if (!colorMatch) continue;
18
19     targetTracker = tracker;
20     break;
21 }

```

5.6.8. Seguimiento de manos

El módulo de visión integra también el **seguimiento de manos nativo de Meta SDK**, que proporciona la pose tridimensional de las manos del paciente (posición y orientación de la muñeca, articulaciones y puntas de los dedos) mediante visión por computador con alta precisión. El sistema permite seleccionar la mano dominante del paciente (derecha o izquierda) desde la interfaz del juego serio, y esa elección se propaga a todos los *trackers* activos a través del FusionSystemManager. Durante el estado de agarre, la posición del gemelo digital queda anclada a la pose de la mano seleccionada, como se detalla en la Sección 5.7.9.

5.7. MÓDULO DE FUSIÓN

5.7.1. Arquitectura del módulo

El módulo de fusión constituye la **principal aportación técnica** de este trabajo. Su función es integrar las señales visuales e inerciales para generar una estimación continua y robusta de la pose 6-DoF (3 grados de posición + 3 grados de orientación) del objeto terapéutico, incluso durante los periodos de oclusión visual completa. El módulo está compuesto por cuatro subcomponentes que operan de forma coordinada:

- El **FusionSystemManager**, orquestador del estado global.
- El **FusionTracker**, estimador de pose individual por objeto.
- El **CubeContactAssessor**, evaluador multimodal del estado de contacto mano-cubo.
- La jerarquía **ShapePolicy**, abstracción de la geometría dependiente de la forma del objeto.

El **FusionSystemManager** actúa como orquestador del estado global del sistema. Mantiene el conjunto de *trackers* activos, uno por cada objeto instrumentado, y gestiona la asociación determinista entre detecciones visuales y dispositivos BLE. Cuando un periférico BLE se conecta, el *manager* busca en su tabla de mapeos qué color le corresponde e instancia o reutiliza el **FusionTracker** con el prefab del gemelo digital asociado. A cada *tracker* se le configura la mano activa del paciente, la identidad de color y el nombre del dispositivo BLE para que pueda leer sus datos sensoriales. Cada vez que la cámara emite un lote de detecciones visuales, el *manager* las enruta al *tracker* correspondiente mediante una asociación jerárquica. En primer lugar, intenta una correspondencia directa por clase visual y color estimado. Si la mano ocluye el color del cubo y no existe una correspondencia inequívoca, el sistema aplica un mecanismo de respaldo cinemático: si la detección está próxima a la mano activa y un único *tracker* compatible informa de movimiento inercial, la detección se asocia a dicho objeto. Como último recurso, si solo queda un candidato visual razonable no reclamado, se utiliza una asociación por eliminación. Una vez asociada la detección a un *tracker* concreto, se notifica su posición al **CubeContactAssessor** y se evalúa el filtro de falsos positivos de mano descrito en la Sección 5.7.11.

El **FusionTracker** encapsula toda la lógica de estimación de pose para un objeto individual. En cada fotograma, su método `LateUpdate` ejecuta un bucle que:

1. Lee la orientación, la aceleración lineal cruda y la presión más recientes del dispositivo BLE asignado, aplicando la rotación de montaje configurable (`mountingRotation`) para compensar diferencias en la orientación física del sensor dentro del cubo.
2. Aplica la compensación de *bias* del giroscopio si está habilitada.
3. Interpola el *offset* de calibración hacia su valor objetivo (para transiciones suaves).
4. Actualiza los contadores de estabilidad de la IMU y de la cabeza del paciente (cuántos fotogramas consecutivos la orientación no ha cambiado significativamente).
5. Alimenta al **CubeContactAssessor** con todas las señales disponibles (FSR, movimiento inercial enriquecido con aceleración lineal, posición de mano, posición YOLO, posición del tracker).
6. Gestiona el agarre confirmado por FSR y, cuando no existe presión suficiente, el puente inercial de oclusión basado en aceleración y desplazamiento de la mano (Sección 5.7.5).

7. Aplica la transformación final al gemelo digital según el estado actual: anclaje local a la mano si existe agarre FSR, propagación por delta de mano si está activo el puente inercial, o posición visual con bloqueo de cara si está en la mesa.
8. Registra el estado completo del cubo (posición, rotación, velocidad, nivel de contacto, fuente de seguimiento) en el módulo de *logging*.

Las detecciones visuales y las calibraciones de *keypoints* llegan de forma asíncrona desde el *manager* mediante llamadas directas a los métodos públicos del *tracker*, y actualizan su estado interno para que la siguiente iteración del bucle las tenga en cuenta.

5.7.2. Abstracción de forma: ShapePolicy

Para que el sistema de fusión sea extensible a objetos con geometrías diferentes al cubo, la lógica dependiente de la forma del objeto se ha abstraído en una jerarquía de *ScriptableObject*. La clase base *ShapePolicy* define la interfaz que cada forma debe implementar:

Listado 5.6: Interfaz de ShapePolicy: contrato para cada geometría.

```

1 public abstract class ShapePolicy : ScriptableObject
2 {
3     public abstract int KeypointCount { get; }
4     public abstract int MinKeypointsForCalib { get; }
5     public abstract int HighConfidenceKeypointCount {
6         ↳ get; }
7
8     public abstract CubeClass VisibleClass { get; }
9     public abstract CubeClass OccludedClass { get; }
10
11     public abstract Vector3 ComputeForwardXZ(Vector3[] kpts,
12         ↳ bool[] valid, Vector3 bodyCenter);
13     public abstract Quaternion SnapRestingPose(Quaternion
14         ↳ ↳ idealRot,
15         ↳ ref Vector3 cachedDownAxis);
16     public virtual void
17         ↳ ApplyRealDimensionScaling(Transform obj) { }
18 }

```

La implementación concreta *CubeShapePolicy* define: 4 *keypoints* con un mínimo de 3 para calibrar, clases *CuboSensor* (visible) y *CuboNormal* (ocluída), cálculo de dirección frontal a partir de los vectores laterales de los *keypoints* proyectados al plano XZ, bloqueo de cara inferior con histéresis de 0,15 en el producto escalar, y auto-escalado a las dimensiones reales medidas del cubo impreso en 3D.

Durante el desarrollo se realizaron pruebas preliminares con una **taza sensorizada** (*MugGreen*), verificando que la arquitectura de fusión es aplicable a objetos con geometría no cúbica. Si bien estas pruebas fueron limitadas en alcance, demuestran la viabilidad de extender el sistema a objetos terapéuticos de la vida diaria, una línea que se plantea como trabajo futuro.

5.7.3. Evaluador de contacto multimodal: CubeContactAssessor

La detección fiable del estado de interacción mano-cubo es un problema que ninguna señal individual resuelve por completo: el FSR solo detecta presión directa, el seguimiento de manos solo proporciona posición, la IMU solo indica movimiento, y YOLO solo posición visual. Para superar estas limitaciones, se desarrolló el *CubeContactAssessor*, un evaluador que

fusiona **todas** las señales disponibles en cada fotograma para producir un nivel de contacto discreto:

Listado 5.7: Niveles de contacto del CubeContactAssessor.

```
1 public enum ContactLevel
2 {
3     Resting    = 0,    // Cubo en mesa, mano lejos
4     Nearby     = 1,    // Mano cerca sin evidencia de contacto
5     Touching   = 2,    // Mano sobre el cubo (coincidencia YOLO)
6     HeldSoft   = 3,    // Contacto probable sin FSR (IMU + proximidad)
7     HeldHard   = 4,    // Cubo firmemente agarrado (FSR confirmado)
8 }
```

Las señales de entrada al evaluador son: (1) el valor crudo del FSR (0–1023), (2) el movimiento inercial entre fotogramas, calculado a partir del delta angular de la IMU y reforzado con la aceleración lineal cruda cuando la orientación apenas cambia, (3) la posición de la mano del paciente, (4) la confianza del seguimiento de mano, (5) la posición de la detección YOLO para **este cubo concreto**, (6) la frescura temporal de esa detección, y (7) la posición actual del gemelo digital.

De estas señales se derivan dos distancias críticas: la distancia mano ↔ detección YOLO de este cubo (que es siempre sincronizada porque ambas provienen del mismo fotograma visual), y la distancia mano ↔ posición del *tracker* (que puede tener *lag* respecto al movimiento real). Un acumulador con histéresis determina si la IMU está reportando movimiento sostenido. Esta señal de movimiento ya incorpora la aceleración lineal cruda recibida por BLE, por lo que también detecta traslaciones del cubo con baja variación angular.

La máquina de decisión aplica prioridad descendente: HeldHard si el FSR supera el umbral con mano trackeada; HeldSoft si convergen al menos dos señales débiles (por ejemplo, mano justo en la posición YOLO + movimiento inercial, o mano cerca + FSR débil + movimiento inercial); Touching si la mano coincide con la posición YOLO sin más evidencia; Nearby si la mano está en la vecindad; Resting en caso contrario. Esta gradación permite al resto del sistema tomar decisiones diferenciadas: el filtro de falsos positivos solo filtra en nivel Nearby; los niveles Touching y HeldSoft permiten interpretar que existe evidencia convergente de contacto o manipulación probable; y el agarre rígido a la mano se reserva para el camino confirmado por FSR, gestionado por el FusionTracker mediante los umbrales de entrada y salida del sensor.

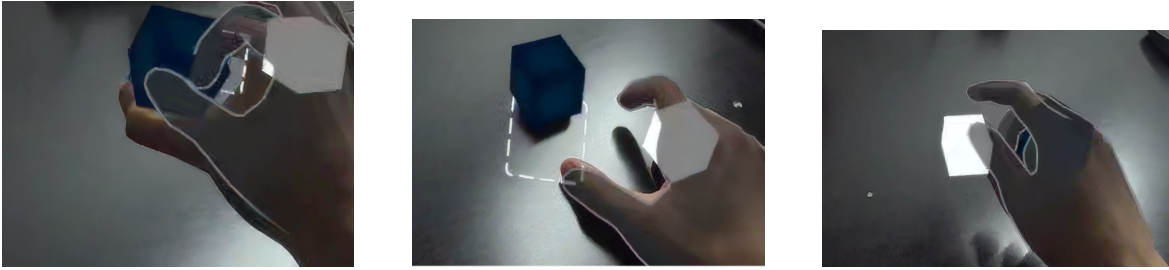
5.7.4. Validación táctil del agarre mediante FSR

Los sistemas de seguimiento exclusivamente visuales presentan dos artefactos conocidos cuando el usuario interactúa con un objeto:

- **Desplazamiento por oclusión** (*occlusion misplacement*): en el momento en que la mano tapa el objeto, el sistema pierde la referencia visual y tiende a colocar el objeto en el centro de la palma en lugar de mantenerlo en su posición real.
- **Adhesión prematura** (*false adhesion*): el objeto virtual se adhiere a la mano antes de que exista contacto físico real, o bien permanece pegado tras la suelta. Este segundo artefacto es particularmente problemático en rehabilitación neurológica, donde pacientes con lesión medular o ictus mantienen con frecuencia una postura de reposo en semiflexión que los sistemas visuales interpretan como un agarre activo, generando falsas repeticiones y distorsionando las métricas terapéuticas.

El sensor FSR elimina ambos problemas de raíz: el anclaje del objeto a la mano y la suspensión de las actualizaciones visuales solo ocurren cuando la señal FSR supera el umbral de entrada, con independencia de lo que la cámara observe en la configuración de los dedos.

La detección del agarre utiliza **histéresis** con dos umbrales separados: un umbral de entrada T_{enter} y un umbral de salida T_{exit} , con $T_{enter} > T_{exit}$. Mientras el sistema está en el estado de no-agarre, el agarre solo se activa cuando la presión supera T_{enter} . Una vez activo, el agarre solo se desactiva cuando la presión cae por debajo de T_{exit} . La razón de esta separación es que, sin ella, cuando la presión fluctúa alrededor de un umbral único (algo habitual en pacientes con temblor o control motor reducido), el estado oscilaría rápidamente entre agarre y no-agarre produciendo un parpadeo del gemelo digital. Además, se aplica un *cooldown* temporal entre la suelta y el siguiente agarre posible para absorber los rebotes mecánicos del sensor, evitando que un mismo gesto de soltar se interprete como una secuencia de suelta-agarre-suelta.



(a) Desplazamiento por oclusión.

(b) Anclaje prematuro.

(c) Dedo pegajoso (*sticky finger*).

Figura 5.5: Los tres artefactos principales de los sistemas de seguimiento exclusivamente visuales, resueltos mediante la validación táctil con FSR.

5.7.5. Puente inercial de oclusión basado en la mano

Una limitación del agarre exclusivamente basado en FSR es que requiere que el paciente aplique presión directamente sobre el sensor. Si el paciente agarra el cubo por una cara sin sensor, o si la colocación del sensor no coincide con su patrón natural de prensión, el FSR puede no registrar presión aunque el cubo esté siendo desplazado físicamente. En versiones anteriores, esta situación se resolvía mediante un mecanismo de *soft grip* basado en la convergencia de señales débiles. La versión actual elimina ese estado como fuente de agarre y lo sustituye por un **puente inercial de oclusión** (*IMU Occlusion Hand Bridge*).

El objetivo del puente no es afirmar que existe un agarre clínicamente confirmado, sino mantener la continuidad espacial del gemelo digital cuando el cubo se mueve bajo oclusión y no hay presión FSR suficiente. Para ello, el firmware envía junto al cuaternión una medida de aceleración lineal cruda. El FusionTracker utiliza esta señal para detectar movimiento traslacional intencional incluso cuando la orientación apenas cambia. Si la mano está suficientemente próxima al cubo o cubre su zona en la mesa, y la aceleración supera el umbral de entrada (T_{bridge}) durante varios fotogramas, el puente toma temporalmente el control de la posición.

