Universidad Internacional de la Rioja (UNIR)

Escuela Superior de Ingeniería y Tecnología

Máster en Inteligencia Artificial

Clasificación de movimientos de un perro robótico mediante mmWave sensing y aprendizaje automático

Trabajo Fin de Estudios

presentado por: Alberto Martínez Méndez

Dirigido por: Alejandro Cervantes Rovira

Ciudad: Madrid

Fecha: 17 de Julio de 2026

Agradecimientos

Quisiera expresar mi agradecimiento a la Universidad Carlos III de Madrid por haberme permitido desarrollar la parte experimental de este Trabajo Fin de Máster en el marco de su entorno investigador. La disponibilidad de sus instalaciones, recursos y equipamiento ha sido fundamental para llevar a cabo los experimentos descritos en esta memoria. En particular, agradezco la posibilidad de trabajar con el robot cuadrúpedo y con la infraestructura necesaria para la adquisición de medidas, sin las cuales este trabajo no habría podido alcanzar su dimensión experimental.

También quiero agradecer sinceramente a mis compañeros del grupo de investigación por la ayuda prestada durante el desarrollo del trabajo. Sus explicaciones, orientación y disponibilidad me han permitido comprender conceptos técnicos complejos, resolver dificultades prácticas y aprender mucho más allá de lo estrictamente necesario para completar esta memoria. Este trabajo recoge resultados concretos, pero detrás de ellos hay muchas conversaciones, pruebas, errores compartidos y aprendizajes que han sido igual de importantes.

A mi familia, gracias por el apoyo constante desde el principio. Por acompañarme desde pequeño, por confiar en mí incluso en los momentos de mayor incertidumbre y por enseñarme, con su ejemplo, el valor del esfuerzo, la constancia y la responsabilidad. Buena parte de este camino académico y personal se sostiene sobre esa base.

Por último, quiero agradecer a mi círculo cercano por estar siempre ahí. A quienes me han escuchado, animado y acompañado durante este proceso, incluso cuando el trabajo parecía avanzar más despacio de lo esperado. Su apoyo ha hecho que este recorrido fuera más llevadero y significativo.

A todos ellos, gracias.

Índice de Contenidos

Agradecimientos I

Resumen VIII

Abstract IX

1. Introducción 1

1.1. Motivación . . . 2

1.2. Planteamiento del problema . . . 3

1.3. Solución propuesta . . . 4

1.4. Contribuciones principales . . . 5

1.5. Estructura del documento . . . 6

2. Objetivos y Metodología 8

2.1. Objetivo principal . . . 8

2.2. Hipótesis de trabajo . . . 9

2.3. Objetivos específicos . . . 9

2.4. Metodología de trabajo . . . 11

2.4.1. Fase 1: estudio previo y formulación del problema . . . 11

2.4.2. Fase 2: configuración del sistema experimental . . . 11

2.4.3. Fase 3: adquisición y construcción del dataset . . . 11

2.4.4. Fase 4: preprocesado y representación de entrada . . . 12

2.4.5. Fase 5: diseño, entrenamiento y evaluación de modelos . . . 12

2.4.6. Fase 6: inferencia online e integración con gemelo digital . . . 12

2.5. Planificación temporal . . . 13

2.6. Herramientas y entorno de desarrollo . . . 13

2.7. Consideraciones éticas y de seguridad . . . 15

2.8. Impacto socioeconómico y científico . . . 15

3. Estado del Arte 17

3.1. Del sensing inalámbrico al paradigma ISAC . . . 17

3.2. CSI como representación del entorno . . . 18

3.3. Sensing con Wi-Fi en bandas sub-7 GHz . . . 19

3.4. Sensing con ondas milimétricas a 60 GHz . . . 19
3.5. Plataformas comerciales para extracción de CSI mmWave . . . 20
3.6. Robots móviles y percepción externa . . . 21
3.7. Aprendizaje automático para señales CSI . . . 21
3.8. Desplazamiento de dominio y generalización entre sesiones . . . 22
3.9. Gemelos digitales en robótica . . . 23
3.10. Síntesis crítica y posicionamiento del trabajo . . . 24

4. Sistema Experimental y Adquisición del Dataset 26
4.1. Visión general del sistema experimental . . . 26
4.2. Hardware empleado . . . 27
4.3. Configuración del enlace mmWave . . . 29
4.4. Formato de las medidas CSI . . . 30
4.5. Protocolo de adquisición del dataset . . . 31
4.6. Organización del dataset . . . 32
4.7. Distribución temporal de las capturas . . . 35
4.8. Modelo matemático de la señal de entrada . . . 35
4.9. Proceso de limpieza y preprocesado . . . 36
4.10. Particularidades y retos de la adquisición . . . 38
4.11. Relación con los modelos de clasificación . . . 39

5. Diseño de los Modelos de Clasificación 41
5.1. Formulación del problema de clasificación supervisada . . . 41
5.2. Representación de entrada . . . 42
5.3. Partición del dataset y protocolos de evaluación . . . 43
5.4. Arquitecturas consideradas . . . 44
5.4.1. SVM con núcleo RBF . . . 44
5.4.2. Perceptrón multicapa . . . 45
5.4.3. CNN simple . . . 45
5.4.4. Modelo basado en atención . . . 45
5.4.5. CsiNet residual . . . 46
5.5. Modelo propuesto: CsiNet . . . 46
5.5.1. Bloques residuales . . . 47
5.5.2. Cabeza principal de clasificación . . . 49

5.5.3. Cabeza auxiliar estático/dinámico 49
5.5.4. Cabeza adversarial de dominio 49
5.5.5. MixStyle 51
5.6. Funciones de pérdida 51
5.7. Aumento de datos 52
5.8. Configuración de entrenamiento 53
5.9. Variantes experimentales 54
5.10. Métricas de evaluación 55
5.11. Pipeline de inferencia online 57
5.11.1. Captura continua de medidas CSI 58
5.11.2. Buffer temporal y ventana deslizante 58
5.11.3. Preprocesado online 58
5.11.4. Inferencia mediante CsiNet 59
5.11.5. Estabilización temporal de predicciones 59
5.11.6. Modelo auxiliar de baja latencia 60
5.11.7. Comunicación mediante JSON compartido 60
5.11.8. Integración con el gemelo digital 61

6. Evaluación experimental y resultados 62
6.1. Introducción 62
6.2. Métricas de evaluación 62
6.3. Protocolos experimentales 63
6.4. Comparativa entre arquitecturas 64
6.5. Análisis de variantes de CsiNet 67
6.6. Evaluación de generalización intersesión 69
6.7. Análisis por tipo de comportamiento 71
6.8. Discusión crítica de los resultados 72
6.9. Validación funcional del pipeline online y gemelo digital 73
6.9.1. Objetivo y alcance de la validación online 74
6.9.2. Medida de latencia del pipeline 74
6.9.3. Estabilización temporal de la salida 76
6.9.4. Salida JSON e intercambio de estado 77
6.9.5. Integración con el gemelo digital 78

6.9.6. Limitaciones de la validación online 80

7. Conclusiones y trabajo futuro **82**

7.1. Conclusiones 82

7.2. Limitaciones del trabajo 85

7.3. Trabajo futuro 86

7.4. Reflexión final 88

Referencias 90

A. Declaración de uso de herramientas de inteligencia artificial **95**

Índice de