A diferencia del agarre FSR, que ancla el objeto en el espacio local de la mano, el puente propaga la posición mediante el **delta de traslación de la mano**. En el instante de entrada se almacenan la posición inicial de la mano y la posición visual vigente del cubo:

$$\mathbf{x}_t = \mathbf{x}_0^{cube} + (\mathbf{h}_t - \mathbf{h}_0) \quad (5.4)$$

donde $\mathbf{x}_0^{cube}$ es la última posición fiable del cubo, $\mathbf{h}_0$ es la posición de la mano al iniciar el puente y $\mathbf{h}_t$ es la posición actual de la mano. De este modo, el objeto no salta al centro de la palma ni queda rígidamente pegado a la mano, sino que sigue el desplazamiento relativo de la mano mientras la IMU confirma que el cubo está en movimiento.

La entrada al puente combina cuatro condiciones principales: (1) mano trackeada y próxima al cubo o a su posición visual, (2) cobertura reciente de la zona de mesa donde estaba el objeto, para cubrir el intervalo en el que la visión empieza a degradarse, (3) aceleración lineal por encima del umbral de entrada (T_{bridge}), y (4) ausencia de otro cubo candidato más cercano o con movimiento lineal claramente superior. Esta última condición es importante cuando hay varios objetos instrumentados en escena, ya que evita que dos cubos reclamen simultáneamente el mismo movimiento de la mano.

Durante el puente, si vuelve a recibirse una detección YOLO intermedia, el sistema no rompe inmediatamente el seguimiento. En su lugar, reajusta de forma suave el origen del anclaje para reducir la deriva acumulada sin generar saltos perceptibles. La salida se produce cuando la aceleración lineal cae por debajo del umbral de reposo (T_{rest}) durante varios fotogramas, cuando la mano se aleja del objeto, cuando expira el tiempo máximo de activación o cuando aparece un agarre FSR confirmado. En este último caso, el puente se detiene y el sistema pasa al anclaje local clásico basado en FSR.

La fuente de estado registrada en los CSV refleja esta distinción: el agarre confirmado se etiqueta como `fsr`, el seguimiento temporal durante oclusión como `imu_bridge`, y la ausencia de agarre o puente como `none`. Esto evita mezclar en el análisis clínico un agarre validado por presión con una propagación cinemática diseñada únicamente para preservar la continuidad visual y espacial durante la oclusión.

5.7.6. Filtrado espaciotemporal

Una vez que un *tracker* recibe una posición candidata $\mathbf{p}_{yolo}$ proveniente de la detección visual, no la aplica directamente al gemelo digital. El mapa de profundidad del visor introduce un ruido de medida (*jitter*) que, sin filtrar, haría que el cubo virtual temblase constantemente, rompiendo la ilusión de solidez. Para mitigar este efecto, el sistema aplica dos filtros complementarios que determinan si la nueva posición representa un movimiento real o es ruido:

- **Deadzone de distancia** (λ_{move}): la posición solo se actualiza si el desplazamiento respecto a la posición actual del gemelo digital supera un umbral mínimo de 5 mm. Los desplazamientos menores se consideran ruido del mapa de profundidad y se descartan.
- **Deadzone angular**: desde la perspectiva del jugador (usando la posición de la cabeza como referencia), si la nueva posición apunta en prácticamente la misma dirección que la actual (diferencia angular inferior a un umbral configurable), se descarta como *jitter* lateral del píxel. Este filtro es complementario al anterior porque captura movimientos aparentes que son pequeños en ángulo pero pueden ser grandes en distancia si el objeto está lejos.

Ambas *deadzones* se desactivan automáticamente en dos situaciones específicas: cuando la mano del paciente se acerca al cubo (distancia inferior a 12 cm), para que la posición se actualice rápidamente y el agarre sea fluido; y justo después de soltar el cubo, para que se repositone sin resistencia en la posición donde la mano lo dejó.

Gate de estabilidad por movimiento de cabeza. Al mover la cabeza, el fotograma de cámara se captura con una pose del visor ligeramente anterior a la actual, lo que introduce latencia en la proyección píxel→mundo. Si el cubo está quieto (IMU confirma estabilidad) pero la cabeza se mueve, la posición 3D de YOLO “salta” aunque el cubo real no se haya movido. Para evitar este artefacto, las actualizaciones de posición se bloquean cuando la IMU indica reposo y la cabeza se está moviendo. Este bloqueo se desactiva cuando hay indicación de fuerza (`_forceNextVisualUpdate`), cuando el cubo se mueve, o cuando la cabeza está quieta.

Además, cuando el objeto está bloqueado sobre la mesa, la posición visual no se usa directamente como centro del cubo. El sistema proyecta la detección visual sobre el plano de mesa calibrado mediante MRUK, incorporando tanto una corrección manual como una corrección automática estimada a partir de muestras visuales estables al inicio de la sesión. Esta proyección reduce los errores derivados de que la profundidad visual pueda corresponder a la superficie visible del cubo y no necesariamente al punto de referencia utilizado por el gemelo digital.

Cuando la posición pasa los filtros, el gemelo digital no salta instantáneamente a ella, sino que interpola suavemente (*lerp*) hacia la nueva posición en cada fotograma, evitando transiciones bruscas que rompan la percepción de solidez.

5.7.7. Calibración de orientación basada en *keypoints*

Como se explicó en la Sección 5.5.3, la IMU de 6 ejes proporciona una orientación estable en *pitch* (cabeceo) y *roll* (balanceo) gracias a la referencia gravitacional del acelerómetro, pero la componente de *yaw* (guiñada, es decir, la rotación horizontal del cubo visto desde arriba) deriva con el tiempo al carecer de magnetómetro. Los *keypoints* del modelo YOLO-Pose ofrecen la referencia angular necesaria para corregir esta deriva de forma periódica y sin intervención del usuario.

5.7.7.1. Principio de funcionamiento

El algoritmo de calibración compara la dirección que apuntan los *keypoints* de la cara superior del cubo con la dirección *forward* que indica la orientación del sensor IMU, ambas proyectadas sobre el plano horizontal (XZ). Concretamente:

1. A partir de los *keypoints* válidos proyectados en 3D, se calcula un vector de dirección en el plano XZ que representa hacia dónde apunta la cara del cubo según la cámara, delegando el cálculo a `ShapePolicy.ComputeForwardXZ()`.
2. Se proyecta igualmente el eje *forward* del cuaternión del IMU sobre el plano XZ.
3. Se calcula el ángulo de *yaw* de ambos vectores y se obtiene la diferencia: esa diferencia es el error de *yaw* que hay que compensar.
4. Se construye un cuaternión de corrección puramente horizontal (`Quaternion.Euler(0, offset, 0)`) que, multiplicado por la orientación del sensor, alinea su *yaw* con lo que la cámara observa.

5.7.7.2. Gates de seguridad

Antes de aplicar cualquier calibración, tres condiciones deben cumplirse simultáneamente:

1. **Gate de forma:** la `ShapePolicy` confirma que hay suficientes *keypoints* válidos para calibrar (`CanCalibrateYaw`).

2. **Gate de estabilidad del IMU:** la orientación del sensor no ha cambiado significativamente durante un número mínimo de fotogramas consecutivos. Si el cubo se está moviendo, los *keypoints* pueden tener un desfase temporal respecto a la orientación real.
3. **Gate de estabilidad de la cabeza:** la rotación de la cabeza del paciente ha sido menor que un umbral durante al menos `headStableRequiredFrames` fotogramas. La proyección píxel→mundo tiene latencia proporcional al movimiento de cabeza, lo que introduce error en el *yaw* calculado.

5.7.7.3. Tres caminos de calibración

Para aplicar esta corrección de forma robusta ante las diferentes situaciones que se producen durante una sesión de rehabilitación, el sistema implementa tres caminos de calibración que se fueron añadiendo incrementalmente según surgían necesidades durante el desarrollo:

Calibración inicial. La primera vez que el sistema recibe *keypoints* válidos (≥ 3) y la IMU se ha mantenido estable durante al menos 3 fotogramas consecutivos (es decir, la orientación no ha cambiado significativamente entre muestras), se aplica el *offset* de *yaw* de forma inmediata. Se exige un umbral de estabilidad más relajado que en los otros caminos (3 fotogramas en lugar de 8) para que el cubo arranque rápido tras la conexión BLE y el paciente no tenga que esperar. Una vez aplicada, la calibración inicial se marca como completada y no vuelve a ejecutarse.

Calibración post-suelta. Cuando el paciente suelta el cubo después de haberlo manipulado en la mano, es muy probable que la orientación haya cambiado (el paciente puede haber girado el cubo) y que la IMU haya acumulado algo de deriva durante el agarre. Para corregir esto rápidamente, el sistema activa un modo de espera que busca una detección de **alta confianza**: la detección debe pasar el filtro `ShapePolicy.IsHighConfidence()` (por defecto, todos los *keypoints* válidos + validación geométrica positiva) y el error debe ser menor de 90° . Cuando la recibe, aplica la corrección de forma inmediata (sin pasar por el filtro de mediana del camino continuo, para que la corrección sea lo más rápida posible tras la suelta). Si no llega ninguna detección suficientemente buena en 5 segundos, el modo de espera se desactiva y el sistema vuelve al camino continuo. Este *timeout* evita que el sistema quede bloqueado esperando indefinidamente una detección que puede no llegar si las condiciones de iluminación o el ángulo de visión no son favorables.

Calibración continua (*ongoing*). Durante el funcionamiento normal, la deriva de *yaw* se acumula lentamente. Para corregirla de forma suave y sin interrupciones, las muestras de *offset* de *yaw* se acumulan en un **buffer circular** de tamaño configurable (3 muestras por defecto). Cuando el buffer está lleno, el sistema:

1. Calcula la **mediana** del buffer. Se utiliza la mediana en lugar de la media porque es más robusta ante valores atípicos: si una de las tres muestras es errónea, la mediana la ignora mientras que la media se vería arrastrada por ella.
2. Comprueba la **dispersión** del buffer (diferencia entre el valor máximo y el mínimo). Si supera un umbral configurable (20° por defecto), las muestras son inconsistentes entre sí —posiblemente porque el cubo se estaba moviendo o las detecciones eran ruidosas— y se limpia el buffer entero sin aplicar ninguna corrección.

3. Si la mediana es coherente pero menor que una *deadzone* de corrección mínima, no se aplica (la corrección sería tan pequeña que no merece la pena y podría introducir micro-oscilaciones).
4. Si la mediana supera la *deadzone*, se aplica como corrección. Si el error supera los 15°, la corrección se aplica de golpe (porque una deriva tan grande es perceptible y hay que corregirla cuanto antes). Si es menor de 15°, se aplica mediante interpolación suave (*slerp*) a lo largo de varios fotogramas, para que la transición no sea perceptible para el usuario.

Un *cooldown* de 2 segundos entre recalibraciones evita correcciones excesivamente frecuentes que podrían producir oscilaciones.

Nota sobre la simetría del cubo. Durante el desarrollo se probó un mecanismo de *snap* de simetría de 90° heredado de la implementación anterior con códigos QR. Los QR presentan una ambigüedad de simetría: como las cuatro caras del cubo son idénticas, el código QR puede detectarse en cualquiera de ellas, produciendo hasta 4 orientaciones indistinguibles separadas 90°. Para resolver esta ambigüedad se redondeaba la corrección al múltiplo de 90° más cercano. Sin embargo, con *keypoints* la pegatina solo está en una cara, eliminando la ambigüedad. El mecanismo de *snap* se desactivó porque redondeaba errores intermedios (por ejemplo, un error real de 45°) a 0° y bloqueaba las correcciones legítimas.

5.7.8. Estimación de *bias* del giroscopio en tiempo real

Además de la corrección periódica de *yaw* mediante *keypoints*, el FusionTracker incluye un mecanismo opcional de estimación de *bias* del giroscopio en tiempo real. Cuando el cubo está en reposo confirmado (no agarrado, IMU estable) durante más de un retardo configurable (*biasEstimationDelay*, 1,5 s por defecto), el sistema toma una orientación de referencia y mide cuánto deriva el *yaw* con respecto a ella. La tasa de deriva observada se integra con un factor de aprendizaje bajo (*biasLearningRate* = 0,02) para producir una estimación del *bias* en grados/segundo, que se compensa aplicando una rotación inversa en cada fotograma. Este mecanismo complementa la calibración por *keypoints*: reduce la velocidad de acumulación de error entre correcciones visuales, especialmente útil cuando la pegatina no es visible durante periodos prolongados.

5.7.9. Estados cinemáticos y propagación inercial

El sistema alterna entre tres modos cinemáticos en función de la señal FSR, la proximidad de la mano, la disponibilidad visual y el movimiento inercial. Cada modo determina una lógica de actualización de pose cualitativamente diferente:

Estado de superficie ($f_{grip} < T_{enter}$ y sin puente activo). Cuando el cubo está apoyado sobre la mesa, la posición del gemelo digital sigue la posición visual filtrada proveniente de YOLO (con las *deadzones* y el suavizado descritos anteriormente). La rotación se obtiene aplicando el *offset* de calibración al cuaternión de la IMU ($\mathbf{q}_{cal} \otimes \mathbf{q}_{imu}$) y se estabiliza mediante el bloqueo de cara descrito en la Sección 5.7.10.

Estado de agarre FSR. Al detectarse un agarre confirmado por FSR, las actualizaciones visuales de posición se suspenden completamente: no tiene sentido actualizar la posición del cubo a partir de YOLO cuando la mano lo está cubriendo y la detección sería ruidosa o

inexistente. En su lugar, la posición del gemelo digital se ancla al espacio local de la mano del paciente:

$$\mathbf{x}_t = \mathbf{H}_t \oplus \delta_j \quad (5.5)$$

donde $\mathbf{H}_t$ es la pose de la mano en el instante t y δ_j es el *offset* rígido entre la mano y el cubo, calculado en el momento del agarre transformando la posición del cubo al espacio local de la mano ($\delta_j = \mathbf{H}_t^{-1}(\mathbf{x}_t)$). Este *offset* se mantiene constante durante todo el agarre: cuando la mano se mueve, el cubo la sigue conservando exactamente la misma posición relativa. De este modo, si el paciente ha agarrado el cubo un poco descentrado, el gemelo digital respeta esa posición en lugar de saltar al centro de la palma.

La rotación, en cambio, se toma directamente de la IMU calibrada sin ninguna restricción de eje, lo que permite registrar giros 3D completos del objeto en la mano. Esto incluye rotaciones de muñeca, manipulaciones con los dedos o cualquier otro giro que el paciente aplique al cubo mientras lo sostiene. Estas rotaciones son invisibles para los sistemas exclusivamente visuales porque no producen desplazamiento de la mano, solo del objeto dentro de ella.

Estado de puente inercial de oclusión. Cuando no existe presión FSR suficiente, pero la aceleración lineal del firmware indica movimiento y la mano permanece vinculada espacialmente al cubo, el sistema activa el puente inercial de oclusión. En este modo no se entra en un agarre rígido: la posición se propaga mediante el desplazamiento relativo de la mano desde el instante de entrada, mientras la rotación sigue procediendo de la IMU calibrada:

$$\mathbf{x}_t = \mathbf{x}_0^{cube} + (\mathbf{h}_t - \mathbf{h}_0) \quad (5.6)$$

Este modo está pensado para los intervalos en los que el cubo se mueve bajo oclusión y la visión no ofrece una posición fiable. Si aparece una detección YOLO intermedia, el sistema la utiliza para reajustar suavemente el origen de la propagación, pero no abandona automáticamente el puente mientras la IMU siga indicando movimiento. La salida se produce por reposo inercial, pérdida de vínculo espacial con la mano, expiración del tiempo máximo o transición a agarre FSR.

Transición de suelta (FSR). Al soltar el cubo ($f_{grip} < T_{exit}$) desde un agarre FSR, se ejecutan tres acciones:

1. La posición del gemelo digital se fija en el punto donde la mano lo liberó, calculado como $\mathbf{x}_{release} = \mathbf{H}_t \oplus \delta_j$.
2. Se recalcula el *offset* de calibración **completo en 3D** (no solo la componente de *yaw*) como:

$$\mathbf{q}_{cal} = \mathbf{q}_{visual} \otimes \mathbf{q}_{imu}^{-1} \quad (5.7)$$

donde $\mathbf{q}_{visual}$ es la rotación actual del gemelo digital y $\mathbf{q}_{imu}^{-1}$ es la inversa de la orientación actual del sensor. Esta operación captura toda la información rotacional —qué cara está abajo, hacia dónde apunta el cubo— porque durante el agarre el paciente puede haber girado el cubo en cualquier dirección. Si solo se recalculase el *yaw*, se perdería la información de qué cara quedó hacia abajo tras la manipulación.

3. Se activa el modo de calibración post-suelta para corregir la posible deriva de *yaw* acumulada durante el agarre.

Salida del puente inercial. Al finalizar el puente inercial de oclusión, la posición actual del gemelo digital se conserva como punto de continuidad, pero **no** se recalibra el *offset* de orientación. El puente no representa una suelta física confirmada, sino una propagación temporal durante una pérdida de observación visual. Por ello, cualquier corrección de deriva residual se delega al camino de calibración continua mediante *keypoints* cuando el objeto vuelve a ser visible y estable.

5.7.10. Bloqueo de cara inferior

Cuando el cubo reposa sobre la mesa, la señal de la IMU contiene ruido residual que, aunque pequeño, puede hacer que el gemelo digital oscile ligeramente en *pitch* y *roll*. En la vida real, un cubo apoyado en una mesa no se inclina: tiene una de sus caras perfectamente plana contra la superficie. Para reproducir este comportamiento y que el gemelo digital se perciba como un objeto sólido y estable, el sistema implementa un mecanismo de **bloqueo de cara** (*face-down lock*), delegado a la ShapePolicy.

El algoritmo de la CubeShapePolicy funciona de la siguiente manera:

1. Se evalúa cuál de los 6 ejes locales del cubo (+X, -X, +Y, -Y, +Z, -Z) apunta más hacia abajo en el espacio del mundo, calculando el producto escalar de cada eje transformado con el vector de gravedad (0, -1, 0). El eje con mayor producto escalar es el que más cerca está de apuntar hacia abajo.
2. Se aplica **histéresis**: el eje bloqueado solo cambia si otro eje supera al actual por un margen significativo (0,15 en el producto escalar). Esto evita que, cuando el cubo está en un ángulo límite entre dos caras, el sistema oscile entre una y otra por el ruido del sensor.
3. Se calcula la rotación que alinea el eje bloqueado exactamente con el vector de gravedad.
4. Del resto de la rotación se extrae únicamente la componente de *yaw*, descartando las de *pitch* y *roll*.
5. Se reconstruye la rotación final como la composición de la alineación con la gravedad y el *yaw* puro.

El resultado es que el cubo virtual queda perfectamente plano contra la superficie y solo puede girar horizontalmente sobre sí mismo, imitando el comportamiento físico de un cubo real sobre una mesa. Durante el estado de agarre, el bloqueo se desactiva y la IMU proporciona la rotación 3D completa sin restricciones.

5.7.11. Filtro de falsos positivos de mano

Durante las sesiones de prueba se detectó que, bajo ciertas condiciones de iluminación, el modelo YOLO confundía el puño cerrado o semicerrado del paciente con un cubo. Esto generaba detecciones intermitentes que hacían saltar la posición del gemelo digital entre su ubicación real sobre la mesa y la posición de la mano del paciente, produciendo un parpadeo indeseable.

El filtro de falsos positivos evolucionó durante el desarrollo. La versión inicial comprobaba únicamente la distancia entre la detección y la mano con el estado binario del FSR. La versión actual delega la decisión al CubeContactAssessor, que dispone de mucha más información. El *manager* notifica al *tracker* la posición de cada detección YOLO **antes** de evaluar el filtro, para que el assessor pueda calcular la distancia mano↔YOLO de ese cubo concreto. A continuación consulta ShouldFilterAsHandFP, que devuelve **true** **solo** cuando el nivel de

contacto es Nearby (mano cerca sin más evidencia). En cualquier nivel superior (Touching, HeldSoft, HeldHard), la detección pasa porque hay evidencia convergente de que el cubo está realmente ahí.

Este filtro ejemplifica la filosofía general del sistema de fusión: cada fuente de información tiene puntos débiles, pero cruzando las fuentes entre sí se pueden resolver ambigüedades que ninguna resolvería por separado.

5.7.12. Auto-escalado del gemelo digital

Los cubos fabricados mediante impresión 3D no son perfectamente cúbicos: las dimensiones reales típicas pueden ser, por ejemplo, $3,0 \times 3,0 \times 3,5$ cm. Al rotar el gemelo digital con los datos de la IMU, esta asimetría se hace visible si la *mesh* virtual tiene proporciones diferentes a las del cubo físico: las aristas no coinciden con las del objeto real y la rotación se percibe como incorrecta. Para corregirlo, la CubeShapePolicy implementa el método ApplyRealDimensionScaling: al inicializarse, calcula la escala necesaria usando las dimensiones de la *mesh* base en coordenadas del mundo (lossyScale) y las dimensiones reales medidas con calibrador, ajustando localScale eje por eje.

Algoritmo 5.1: Asociación de detecciones visuales con cubos instrumentados

Entrada: Detecciones visuales D_t , conjunto de *trackers* T , pose de mano H_t
Estado: Clase, color, posición visual, estado IMU y evaluador de contacto de cada cubo

```

for cada detección  $d_i \in D_t$  do
     $\tau_j \leftarrow$  tracker con clase y color coincidentes;
    if no existe correspondencia inequívoca then
         $\tau_j \leftarrow$  respaldo cinemático por proximidad mano–detección e IMU móvil única;
    end
    if todavía no existe correspondencia inequívoca then
         $\tau_j \leftarrow$  candidato razonable no reclamado por eliminación;
    end
    if  $\tau_j = \emptyset$  then
        con la siguiente detección;
    end
    Notificar posición YOLO a  $\tau_j$ .ContactAssessor;
    if  $\tau_j$ .ContactAssessor.ShouldFilterAsHandFP then
        descartar  $d_i$ ;                                     // falso positivo de mano
    end
    else
        Enviar posición y keypoints a  $\tau_j$ ;
    end
end

```

Algoritmo 5.2: Fusión visuo-inercial, validación táctil y puente de oclusión

Entrada: Cuaternión IMU $\mathbf{q}_{imu}$, aceleración lineal a_{lin} , señal FSR f_{grip} , pose de mano $\mathbf{H}_t$
Estado: *trackers* T , offset de calibración $\mathbf{q}_{cal}$, estado de agarre y estado del puente inercial

```

for cada tracker  $\tau_j \in T$  do
     $\mathbf{q}_{raw}, a_{lin}, f_{grip} \leftarrow$  leer paquete IMU extendido y FSR desde BLE;
    Aplicar rotación de montaje y compensar bias;
    Interpolar  $\mathbf{q}_{cal}$  hacia su valor objetivo;
    Actualizar estabilidad IMU y estabilidad de cabeza;
     $m_{imu} \leftarrow$  combinar delta angular y aceleración lineal;
    Alimentar ContactAssessor con FSR,  $m_{imu}$ , mano y YOLO;

    if  $\tau_j.grippingFSR$  then
        if  $FSR < T_{exit}$  then
            Soltar el cubo;
             $\mathbf{q}_{cal} \leftarrow \mathbf{q}_{visual} \otimes \mathbf{q}_{raw}^{-1}$ ;
            Activar calibración post-suelta;
            Detener puente inercial si estuviera activo;
        end
    end
    else
        if  $FSR \geq T_{enter}$  then
            Detener puente inercial si estuviera activo;
            Entrar en agarre FSR;
             $\delta_j \leftarrow \mathbf{H}_t^{-1}(\tau_j.\mathbf{x})$ ;
        end
        else if  $mano\ trackeada \wedge mano\ próxima/cobertura\ reciente \wedge a_{lin} \geq T_{bridge} \wedge no\ existe\ otro$ 
             $candidato\ mejor$  then
                Activar o mantener puente inercial de oclusión;
                 $\tau_j.\mathbf{x} \leftarrow \mathbf{x}_0^{cube} + (\mathbf{h}_t - \mathbf{h}_0)$ ;
            end
        else if  $puente\ activo \wedge (a_{lin} \leq T_{rest} \vee mano\ alejada \vee tiempo\ máximo)$  then
            Detener puente;
            Mantener posición y offset actuales;
        end
    end

    if  $\tau_j.grippingFSR$  then
         $\tau_j.\mathbf{x} \leftarrow \mathbf{H}_t \oplus \delta_j$ ;
         $\tau_j.\mathbf{q} \leftarrow \mathbf{q}_{cal} \otimes \mathbf{q}_{raw}$ ;
    end
    else if  $\tau_j.bridgeActivo$  then
         $\tau_j.\mathbf{q} \leftarrow \mathbf{q}_{cal} \otimes \mathbf{q}_{raw}$ ;
    end
    else
         $\tau_j.\mathbf{x} \leftarrow$  FiltroDeadzone( $\mathbf{p}_{yolo}$ );
         $\tau_j.\mathbf{q} \leftarrow$  ShapePolicy.SnapRestingPose( $\mathbf{q}_{cal} \otimes \mathbf{q}_{raw}$ );
        Ejecutar calibración de yaw si hay keypoints y se cumplen las condiciones de validación;
    end
end

```

5.8. MÓDULO DE JUEGO SERIO

5.8.1. Objetivo terapéutico

La capa de aplicación implementa un **juego de persecución** (*chase game*) diseñado para motivar la manipulación repetida de los cubos bajo presión temporal progresiva. La decisión de diseñar una mecánica de juego específica en lugar de limitarse a instrumentar el BBT estándar responde a dos motivaciones: en primer lugar, el BBT es una prueba de evaluación que se administra de forma puntual y no está pensada como ejercicio terapéutico continuado; en segundo lugar, la gamificación de la tarea permite introducir componentes cognitivos y de velocidad de reacción que complementan el trabajo motor puro y ofrecen al terapeuta una fuente adicional de métricas sobre la capacidad funcional del paciente.

El juego persigue un doble objetivo terapéutico: por un lado, la aparición de plataformas en posiciones aleatorias con cuenta atrás decreciente ejerce las funciones de atención sostenida, velocidad de procesamiento y seguimiento visual; por otro, la necesidad de alcanzar, transportar y colocar los cubos físicos sobre posiciones concretas dentro del límite de tiempo trabaja la coordinación visuomotora, la estabilidad del agarre y la precisión de la suelta.

5.8.2. Calibración del espacio de trabajo

Antes de iniciar la partida, el juego ejecuta una **calibración del espacio de trabajo** del paciente para adaptar el tamaño y la posición del área de juego a su rango de movilidad real. Esto es especialmente importante en pacientes con limitaciones articulares que podrían no alcanzar un área de tamaño fijo.

El procedimiento consta de los siguientes pasos guiados por instrucciones en pantalla y retroalimentación audiovisual:

1. **Selección de mano dominante.** El paciente (o el terapeuta) selecciona la mano que se usará durante la sesión mediante botones en la interfaz (*Calibrar Derecha / Calibrar Izquierda*). Esta elección se propaga al `FusionSystemManager`, que la asigna como mano activa para todos los *trackers* del sistema de fusión.
2. **Captura de tres puntos de referencia.** Se le pide al paciente que extienda la mano seleccionada hacia tres posiciones sucesivas: máximo alcance a la derecha, centro (frente a él) y máximo alcance a la izquierda. En cada posición, debe mantener la mano quieta durante 5 segundos. El sistema monitoriza el movimiento de la mano y, si detecta un desplazamiento mayor que un umbral de quietud (4 cm), reinicia el temporizador. Una esfera de previsualización muestra al paciente la posición de su mano y crece progresivamente durante la captura, proporcionando retroalimentación visual del progreso. Tras capturarlo, se valida que esté suficientemente separado del punto anterior (al menos 15 cm) para evitar que un área colapsada sea inutilizable. El sistema reproduce instrucciones audiovisuales a través del `TutorialMediaManager` para guiar al paciente en cada paso.
3. **Instanciación del área de juego.** Con los tres puntos capturados, el sistema calcula la anchura disponible del espacio de trabajo como la distancia horizontal entre el punto derecho y el izquierdo. La profundidad se calcula como un 25 % de la anchura. La altura de la mesa se detecta automáticamente mediante el módulo **MRUK** (*Mixed Reality Utility Kit*) de Meta, con *fallback* a la tabla de alturas de los *trackers* de fusión si están calibrados, o en última instancia a un *raycast* hacia abajo. El área se instancia en la posición del punto central, a la altura de la mesa detectada (con un pequeño *offset*

configurable para que las plataformas queden ligeramente por encima de la superficie), orientada hacia el paciente.

Esta calibración garantiza que las plataformas quedan al alcance del paciente independientemente de su posición, altura o limitación de movilidad.

5.8.3. Mecánica del juego: modo persecución

A diferencia de la mecánica de memoria con múltiples plataformas y fases de memorización/ocultación/mezcla diseñada inicialmente, la mecánica actual se estructura como un juego de **persecución temporal** (*chase game*) que presenta una sola plataforma activa a la vez con cuenta atrás adaptativa. Cada ronda se desarrolla de la siguiente manera:

1. **Aparición.** Una plataforma de un color aleatorio (rojo o azul) aparece en una posición aleatoria dentro del espacio de trabajo calibrado. Un temporizador 3D flotante sobre la plataforma muestra la cuenta atrás en segundos.
2. **Interacción.** El paciente debe localizar el cubo físico del color indicado, agarrarlo, transportarlo y colocarlo sobre la plataforma antes de que expire el temporizador. El sistema detecta la colocación comparando la identidad de color del cubo (proporcionada por el FusionTracker) con el color asignado a la plataforma.
3. **Acierto.** Si la colocación es correcta, la plataforma se oculta y un **orbe VFX** viaja con una trayectoria en arco desde la plataforma acertada hasta la posición donde aparecerá la siguiente, proporcionando una guía visual inmersiva. La cuenta atrás para la siguiente plataforma se **reduce** en un incremento configurable (1 segundo por defecto), hasta un mínimo configurable.
4. **Fallo.** Si el temporizador llega a cero o se coloca un cubo del color incorrecto, se aplica una penalización, se rompe la racha de *combo*, y aparece una nueva plataforma.

Cada ronda comprende un número configurable de plataformas por ciclo (`targetPerCycle`, 10 por defecto). La dificultad progresiva se materializa en la reducción de la cuenta atrás: la primera plataforma dispone de 20 segundos y cada acierto consecutivo resta 1 segundo, pudiendo llegar a un mínimo de 2 segundos. Esta mecánica obliga al paciente a aumentar progresivamente la velocidad y precisión de sus movimientos, ejercitando simultáneamente la velocidad de procesamiento cognitivo y la coordinación motora. La partida finaliza al completar todas las rondas configuradas, o puede jugarse en modo infinito si se configura el número de rondas a cero.

5.8.4. Sistema de puntuación y retroalimentación

El sistema de puntuación está diseñado para reforzar tanto la precisión como la consistencia del paciente. La retroalimentación al paciente se proporciona a través de múltiples canales sensoriales: efectos de sonido diferenciados para aciertos y errores, música de fondo durante el juego, efectos de partículas (VFX) al aparecer cubos y completar rondas, el orbe animado que guía hacia la siguiente plataforma, el temporizador 3D flotante con el color de la plataforma, y una interfaz de texto que muestra la puntuación, los aciertos, los fallos y el tiempo transcurrido. Al finalizar el juego, se ofrecen los botones de calibración para iniciar una nueva partida.

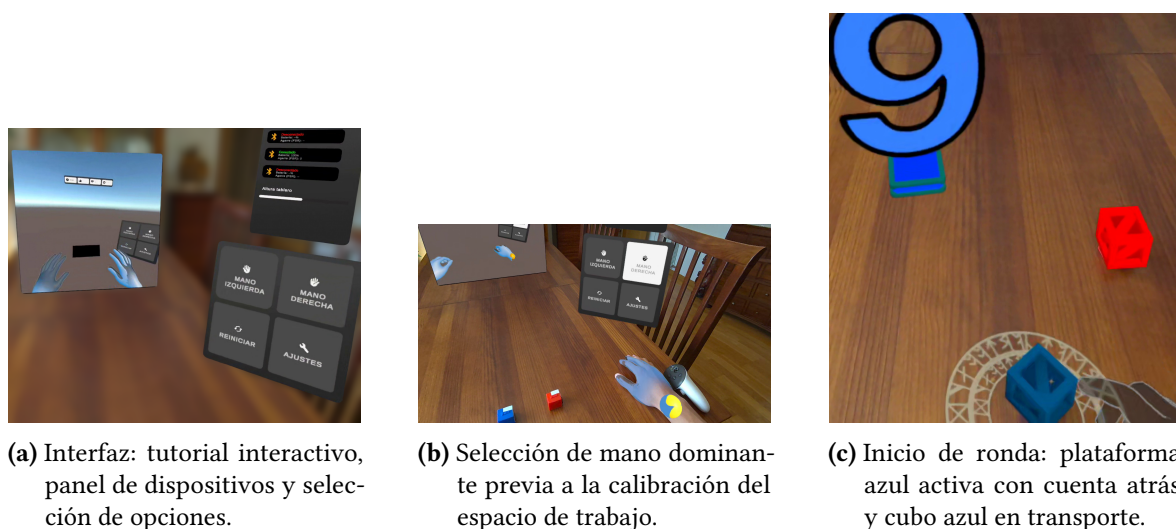


Figura 5.6: Configuración e inicio del juego serio de persecución temporal.



Figura 5.7: Mecánica del juego serio: colocación, acierto, penalización y transición entre rondas.

5.9. MÓDULO DE REGISTRO DE DATOS

5.9.1. Motivación y arquitectura

La instrumentación de los objetos terapéuticos genera un volumen considerable de información cinemática y sensorial durante cada sesión de rehabilitación. Sin un mecanismo de persistencia, estos datos se perderían al finalizar la sesión, desaprovechando el potencial de los objetos sensorizados como herramienta de evaluación clínica objetiva. El módulo de registro (`BackgroundDataLogger`) resuelve esta necesidad almacenando de forma continua y asíncrona los datos generados por todos los módulos del sistema en archivos CSV estructurados, que quedan disponibles en el almacenamiento del visor para su análisis posterior.

La arquitectura del módulo sigue el patrón *Singleton* y delega toda la escritura a disco en hilos secundarios mediante instancias de `AsyncCsvLogger`, lo que garantiza que la operación de E/S no bloquee el hilo principal de Unity ni afecte a la tasa de refresco del visor. Al iniciar una sesión, el módulo crea una carpeta con marca temporal (`SESSION_yyyyMMdd_HHmmss`) y dentro de ella abre los siguientes archivos de registro:

- **sensor_imu.csv:** cuaternión de orientación, aceleración lineal cruda y valor de agarre bruto de cada tracker, a la frecuencia de notificación BLE (≈ 33 Hz para IMU, ≈ 10 Hz para FSR).

- **vision_data.csv**: detecciones YOLO filtradas con *bounding box*, puntuación de confianza, coordenadas de *keypoints* y estado de calibración. Se registra solo cuando el modo de experimentación está activo.
- **vision_data_all.csv**: flujo bruto de todas las detecciones visuales, siempre activo independientemente del modo de experimentación. Permite la depuración y el *replay* completo de la sesión.
- **fusion_result.csv**: posición y orientación fusionada del gemelo digital, estado de agarre y cara inferior bloqueada.
- **cube_state.csv**: registro completo del estado de cada cubo por fotograma, incluyendo posición (XYZ), rotación (cuaternión y ángulos de Euler), velocidad lineal y angular, nivel de contacto del CubeContactAssessor, fuente de seguimiento (*fsr*, *imu_bridge* o *none*), delta angular de la IMU, y vector de cara inferior bloqueada.
- **patient_kinematics.csv**: posición y orientación de la cabeza y ambas manos del paciente, proporcionadas por el sistema de seguimiento nativo de Meta SDK.
- **game_events.csv**: eventos del juego serio (inicio de ronda, colocación correcta/incorrecta, *timeout*, fin de partida) con puntuación, *combo* y colores esperado/colocado.
- **interaction_segments.csv**: segmentos automáticos de interacción delimitados por las transiciones de agarre, con métricas agregadas por segmento (véase la Sección 5.9.2).

5.9.2. Segmentación automática de interacciones

Además del registro continuo, el módulo implementa una segmentación automática de las interacciones del paciente con los objetos. Cada vez que el sensor FSR detecta una transición de no-agarre a agarre, se inicia un nuevo segmento de interacción. Mientras dura el agarre, el módulo acumula la longitud de trayectoria recorrida, la velocidad media e instantánea, y la posición de inicio. Al detectarse la suelta (transición de agarre a no-agarre), se cierra el segmento y se registran las métricas agregadas: posición de inicio y fin, duración, longitud de trayectoria, velocidad media y máxima, nivel de contacto final, y tipo de suelta.

El tipo de suelta se clasifica automáticamente en tres categorías mediante el método `ClassifyReleaseType`:

- **Placement**: suelta controlada con velocidad inferior a 0,05 m/s y contacto significativo con la superficie. Indica una colocación deliberada del objeto.
- **Drop**: velocidad de suelta superior a 0,20 m/s. Indica una caída o suelta brusca.
- **Release**: suelta genérica que no cumple ninguno de los criterios anteriores.

Si la sesión se cierra mientras un segmento está activo, se finaliza con tipo `session_end`. Esta segmentación permite al terapeuta analizar cada intento de manipulación de forma aislada, identificando patrones de mejora o deterioro a lo largo de la sesión sin necesidad de revisar manualmente los datos continuos.

Experimentación y resultados

Antes de su despliegue en un entorno clínico, resulta necesario establecer una línea base técnica que permita comprobar que el sistema satisface los requisitos mínimos de continuidad, latencia y precisión exigibles en una tarea de rehabilitación basada en manipulación tangible. Esta validación es especialmente relevante porque el sistema combina visión por computador, sensores inerciales, señal táctil y seguimiento de manos en una misma arquitectura de fusión.

El capítulo se organiza en dos bloques. En primer lugar, se presenta la validación técnica del seguimiento visuo-inercial, analizando la latencia de comunicación, el rendimiento del detector visual, la deriva rotacional de la IMU y la robustez de la fusión frente a oclusiones. En segundo lugar, se describen las pruebas preliminares realizadas en el Hospital Nacional de Paraplégicos de Toledo, donde se evalúa la viabilidad del sistema con pacientes, perfiles profesionales del entorno hospitalario y personas sin patología motora conocida.

Como material complementario, se ha incluido en el repositorio una carpeta con los registros CSV originales generados durante las sesiones experimentales. Estos ficheros constituyen la traza bruta de ejecución del sistema y permiten conservar la evidencia completa de las pruebas realizadas. No obstante, dado que su análisis manual resulta poco práctico y queda fuera del alcance principal de esta memoria, la interpretación de los resultados se apoya en los informes derivados por participante, incluidos también en el repositorio, donde se resumen las métricas funcionales, cinemáticas y técnicas utilizadas en esta sección. Dicha carpeta se referencia en el Anexo A.

6.1. VALIDACIÓN TÉCNICA DEL SISTEMA

La validación técnica se centró en cuatro aspectos críticos para una tarea de manipulación tangible en realidad mixta: latencia de comunicación, rendimiento visual, deriva rotacional de la IMU y robustez de la fusión frente a oclusiones.

6.1.1. Latencia de comunicación

La latencia extremo a extremo se midió mediante un protocolo *Round-Trip Time* (RTT) de tipo *ping-pong* sobre $N = 1121$ eventos, estimando la latencia unidireccional como $RTT/2$. Durante la manipulación activa, el *firmware* negoció un intervalo de conexión BLE de 7,5 ms. La Tabla 6.1 recoge el resumen estadístico empleado para valorar la viabilidad temporal de la comunicación.

La latencia media de 34,15 ms se sitúa dentro del rango de ≤ 50 ms considerado aceptable para tareas de manipulación de objetos [2]. Aunque supera el umbral de 20 ms asociado a una respuesta prácticamente indistinguible [17], la latencia mínima confirma el funcionamiento del modo BLE de baja latencia. El *jitter* inalámbrico ($\sigma \approx 10$ ms) se atenúa visualmente

Tabla 6.1: Latencia unidireccional estimada a partir de RTT ($N = 1\,121$).

Métrica	Tiempo (ms)
Mínimo	11,84
Media	34,15
Mediana	32,60
Desv. típica	10,28
Percentil 95	56,35

mediante interpolación *Slerp*, evitando saltos perceptibles en la representación del cubo [27]. Además, el protocolo RTT puede considerarse conservador, ya que obliga al visor a responder a paquetes de prueba mientras ejecuta simultáneamente visión por computador y lógica de interacción, una sobrecarga mayor que la transmisión normal de telemetría durante el juego.

6.1.2. Rendimiento del subsistema de visión

El subsistema visual utiliza YOLOv8-Pose nano ejecutado en Meta Quest 3 para detectar dos cubos activos y cuatro *keypoints* en la cara superior de cada cubo. Estos puntos permiten validar geométricamente la detección y estimar la orientación horizontal del objeto, sustituyendo los marcadores QR empleados en versiones previas. La evaluación se realizó durante sesiones de juego real con los dos cubos sensorizados activos simultáneamente, segmentando las muestras según el estado de agarre: las muestras sin agarre representan condiciones de mayor visibilidad, mientras que las muestras con agarre activo aproximan situaciones de oclusión parcial o total por la mano.

La cadencia del modelo se estimó mediante:

$$\text{FPS}_{\text{model}} = \frac{1}{\text{mediana}(\Delta t_{\text{det}})}, \quad (6.1)$$

donde Δt_{det} es el intervalo entre detecciones consecutivas válidas. La antigüedad del *bounding box* se calculó como:

$$\text{BBoxAge} = t_{\text{actual}} - t_{\text{ultima_det}}, \quad (6.2)$$

y la tasa de obsolescencia $\text{Stale}_{0,5}$ se definió como la proporción de muestras con $\text{BBoxAge} > 0,5$ s.

Tabla 6.2: Rendimiento de YOLOv8-Pose nano durante sesiones de juego real.

Fase	FPS (Hz)	BBoxAge (s)	Stale _{0,5} (%)	Reacq. (s)
Blue visible	5,13	0,61	29,8	—
Blue oclusión	5,13	0,41	22,6	0,59
Red visible	5,14	0,69	16,1	—
Red oclusión	5,14	0,51	34,5	0,49

La cadencia mediana de detección fue aproximadamente 5,1 Hz, superior a la versión anterior basada en YOLOv11-nano sin *keypoints*. Durante el agarre, la visión puede degradarse por oclusión parcial o total de la mano, pero la reacquisición tras pérdida visual se mantuvo entre 0,49 y 0,59 s. Esta latencia de reacquisición es asumible porque, durante la pérdida visual, la pose del cubo se mantiene mediante el anclaje a la mano y la IMU. La obsolescencia

observada no impide la continuidad del seguimiento, sino que justifica precisamente la necesidad de la arquitectura híbrida: la visión identifica y corrige, mientras que la fusión mantiene la pose durante los intervalos en los que la cámara no ofrece una referencia fiable.

6.1.3. Deriva de la IMU y corrección mediante *keypoints*

La IMU proporciona orientación continua, pero la componente *yaw* deriva al carecer de una referencia absoluta. El sistema corrige dicha deriva cuando la visión ofrece una orientación fiable a partir de los *keypoints*. Dado que el cubo presenta simetría de 90°, el error perceptual se calcula como:

$$\varepsilon_{\text{visual}} = \min(|\theta_{\text{IMU}} - \theta_{\text{KP}}| \bmod 90^\circ). \quad (6.3)$$

El sistema registra el error cada vez que se aplica una corrección de *yaw*, clasificando la corrección según el camino que la desencadena. **Initial** corresponde a la primera calibración tras la conexión BLE; **Post-release** se aplica al soltar el cubo y mide la deriva acumulada durante el agarre; y **Ongoing** agrupa las correcciones periódicas cuando el cubo reposa sobre la mesa con los *keypoints* visibles.

Tabla 6.3: Error de registro visual con corrección de simetría.

Métrica	CubeBlue (°)	CubeRed (°)	Agregado (°)
Mediana	9,20	10,98	9,88
Media	8,95	14,37	12,09
RMSE	10,45	16,77	14,45
Máximo	31,71	35,49	—
<i>N</i> correcciones	50	69	119

Tabla 6.4: Desglose por camino de calibración, ambos cubos agregados.

Camino	<i>N</i>	Mediana (°)	Media (°)	Máx. (°)
Initial	2	7,94	7,94	9,11
Post-release	52	11,06	12,59	35,49
Ongoing	65	9,29	11,17	34,01

La mediana global de 9,88° es comparable a la obtenida previamente con calibración QR (8,67°), por lo que la sustitución de marcadores físicos por *keypoints* mantiene una precisión angular similar. El error *post-release* es ligeramente superior al *ongoing*, lo cual resulta coherente porque acumula la deriva durante el periodo de agarre, cuando el cubo puede estar ocluido. La frecuencia de corrección, entre 12 y 17 correcciones por minuto, resultó suficiente para mantener el *yaw* dentro de márgenes compatibles con la tarea. CubeRed presentó más correcciones instantáneas superiores a 15° que CubeBlue, posiblemente por diferencias de sesgo del giroscopio o por patrones de manipulación más exigentes.

6.1.4. Estrategia de fusión y robustez ante oclusiones

La fusión combina cuatro fuentes: detección visual para identificación, reacquisición y corrección angular; FSR para confirmar contacto físico; anclaje a la mano para mantener la posición durante el agarre; e IMU para conservar la orientación intrínseca del objeto. Esta combinación evita dos problemas habituales en seguimiento exclusivamente visual:

el desplazamiento del cubo hacia la palma cuando la mano lo ocluye y los falsos agarres asociados a posturas de semiflexión o pinza por tenodesis.

El FSR proporciona una señal táctil determinista que reduce la adhesión falsa del cubo a la mano, pero no se usa como requisito único de agarre. El puente inercial de oclusión permite mantener la continuidad espacial del gemelo digital aunque el FSR no se active, siempre que exista movimiento inercial confirmado y proximidad entre la mano y el cubo; internamente, el CubeContactAssessor puede alcanzar el nivel HeldSoft por convergencia de señales débiles (proximidad, movimiento IMU y detección visual), lo que a su vez puede contribuir a activar el puente. Esta redundancia es importante en pacientes con agarres atípicos, ya que pueden sostener el cubo sin ejercer presión directamente sobre la zona sensorizada.

Para cuantificar la robustez, se comparó el seguimiento exclusivamente visual con la fusión híbrida en un circuito de traslación y rotación en mano bajo condiciones de visibilidad creciente: visible, oclusión parcial, oclusión total y rotación del cubo con los dedos. Un *tracker* montado en un controlador actuó como *ground truth*. En traslación se midió la dispersión espacial media $\sigma_{xyz} = (\sigma_x + \sigma_y + \sigma_z)/3$; en rotación se midió el barrido angular de *yaw* y el *jitter* posicional.

Tabla 6.5: Comparación entre seguimiento visual y fusión híbrida.

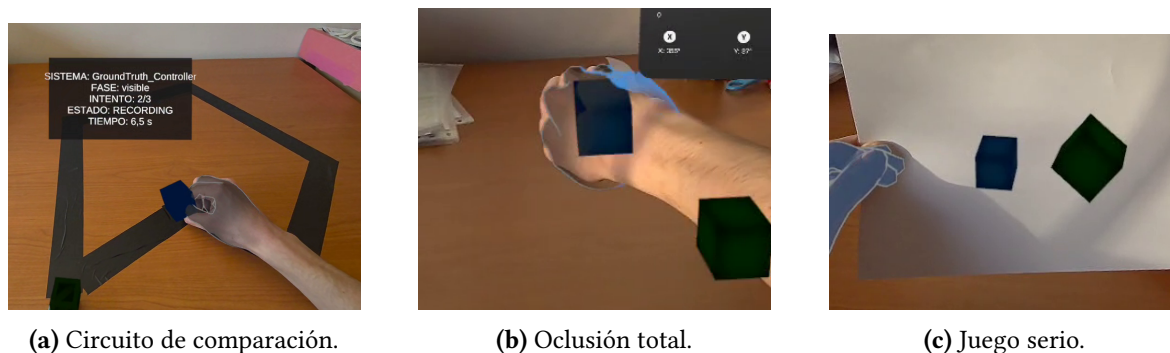
Condición	Métrica	Ground truth	Solo visión	Fusión
Visible	σ_{xyz} (cm)	11,8	11,2	11,5
Oclusión parcial	σ_{xyz} (cm)	11,6	7,1	10,8
Oclusión total	σ_{xyz} (cm)	11,5	3,2	13,5
Rotación en mano	Barrido yaw (°)	358,8	50,0	314,1
Rotación en mano	Jitter YZ (cm)	1,8	3,7	1,8

Bajo oclusión total, el sistema visual tiende a congelar la pose ($\sigma_{xyz} = 3, 2$ cm), mientras que la fusión mantiene una dispersión comparable al *ground truth*. En rotación en mano, la visión solo captura 50° del giro real, ya que la muñeca apenas rota, mientras que la fusión registra 314,1°. Esta diferencia es especialmente relevante para rehabilitación, porque los pacientes pueden producir movimientos finos del objeto con los dedos sin que exista una rotación equivalente de la muñeca. Estos resultados proceden de la versión previa con QR, pero siguen siendo representativos de la arquitectura actual porque la lógica de fusión no se ha modificado; el cambio afecta únicamente a la fuente de corrección angular, validada en la Sección 6.1.3.

La comparación no pretende valorar únicamente si el cubo se ve o no se ve, sino si la pose mantiene continuidad durante la acción funcional. En este sentido, un valor anormalmente bajo de dispersión en el sistema visual durante la oclusión total no indica mejor seguimiento, sino congelación de la última pose válida. La fusión, en cambio, conserva el movimiento espacial y permite que el objeto siga acompañando a la mano hasta que la visión vuelve a estar disponible.

6.2. PRUEBAS CON USUARIOS

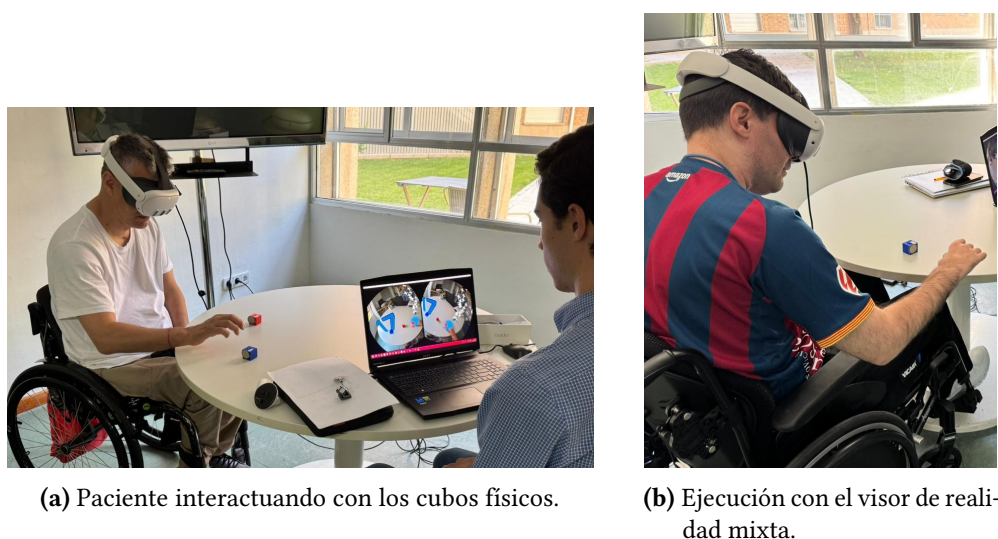
Tras la validación técnica, se realizó una evaluación preliminar de viabilidad con seis participantes en el Hospital Nacional de Paraplégicos de Toledo. El objetivo no fue obtener evidencia clínica generalizable, sino comprobar que el sistema podía ser utilizado por pacientes reales, que el ejercicio resultaba comprensible y que el registro generaba métricas funcionales, cinemáticas y técnicas analizables.



(a) Circuito de comparación.

(b) Oclusión total.

(c) Juego serio.

Figura 6.1: Comparación de estrategias de seguimiento, oclusión total e interfaz del juego serio.

(a) Paciente interactuando con los cubos físicos.

(b) Ejecución con el visor de realidad mixta.

Figura 6.2: Pruebas realizadas en el Hospital Nacional de Paraplégicos de Toledo.

6.2.1. Participantes y registros analizados

Todos los participantes fueron evaluados en el hospital y utilizaron la mano derecha. El grupo incluyó dos pacientes con lesión medular (P1 y P2), una profesional con perfil de ingeniería biomédica (T1), una terapeuta ocupacional (T2) y dos personas sin patología motora conocida (U1 y U2), estas últimas como referencia funcional descriptiva y no como grupo de control estadístico. El número de pacientes estuvo condicionado por la planificación diaria de actividades terapéuticas del centro, por lo que esta fase debe interpretarse como una evaluación preliminar de viabilidad, no como un estudio clínico inferencial.

Tabla 6.6: Participantes y registros continuos disponibles.

ID	Perfil	Exp. previa RV	Mano	Rondas
P1	Paciente con LME	Frecuente (6–20)	Derecha	3
P2	Paciente con LME	Ocasional (1–5)	Derecha	2
T1	Ing. biomédica	Muy frecuente (>20)	Derecha	5
T2	Terapeuta ocupacional	Frecuente (6–20)	Derecha	3
U1	Persona sana	Frecuente (6–20)	Derecha	2
U2	Persona sana	Ocasional (1–5)	Derecha	2

La columna “Rondas” indica el número de rondas con registro continuo utilizadas en el análisis. Algunas sesiones contenían rondas adicionales con interrupciones temporales o

carpetas parciales; se excluyeron únicamente por criterios de calidad del registro, no por el perfil del participante. En particular, no se descartó a ningún participante por ser externo al personal clínico: todos los registros proceden del mismo entorno hospitalario y se diferencian solo para contextualizar la interpretación. Cada ronda constó de 10 objetivos y el sistema registró eventos del juego, cinemática, detecciones visuales, muestras inerciales, resultados de fusión y calibración.

La categorización de los participantes en la Tabla 6.6 diferencia perfiles clínicos y técnicos para contextualizar los resultados; así, T1 representa un perfil de ingeniería biomédica, mientras que T2 aporta la perspectiva clínica de la terapia ocupacional. Por su parte, U1 y U2 se mantienen en el análisis porque realizaron la prueba en las mismas condiciones físicas y con el mismo protocolo, con la función de aportar una referencia descriptiva de ejecución sin afectación motora conocida.

6.2.2. Rendimiento funcional

La Tabla 6.7 resume el rendimiento en las rondas con registro continuo. Para facilitar la revisión, se incluye como material complementario una carpeta con los informes de resultados de cada participante, referenciada en el Anexo A.2. Estos informes contienen el resumen funcional, cinemático y técnico utilizado para generar las tablas de esta sección. No se presentan como un *pipeline* de análisis reproducible ni como parte del software desarrollado, sino como documentación auxiliar para interpretar de forma ordenada los registros de las sesiones.

Tabla 6.7: Rendimiento funcional por participante.

ID	Perfil	Rondas	Tiempo (s)	Aciertos	Err.	TO	Precisión	Ritmo	Combo
P1	Paciente con LME	3	133,43	30	5	1	83,3 %	13,5	9
P2	Paciente con LME	2	117,77	20	2	1	87,0 %	10,2	8
T1	Ing. biomédica	5	180,52	50	9	0	84,7 %	16,6	18
T2	Terapeuta ocupacional	3	102,65	30	2	0	93,8 %	17,5	13
U1	Persona sana	2	64,75	20	0	0	100,0 %	18,5	20
U2	Persona sana	2	45,28	20	1	0	95,2 %	26,5	11

Err. = errores de color; TO = *timeouts*; Tiempo = duración efectiva acumulada; Ritmo = aciertos/min;

Precisión = $\text{Aciertos} / (\text{Aciertos} + \text{Err.} + \text{TO}) \cdot 100$.

Los seis participantes completaron los objetivos de sus rondas analizadas, con 170 aciertos sobre 170 objetivos. Los *timeouts* fueron escasos y aparecieron solo en P1 y P2, mientras que los errores de color constituyeron el fallo predominante. Los pacientes presentaron ritmos menores que los perfiles profesionales y las personas sin patología motora, lo que resulta coherente con una ejecución más pausada y con mayor necesidad de corrección. La inclusión del tiempo efectivo es relevante porque una mayor trayectoria acumulada puede deberse simplemente a mayor permanencia en la tarea; por ello, el análisis cinemático utiliza también trayectoria por acierto y velocidad media. U2 alcanzó el mayor ritmo, mientras que P2 mostró el ritmo más bajo, influido por una segunda ronda más lenta con un error de color y un *timeout*.

6.2.3. Métricas cinemáticas

La trayectoria por acierto normaliza la distancia recorrida respecto a la carga de trabajo. U1 presentó el valor más bajo (0,77 m/acierto), mientras que P1 y P2 mostraron valores superiores (1,38 y 1,64 m/acierto), compatibles con trayectorias menos directas, mayor exploración del espacio o más correcciones. Sin embargo, estos valores no deben interpretarse de forma

Tabla 6.8: Métricas cinemáticas por participante.

ID	Tiempo (s)	Tray. mano (m)	Tray. cabeza (m)	Cab./mano (%)	$\bar{v}$ mano (m/s)	Tray./acierto (m)
P1	133,43	41,29	6,70	16,2	0,294	1,38
P2	117,77	32,86	5,36	16,3	0,260	1,64
T1	180,52	49,10	4,41	9,0	0,272	0,98
T2	102,65	32,90	5,27	16,0	0,322	1,10
U1	64,75	15,48	1,88	12,1	0,222	0,77
U2	45,28	27,11	5,28	19,5	0,545	1,36

Tray. = trayectoria acumulada; $\bar{v}$ = velocidad media de la mano derecha; Tray./acierto = trayectoria media de mano por acierto.

aislada, ya que los pacientes permanecieron más tiempo en la tarea. Por ello, la trayectoria total se analiza junto con el tiempo efectivo y la velocidad media.

La relación cabeza/mano se utilizó como indicador indirecto de compensación. P1 y P2 se situaron en torno al 16 %, T1 obtuvo el valor más bajo y U2 el más elevado. En los pacientes no se observó un incremento progresivo de esta relación entre rondas, por lo que los datos no sugieren una compensación acumulativa mediante cabeza o tronco superior durante la sesión. No obstante, para confirmar compensaciones clínicas sería necesario registrar también tronco, hombro y otros segmentos corporales.

6.2.4. Calidad técnica del registro

Tabla 6.9: Indicadores de calidad del seguimiento por participante.

ID	Frec. fusión (Hz)	Frec. visión (Hz)	Unknown (%)	Conf. media
P1	31,1	8,0	6,1	0,69
P2	30,9	7,3	2,5	0,70
T1	31,0	8,5	2,4	0,71
T2	30,9	6,0	6,3	0,69
U1	30,8	8,4	3,1	0,71
U2	30,9	8,9	6,7	0,69

Conf. media = score medio de confianza YOLOv8-Pose, no precisión funcional.

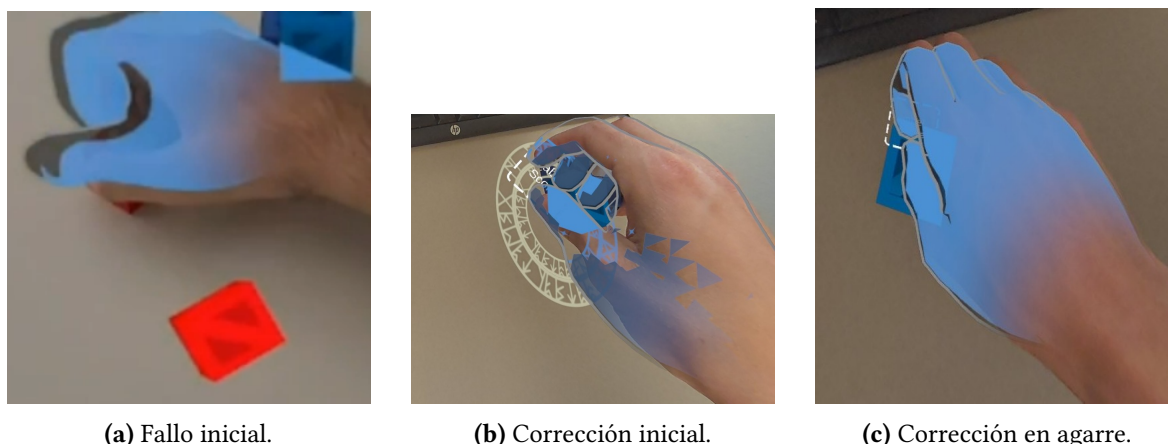
La frecuencia de fusión permaneció estable en torno a 31 Hz y la frecuencia visual osciló entre 6,0 y 8,9 Hz. Las detecciones Unknown se mantuvieron por debajo del 7 % y las detecciones válidas conservaron los cuatro *keypoints*. La confianza media de 0,69–0,71 no debe interpretarse como una precisión clínica del 70 %, sino como el score interno del detector para detecciones aceptadas; la precisión funcional se calculó de forma independiente a partir de los eventos del juego. Aunque no se etiquetaron manualmente todos los episodios de oclusión en sesiones reales, la continuidad de fusión a 31 Hz, la baja proporción de Unknown y la validación controlada de la Sección 6.1.4 indican que las pérdidas visuales temporales no interrumpieron el seguimiento del cubo.

La respuesta ante oclusiones se apoya, por tanto, en dos niveles de evidencia. El primero es el experimento controlado, donde se compara explícitamente la congelación de la pose visual frente a la continuidad de la fusión. El segundo procede de las sesiones reales: aunque no se dispone de una anotación manual de cada oclusión, el sistema registró antigüedad de la detección, estado de agarre, detecciones desconocidas y continuidad de la pose fusionada. La ausencia de interrupciones en la frecuencia de fusión durante las rondas válidas sugiere que el sistema siguió proporcionando una pose utilizable incluso cuando la mano reducía temporalmente la visibilidad del cubo.

La señal FSR se interpretó como medida complementaria de presión/contacto, no como condición obligatoria de agarre, ya que el usuario puede manipular el cubo sin presionar directamente la zona sensorizada. Por esta razón, el fichero de segmentos de interacción se consideró una fuente secundaria: en algunos registros capturaba eventos muy breves o transiciones de estado, pero no ciclos completos de coger, transportar y soltar. Para el análisis principal se priorizaron los eventos del juego, la cinemática del usuario, la calidad visual, la fusión y la calibración.

6.2.5. Incidencia técnica durante el agarre por tenodesis

Durante las pruebas con pacientes se detectó una incidencia relevante en un participante con dificultades severas de agarre: la pinza por tenodesis provocaba un error apreciable en la pose del cubo, especialmente en profundidad. El problema se producía en un momento crítico de la tarea, el agarre y transporte del objeto físico, y afectaba a la correspondencia perceptual entre el cubo real y el cubo virtual. La incidencia fue revisada con el equipo del hospital y permitió ajustar el sistema para mejorar la correspondencia entre ambos.



(a) Fallo inicial.

(b) Corrección inicial.

(c) Corrección en agarre.

Figura 6.3: Incidencia durante el agarre por tenodesis y resultado tras la corrección.

Este caso refuerza la necesidad de evaluar con pacientes reales, ya que algunos patrones de agarre no aparecen en pruebas controladas ni en usuarios sin afectación motora. Además, conecta la precisión percibida del alineamiento físico-virtual con la robustez ante oclusiones generadas por la propia mano.

6.2.6. Percepción subjetiva

Tras la sesión, los participantes cumplieron un cuestionario sobre usabilidad, cibermareo, interacción con cubos, motivación y valor terapéutico percibido. Las respuestas utilizaron escalas Likert de 5 puntos, salvo la escala SUS, normalizada al rango 0–100. El formulario completo se incluye en el Anexo A.3.

La usabilidad se evaluó mediante la escala System Usability Scale (SUS) [6]. La puntuación media fue de 87,0 puntos, con una mediana de 90,3, lo que sitúa la usabilidad percibida en una banda excelente según la interpretación habitual de la escala [5]. Cuatro de los seis participantes superaron el umbral de 80,3 asociado al percentil 90 de la distribución normativa [34].

Tabla 6.10: Resumen de la percepción subjetiva de los participantes.

Dimensión	Resultado principal
Usabilidad SUS	Media de 87,0 y mediana de 90,3 sobre 100.
Cibermareo	Media global de 1,03 sobre 5; cinco participantes sin síntomas.
Interacción con cubos	Media global de 4,70 sobre 5; agarre y <i>feedback</i> obtuvieron 5,00.
Precisión virtual/física	Aspecto peor valorado dentro de la interacción, con media de 4,17 sobre 5.
Comodidad y motivación	Comodidad del visor de 4,89 sobre 5 y valoración motivacional positiva.
Valor terapéutico	Efectividad percibida de 4,83 sobre 5; disposición a seguir usando realidad mixta de 5,00.

Dimensión	Valor favorable	Representación
Usabilidad SUS	87,0 %	
Tolerancia al cibermareo	99,3 %	
Interacción con cubos	94,0 %	
Precisión virtual/física	83,4 %	
Comodidad del visor	97,8 %	
Valor terapéutico percibido	96,6 %	
Disposición/uso complementario	100,0 %	

Figura 6.4: Resumen gráfico de la percepción subjetiva. Las dimensiones Likert se normalizan a 0–100; en cibermareo se representa tolerancia, no intensidad de síntomas.

La percepción fue globalmente positiva: SUS medio de 87,0, ausencia prácticamente total de cibermareo y alta valoración del *feedback*, la motivación y el posible uso como complemento de la rehabilitación convencional. En relación con el cibermareo, cinco participantes no reportaron síntomas y solo U2 indicó síntomas visuales leves. Este resultado sugiere que el uso de *passthrough* en realidad mixta y la corta duración de las sesiones redujeron el conflicto vestibulo-visual. El punto más débil fue la precisión percibida entre cubo físico y virtual, coherente con la incidencia de profundidad descrita y con la necesidad de seguir refinando la calibración.

La interacción tangible con cubos físicos también fue bien valorada. La facilidad de agarre y la utilidad del *feedback* visual y sonoro obtuvieron las puntuaciones más altas, lo que indica que los participantes comprendieron la tarea y pudieron relacionar la acción física con la respuesta del entorno virtual. La precisión virtual/física fue el aspecto relativamente peor valorado, aunque permaneció en un rango positivo. Esta observación es útil para priorizar mejoras: el sistema ya resulta usable y motivador, pero la alineación espacial debe seguir refinándose, especialmente en profundidad y durante agarres atípicos.

La valoración terapéutica debe interpretarse como percepción inicial, no como evidencia clínica de eficacia. Aun así, que los participantes valoren positivamente el sistema como complemento a la rehabilitación convencional es relevante para justificar futuras pruebas, ya que la aceptación por parte de pacientes y profesionales condiciona la viabilidad de integrar este tipo de herramientas en sesiones reales.

6.2.7. Discusión y limitaciones

La evaluación confirma la viabilidad preliminar del sistema en un entorno hospitalario: los pacientes completaron sus rondas, el registro produjo métricas analizables y la interacción tangible fue bien aceptada. Esta observación es relevante porque la validación técnica por sí sola no garantiza que el sistema sea usable en condiciones clínicas reales.

Las diferencias entre perfiles aparecen reflejadas en el ritmo, la trayectoria por acierto y los errores de color, aunque el tamaño muestral impide cualquier generalización clínica. Los perfiles profesionales mostraron ritmos superiores y ausencia de *timeouts*; las personas sin patología motora alcanzaron los ritmos más altos; y los pacientes necesitaron más tiempo y mayor trayectoria por acierto. Esta gradación sugiere que las métricas registradas son sensibles a diferencias funcionales en la ejecución de la tarea.

La calibración previa tiene un papel central en la seguridad del ejercicio. No solo alinea objetos físicos y virtuales, sino que sitúa los objetivos dentro del rango de alcance funcional del usuario, evitando posiciones inaccesibles, sobreesfuerzos y compensaciones con segmentos corporales no afectados. Desde el punto de vista clínico, este ajuste reduce el riesgo de que el paciente utilice partes sanas del cuerpo para compensar la falta de alcance y facilita que el movimiento se mantenga dentro del objetivo terapéutico. Por tanto, debe interpretarse como un mecanismo de personalización terapéutica, no solo como un ajuste técnico.

Las principales limitaciones son el número reducido de pacientes, la existencia de rondas excluidas por interrupciones o registros parciales, el uso de la relación cabeza/mano como indicador indirecto de compensación y la administración de una versión SUS de nueve ítems. En estudios posteriores será necesario ampliar la muestra, etiquetar explícitamente los episodios de oclusión, registrar tronco y hombro, y analizar con mayor detalle eficiencia motora, fatiga y estrategias compensatorias. En conjunto, los resultados proporcionan una base razonable para una segunda fase experimental con mayor número de pacientes y un protocolo más controlado.

6.2.8. Divulgación del trabajo

Una vez desarrollada la propuesta, parte del trabajo fue presentado en el Congreso XRSalento2026 celebrado en Italia. Esta presentación permitió dar a conocer el proyecto en un contexto académico y profesional, así como recoger impresiones externas sobre su planteamiento y resultados.



(a) Portada de la presentación.



(b) Esquema de la arquitectura propuesta.

Figura 6.5: Presentación del artículo en XRSalento2026.

CAPÍTULO 7

Conclusiones

Este capítulo sintetiza las aportaciones del trabajo, evalúa el grado de consecución de los objetivos planteados, justifica las competencias aplicadas y propone las principales líneas de trabajo futuro.

7.1. CONCLUSIONES GENERALES

El presente Trabajo Fin de Grado ha abordado el diseño, desarrollo e integración de un sistema de objetos sensorizados para la rehabilitación del miembro superior en entornos de realidad mixta. La solución combina cubos terapéuticos instrumentados con sensores inerciales y de presión, visión por computador basada en YOLOv8-Pose, un algoritmo de fusión visuo-inercial y un juego serio ejecutado sobre el visor Meta Quest 3.

La principal contribución técnica es la estrategia de fusión multimodal, que integra señales visuales, inerciales y táctiles para mantener la correspondencia entre el objeto físico y su gemelo digital durante la interacción, incluso en periodos de oclusión total. La validación experimental confirma que el sistema mejora la continuidad del seguimiento, captura la rotación intrínseca del objeto y aborda las principales limitaciones identificadas en el estado del arte: oclusión visual, detección fiable del agarre e interacción tangible. Frente a un *ground truth*, mientras el seguimiento exclusivamente visual congela la pose bajo oclusión total, la fusión conserva una dispersión espacial comparable a la referencia, y en la rotación del objeto en la mano captura 314,1° del giro real frente a los apenas 50° que registra la visión por sí sola, un aspecto especialmente valioso en rehabilitación porque permite registrar movimientos finos de los dedos sin rotación equivalente de la muñeca.

Más allá de la validación técnica, el sistema fue llevado al entorno clínico y probado con seis participantes en el Hospital Nacional de Paraplégicos de Toledo, incluyendo pacientes con lesión medular, una terapeuta ocupacional y una profesional de ingeniería biomédica. Los participantes completaron la totalidad de los objetivos planteados 170 de 170 y valoraron la herramienta con una usabilidad media (SUS) de 87,0 sobre 100, situada en la banda excelente de la escala, una efectividad terapéutica percibida de 4,83 sobre 5 y una disposición unánime a seguir utilizando la realidad mixta como complemento de la terapia convencional, todo ello con ausencia prácticamente total de cibermareo. Las pruebas con pacientes reales aportaron además un valor difícil de obtener en laboratorio: la detección de patrones de agarre atípicos, como la pinza por tenodesis, que motivaron ajustes concretos del sistema validados junto al equipo del hospital. Los resultados muestran también una latencia BLE media de 34,15 ms, compatible con la manipulación en tiempo real, y una corrección de *yaw* mediante *keypoints* con un error mediano de 9,88° equivalente al de los marcadores QR (8,67°) que permite prescindir tanto del magnetómetro como de los marcadores físicos.

Desde el punto de vista clínico, el sistema restituye parte de la validez ecológica de la terapia al permitir trabajar con objetos físicos reales dentro de un entorno digital motivador, a la vez que proporciona métricas cinemáticas y de agarre registradas de forma automática. Su bajo coste, autonomía energética e integración embebida favorecen una posible implantación progresiva en entornos hospitalarios y, potencialmente, domiciliarios.

7.2. OBJETIVOS ALCANZADOS

Los cinco objetivos específicos planteados se han cumplido en su totalidad, y los resultados experimentales del Capítulo 6 respaldan de forma cuantitativa esta afirmación. El sistema operó con una latencia BLE media de 34,15 ms (mediana de 32,60 ms sobre $N = 1\,121$ eventos), por debajo del umbral de 50 ms considerado aceptable para la manipulación tangible; la fusión visuo-inercial se mantuvo estable a 31 Hz durante todas las sesiones, con menos de un 7 % de detecciones desconocidas; y la corrección de *yaw* por *keypoints* alcanzó una mediana de error de $9,88^\circ$, prácticamente equivalente a la de los marcadores QR ($8,67^\circ$) pese a prescindir de ellos. En las pruebas con seis participantes del Hospital Nacional de Paraplégicos de Toledo, los usuarios completaron 170 de 170 objetivos, con una usabilidad media (SUS) de 87,0 sobre 100 banda excelente y una valoración terapéutica percibida de 4,83 sobre 5. Estas cifras evidencian que el sistema no solo se implementó, sino que funcionó de forma robusta con pacientes reales. A continuación se detalla el grado de consecución de cada objetivo:

- **OBJ1. Análisis y selección de tecnologías.** Cumplido. Se compararon plataformas hardware, sensores y protocolos de comunicación, seleccionando el Seeed Studio XIAO nRF52840 Sense por su integración de IMU, BLE y gestión de carga en un formato reducido. También se evaluó y descartó el magnetómetro por interferencias electromagnéticas.
- **OBJ2. Diseño e implementación del objeto instrumentado.** Cumplido. Se fabricaron tres cubos sensorizados mediante impresión 3D, con electrónica embebida, sensor FSR y batería LiPo. El firmware bajo Zephyr OS incorpora calibración, filtrado de orientación y transmisión BLE con bajo consumo, negociando un intervalo de conexión de 7,5 ms durante la manipulación activa.
- **OBJ3. Capa de comunicación e integración en RM.** Cumplido. Se desarrolló el componente BLEManager, encargado de gestionar la conexión, recepción, conversión de datos y asociación de múltiples dispositivos mediante dirección MAC.
- **OBJ4. Algoritmo de seguimiento resistente a la oclusión.** Cumplido. El módulo de fusión combina posición visual, orientación inercial y validación táctil del agarre, incorporando calibración de *yaw* por *keypoints*, evaluación multimodal de contacto y mecanismos de control frente a oclusiones y falsas detecciones. En el experimento controlado frente a un *ground truth*, bajo oclusión total la fusión conservó una dispersión espacial comparable a la referencia, frente a la congelación de pose del seguimiento solo visual, y en rotación en mano capturó $314,1^\circ$ del giro real frente a los 50° registrados por la visión exclusivamente.
- **OBJ5. Aplicación de rehabilitación y validación técnica.** Cumplido. Se desarrolló un juego serio con calibración del espacio de trabajo, dificultad adaptativa, retroalimentación multimodal y registro estructurado de métricas cinemáticas e interacción en archivos CSV.

7.3. JUSTIFICACIÓN DE COMPETENCIAS ADQUIRIDAS

El desarrollo del trabajo ha permitido aplicar de forma directa varias de las competencias asociadas a la intensificación en Ingeniería de Computadores:

- **IC2:** Desarrollo de procesadores específicos y sistemas empuotrados, así como la optimización de su software. Se ha ejercido al diseñar e implementar los cubos sensorizados sobre el microcontrolador Seeed Studio XIAO nRF52840 Sense, programando bajo Zephyr RTOS un firmware que integra adquisición de la IMU y del sensor FSR, filtrado de orientación, gestión de energía y transmisión BLE de baja latencia.
- **IC5:** Análisis, evaluación y selección de las plataformas hardware y software más adecuadas para aplicaciones empuotradas y de tiempo real. Se evidencia en el proceso de comparación y elección razonada de la plataforma, los sensores y el sistema operativo de tiempo real, así como en la decisión justificada de descartar el magnetómetro por las interferencias electromagnéticas detectadas.
- **IC1:** Diseño y construcción de sistemas digitales y de comunicaciones. Se cumple mediante la electrónica embebida de los objetos instrumentados y la capa de comunicación inalámbrica que sincroniza múltiples dispositivos con el visor a través de su dirección MAC.
- **IC4:** Diseño e implementación de software de sistema y de comunicaciones. Materializada en el componente BLEManager y en la gestión de la pila BLE encargada de la conexión, la recepción y la conversión de los datos transmitidos por los cubos.
- **IC7:** Análisis, evaluación, selección y configuración de plataformas hardware para la ejecución de aplicaciones informáticas. Aplicada al desplegar y ejecutar sobre el visor Meta Quest 3 tanto el modelo de visión como la lógica de fusión e interacción en tiempo real.

Junto a las competencias de intensificación, el trabajo ha requerido la aplicación de varias competencias comunes a la rama de informática:

- **CO14:** Programación concurrente y de tiempo real. Está presente tanto en la arquitectura del firmware como en la lógica de fusión, que procesa en paralelo señales visuales, inerciales y táctiles manteniendo una frecuencia estable de 31 Hz.
- **CO15:** Principios y técnicas de los sistemas inteligentes y su aplicación práctica. Se ejerce mediante el despliegue del modelo de visión por computador YOLOv8-Pose para la detección de los cubos y de sus *keypoints*.
- **CO1:** Diseño, desarrollo y evaluación de aplicaciones y sistemas informáticos asegurando su fiabilidad y calidad. Se refleja en el proceso completo de construcción del sistema y en su validación técnica, que verifica de forma cuantitativa la latencia, la continuidad del seguimiento y la robustez frente a oclusiones.
- **CO17:** Diseño y evaluación de interfaces persona-computador que garanticen la accesibilidad y la usabilidad. Se evidencia en el diseño del juego serio y en su evaluación con usuarios mediante la escala SUS, que arrojó una puntuación media de 87,0 sobre 100.

7.4. LIMITACIONES DEL SISTEMA

A pesar de los resultados obtenidos, el sistema presenta varias limitaciones que conviene reconocer. La primera tiene que ver con la identificación de los objetos, basada actualmente

en el color del cubo: este enfoque restringe el número de elementos que pueden manipularse de forma simultánea y no escala directamente a una prueba Box and Block Test estándar, en la que coexisten numerosos bloques de colores repetidos. A ella se añade la deriva de *yaw*, ya que cuando los *keypoints* de la cara superior no permanecen visibles durante periodos prolongados el *drift* del giroscopio se acumula hasta que la cámara vuelve a detectarlos y aplica una corrección. No obstante, la limitación más relevante es la ausencia de una posición absoluta independiente de la visión; aunque se ha mitigado parcialmente mediante el puente inercial de oclusión, que conserva la continuidad espacial del gemelo digital durante las pérdidas visuales, no llega a resolverse por completo al carecer el sistema de una referencia posicional propia.

7.5. TRABAJO FUTURO

Los resultados obtenidos permiten plantear varias líneas de continuación que abordan tanto las limitaciones identificadas como nuevas oportunidades de aplicación. La más prioritaria, directamente ligada a la principal limitación del sistema, consiste en incorporar una IMU de nueve ejes que proporcione una estimación de posición absoluta; integrarla de forma efectiva y precisa simplificaría notablemente la fusión visuo-inercial y mejoraría la exactitud posicional del gemelo digital. En paralelo, sería deseable ampliar el catálogo de objetos instrumentados más allá de los cubos, incorporando elementos representativos de actividades de la vida diaria, como vasos, cubiertos o cepillos, mediante la arquitectura ShapePolicy ya prevista en el diseño.

Desde el punto de vista clínico, las métricas de agarre y movimiento ya registradas abren la puerta a una adaptación dinámica de la dificultad, capaz de ajustar automáticamente parámetros del juego como el tiempo disponible, la distancia entre plataformas o el nivel de asistencia. Estas mismas métricas podrían explotarse en un *dashboard* de análisis cinemático que permitiera al equipo terapéutico visualizar trayectorias, perfiles de fuerza, segmentos de interacción y la evolución longitudinal del paciente. Finalmente, todo ello sienta las bases para extender el sistema hacia la telerrehabilitación domiciliaria, sincronizando los datos de cada sesión con el equipo clínico mediante servicios en la nube y acercando así la terapia al entorno cotidiano del paciente.

7.6. USO DE LA INTELIGENCIA ARTIFICIAL

En línea con las recomendaciones actuales sobre transparencia en el uso de inteligencia artificial en ámbitos académicos, se detalla a continuación cómo se han empleado estas herramientas en el presente Trabajo Fin de Grado.

En la redacción de la memoria, la IA sirvió de apoyo para la revisión ortográfica y gramatical, reformulando puntualmente fragmentos cuando la redacción original no resultaba clara. En ningún caso intervino en la generación de contenido nuevo, en la estructura de los capítulos ni en las conclusiones, siendo la idea, el diseño y el texto obra del autor.

En el desarrollo del código, el uso fue puntual e iterativo, siempre posterior a un diseño propio definido mediante diagramas, requisitos y decisiones de arquitectura. La IA se empleó únicamente como apoyo en la resolución de errores concretos, sin intervenir en la concepción de la idea, el diseño de algoritmos ni la arquitectura, responsabilidad exclusiva del autor.

Bibliografía

- [1] Kaushik Ahuja, S. Paredy, Robert Xiao, Mayank Goel, and Chris Harrison. LightAnchors: Appropriating point lights for spatially-anchored augmented reality interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology (UIST '19)*, pages 189–196. ACM, 2019.
- [2] Robert S. Allison, Laurence R. Harris, Michael Jenkin, Urszula Jasiobedzka, and John E. Zacher. Tolerance of temporal delay in virtual environments. In *Proceedings of the 2001 IEEE Virtual Reality Conference (VR'01)*, pages 247–254, Yokohama, Japan, 2001. IEEE.
- [3] M. Alvarez-Rodríguez, E. López-Dolado, M. Salas-Monedero, V. Lozano-Berrio, S. Ceruelo-Abajo, A. Gil-Agudo, and A. de los Reyes-Guzmán. Concurrent validity of a virtual version of box and block test for patients with neurological disorders. *World Journal of Neuroscience*, 10(1):79–89, February 2020.
- [4] Walter Baccinelli, Franco Molteni, and Maria Bulgheroni. Smart objects in rehabilitation. In Lorenzo Masia, Silvestro Micera, Metin Akay, and José L. Pons, editors, *Converging Clinical and Engineering Research on Neurorehabilitation III: Proceedings of the 4th International Conference on NeuroRehabilitation (ICNR2018), October 16–20, 2018, Pisa, Italy*, volume 21 of *Biosystems & Biorobotics*, pages 12–15. Springer, Cham, 2019.
- [5] Aaron Bangor, Philip T. Kortum, and James T. Miller. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, 2008.
- [6] John Brooke. Sus: A quick and dirty usability scale. In Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland, editors, *Usability Evaluation in Industry*, pages 189–194. Taylor & Francis, London, 1996.
- [7] Lynn Connelly, Yang Jia, Maria L. Toro, Mary Ellen Stoykov, Robert V. Kenyon, and Derek G. Kamper. A pneumatic glove and immersive virtual reality environment for hand rehabilitative training after stroke. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 18(5):551–559, 2010.
- [8] Ana de Los Reyes-Guzmán et al. RehabHand: Oriented-tasks serious games for upper limb rehabilitation by using Leap Motion Controller and target population in spinal cord injury. *NeuroRehabilitation*, 48(3):365–373, 2021.
- [9] Iris Dimbwadyo-Terrer, Francisco Trincado-Alonso, Ana de los Reyes-Guzmán, Miguel A. Aznar, Carmen Alcubilla, Sara Pérez-Nombela, and Ángel Gil-Agudo. Upper limb rehabilitation after spinal cord injury: A treatment based on a data glove and an immersive virtual reality environment. *Disability and Rehabilitation: Assistive Technology*, 11(6):462–467, 2016.

- [10] Iris Dimbwadyo-Terrer, Francisco Trincado-Alonso, Ana de los Reyes-Guzmán, Antonio Bernal-Sahún, Pilar López-Monteagudo, Begoña Polonio-López, and Ángel Gil-Agudo. Effectiveness of the virtual reality system toyra on upper limb function in people with tetraplegia: A randomized clinical trial. *BioMed Research International*, 2016:6397828, 2016.
- [11] Filippo D’Ippolito and Marco Massaro. A multi-sensor multi-rate algorithm for motor rehabilitation with Augmented Reality devices. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, pages 759–765, Singapore, Singapore, June 2016. IEEE.
- [12] Ángel Gil-Agudo, Iris Dimbwadyo-Terrer, Begoña Peñasco-Martín, Ana de los Reyes-Guzmán, Antonio Bernal-Sahún, and Antonio Berbel-García. Experiencia clínica de la aplicación del sistema de realidad TOyRA en la neuro-rehabilitación de pacientes con lesión medular. *Rehabilitación*, 46(1):24–34, 2012.
- [13] Vanesa Herrera, Javier Albusac, José Jesús Castro-Schez, Carlos González-Morcillo, Dorothy N. Monekosso, Santiago Pacheco, Rubén Perales, and Ana de los Reyes-Guzmán. Creating adapted environments: Enhancing accessibility in virtual reality for upper limb rehabilitation through automated element adjustment. *Virtual Reality*, 29:28, 2025.
- [14] Vanesa Herrera, Javier Alonso Albusac, Eusebio Angulo, Carlos González-Morcillo, Ana de los Reyes-Guzmán, David Vallejo, and José Jesús Castro-Sánchez. Virtual reality-assisted goalkeeper training for upper limb rehabilitation in a safe and adapted patient environment. *IEEE access : practical innovations, open solutions*, 12:194256–194279, 2024.
- [15] Vanesa Herrera, David Vallejo, José J. Castro-Schez, Dorothy N. Monekosso, Ana de los Reyes-Guzmán, Carlos Glez-Morcillo, and Javier Albusac. Rehab-Immersive: A framework to support the development of virtual reality applications in upper limb rehabilitation. *SoftwareX*, 23:101412, 2023.
- [16] C. L. Hunt, A. Sharma, L. E. Osborn, R. R. Kaliki, and N. V. Thakor. Predictive trajectory estimation during rehabilitative tasks in augmented reality using inertial sensors. In *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2018.
- [17] Jason J. Jerald. *Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays*. PhD thesis, University of North Carolina at Chapel Hill, 2009.
- [18] Reneh Karamians, Rachel Proffitt, David Kline, and Lynne V. Gauthier. Effectiveness of virtual reality- and gaming-based interventions for upper extremity rehabilitation poststroke: A meta-analysis. *Archives of Physical Medicine and Rehabilitation*, 2020.
- [19] Peter Langhorne, Julie Bernhardt, and Gert Kwakkel. Stroke rehabilitation. *The Lancet*, 377(9778):1693–1702, 2011.
- [20] Kate E. Laver, Belinda Lange, Stacey George, Judith E. Deutsch, Gustavo Saposnik, and Maria Crotty. Virtual reality for stroke rehabilitation. *Cochrane Database of Systematic Reviews*, (11):CD008349, 2017.
- [21] Brian J. B. Lee, A. Williams, and Pinhas Ben-Tzvi. Intelligent object grasping with sensor fusion for rehabilitation and assistive applications. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(8):1556–1565, 2018.
- [22] Han Suk Lee, Yoo Junk Park, and Sun Wook Park. The effects of virtual reality training on function in chronic stroke patients: A systematic review and meta-analysis. *BioMed Research International*, 2019, 2019.

- [23] T. K. M. Lee, J. G. Lim, K. H. Leo, and S. Sanei. Indications of neural disorder through automated assessment of the box and block test. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*. IEEE, 2018.
- [24] S. C. Leong, Y. M. Tang, F. M. Toh, and K. N. K. Fong. Examining the effectiveness of virtual, augmented, and mixed reality (VAMR) therapy for upper limb recovery and activities of daily living in stroke patients: A systematic review and meta-analysis. *Journal of NeuroEngineering and Rehabilitation*, 19(93), 2022.
- [25] Guan Ming Lim, Prayook Jatesiktat, Christopher Wee Keong Kuah, and Wei Tech Ang. Wireless pressure sensor array module for sensorized object. In *45th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 1–5, 2023.
- [26] Sebastian O. H. Madgwick, Andrew J. L. Harrison, and Ravi Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7. IEEE, 2011.
- [27] Katerina Mania, Bernard D. Adelstein, Stephen R. Ellis, and Michael I. Hill. Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity. In *Proceedings of the 1st Symposium on Applied Perception in Graphics and Visualization (APGV '04)*, pages 39–47, Los Angeles, CA, USA, 2004. ACM.
- [28] Laura Marchal-Crespo and David J. Reinkensmeyer. Review of control strategies for robotic movement training after neurological injury. *Journal of NeuroEngineering and Rehabilitation*, 6:20, 2009.
- [29] Virgil Mathiowetz, Gloria Volland, Nancy Kashman, and Karen Weber. Adult norms for the Box and Block Test of manual dexterity. *The American Journal of Occupational Therapy*, 39(6):386–391, 1985.
- [30] J. Monge and O. Postolache. Augmented reality and smart sensors for physical rehabilitation. In *2018 International Conference and Exposition on Electrical and Power Engineering (EPE)*, pages 1010–1014, Iași, Romania, 2018. IEEE.
- [31] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, and Mary Rodgers. A review of wearable sensors and systems with application in rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 9(1):21, 2012.
- [32] A. Pillai, M. S. H. Sunny, M. T. Shahria, N. Banik, and M. H. Rahman. Gamification of upper limb rehabilitation in mixed-reality environment. *Applied Sciences*, 12(23), 2022.
- [33] O. Postolache, D. J. Hemanth, R. Alexandre, D. Gupta, O. Geman, and A. Khanna. Remote monitoring of physical rehabilitation of stroke patients using IoT and virtual reality. *IEEE Journal on Selected Areas in Communications*, 39(2):562–573, 2021.
- [34] Jeff Sauro. *A Practical Guide to the System Usability Scale: Background, Benchmarks and Best Practices*. MeasuringU Press, Denver, CO, 2011.
- [35] Unidad de Biomécanica y Ayudas Técnicas, HNP. Proyecto Naviga: Estudio piloto de rehabilitación del miembro superior mediante guante CyberGlove. Informe técnico interno, Hospital Nacional de Paraplégicos, Toledo, 2012.
- [36] Dilmer Valecillos. Ai building blocks demos, 2024.
- [37] Y. Zhang, Y. Chen, H. Yu, Z. Lv, P. Shang, Y. Ouyang, X. Yang, and W. Lu. Wearable sensors based automatic box and block test system. In *2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing*

& Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/Cbdcom/IOP/SCI), pages 952–959. IEEE, 2019.

ANEXOS

Recursos complementarios del proyecto

Este anexo recoge los recursos complementarios generados durante el desarrollo del proyecto. Estos materiales no son necesarios para la comprensión de la memoria, pero permiten consultar el código fuente, los recursos experimentales y evidencias adicionales asociadas al sistema desarrollado. Todos los recursos se publican de forma abierta para su consulta directa.

A.1. REPOSITORIO DE CÓDIGO FUENTE

El código fuente del proyecto se encuentra alojado en un repositorio público de GitHub, disponible para su consulta directa. El repositorio incluye los componentes desarrollados para la aplicación de realidad mixta, el firmware del objeto sensorizado, los módulos de comunicación BLE, el sistema de fusión visuo-inercial, el juego serio y las utilidades de registro de datos.

- **Tipo de recurso:** repositorio público de código fuente.
- **Plataforma:** GitHub.
- **Contenido:** scripts de Unity, firmware embebido, módulos de visión, fusión, comunicación BLE, registro de datos y recursos auxiliares del proyecto.
- **Acceso:** público.
- **URL:** <https://github.com/rafaechevace/RehabSensors/tree/main>

El repositorio se ofrece en abierto para facilitar la consulta directa del código, si bien corresponde a una línea de investigación en desarrollo y contiene configuraciones específicas del prototipo.

A.2. DATOS EXPERIMENTALES E INFORMES DERIVADOS

Como complemento a la memoria, se ha organizado un repositorio público en GitHub con los registros originales generados durante las sesiones experimentales. Estos registros se almacenan en formato CSV y contienen la información bruta capturada por el sistema, incluyendo eventos del juego, cinemática del usuario, detecciones visuales, estado de los cubos, resultados de fusión, datos de sensores y registros de calibración.

Dado que estos ficheros son extensos y no están pensados para una revisión manual directa, se incluyen además informes derivados por participante. Estos informes sintetizan los principales resultados funcionales, cinemáticos y técnicos de cada sesión, y son la fuente de apoyo recomendada para revisar los resultados presentados en la Sección 6.2.2.

El repositorio incluye, por tanto, dos tipos de material:

- Registros CSV originales, utilizados como evidencia bruta de las sesiones realizadas.
- Informes derivados por participante, empleados como soporte interpretativo para la elaboración de las tablas y discusión de resultados.
- **Tipo de recurso:** repositorio público de datos experimentales.
- **Plataforma:** GitHub.
- **Contenido:** registros CSV originales e informes derivados por participante.
- **Acceso:** público, disponible para consulta directa.
- **URL:** <https://github.com/rafaechevace/RehabSensorsData>

Estos materiales no forman parte del software desarrollado, sino que se aportan como documentación experimental complementaria.

Adicionalmente, se dispone de material audiovisual de las sesiones, correspondiente a la grabación en primera persona de lo que el paciente visualiza a través del visor durante la prueba. Por motivos de privacidad, este material no se publica de forma abierta y debe solicitarse directamente al autor.

A.3. CUESTIONARIO DE PERCEPCIÓN SUBJETIVA

Tras la sesión de uso, los participantes cumplieron un cuestionario breve orientado a valorar la usabilidad, la tolerancia al sistema, la interacción con los cubos físicos y virtuales, la motivación y el valor percibido de la herramienta como apoyo a la rehabilitación. Salvo los ítems SUS, las preguntas se respondieron mediante una escala Likert de 1 a 5, donde 1 indicaba la peor valoración y 5 la mejor valoración. En el bloque de cibermareo, 1 indicaba “ausencia de síntomas” y 5 indicaba “síntomas intensos”.

Datos generales

1. Identificador del participante.
2. Perfil del participante: paciente, profesional del ámbito clínico o biomédico, o persona sin patología motora conocida.
3. Mano utilizada durante la prueba.
4. Experiencia previa con realidad virtual o realidad mixta: ninguna, ocasional (1–5 usos), frecuente (6–20 usos) o muy frecuente (>20 usos).

Usabilidad percibida: escala SUS adaptada

Los ítems de usabilidad se formularon a partir de la escala System Usability Scale (SUS), adaptándolos al sistema de rehabilitación en realidad mixta. La versión administrada incluyó nueve ítems, por lo que la puntuación final se normalizó al rango 0–100.

1. Me gustaría utilizar este sistema con frecuencia.
2. El sistema me pareció innecesariamente complejo.
3. El sistema me pareció fácil de utilizar.
4. Creo que necesitaría ayuda de una persona experta para poder usar el sistema.
5. Las funciones del sistema me parecieron bien integradas.
6. El sistema me pareció inconsistente o difícil de predecir.
7. Creo que la mayoría de las personas aprenderían a utilizar el sistema rápidamente.

8. El sistema me pareció incómodo o engorroso de utilizar.
9. Me sentí seguro/a utilizando el sistema.

Cibermareo y tolerancia

Indique la intensidad de los siguientes síntomas durante o después de la prueba, usando una escala de 1 a 5, donde 1 significa ausencia de síntomas y 5 síntomas intensos.

1. Malestar general.
2. Fatiga.
3. Molestias o síntomas visuales.
4. Dolor de cabeza.
5. Mareo.
6. Náuseas.

Interacción con los cubos físicos y virtuales

1. El cubo físico fue fácil de agarrar.
2. El cubo físico fue fácil de colocar en la posición objetivo.
3. La correspondencia entre el cubo físico y su representación virtual fue precisa.
4. El feedback visual ayudó a entender la tarea.
5. El feedback sonoro ayudó a entender si la acción era correcta.
6. La interacción con los cubos resultó natural.

Motivación, comodidad y valor terapéutico percibido

1. El ejercicio me resultó motivador.
2. La duración de la prueba me pareció adecuada.
3. El visor resultó cómodo durante la sesión.
4. Considero que el sistema podría ayudar en ejercicios de rehabilitación del miembro superior.
5. Considero que el sistema podría complementar la rehabilitación convencional.
6. Estaría dispuesto/a a volver a utilizar un sistema de realidad mixta de este tipo.

Observaciones abiertas

1. Indique cualquier dificultad encontrada durante la prueba.
2. Indique qué aspectos del sistema le han resultado más útiles.
3. Indique posibles mejoras para futuras versiones del sistema.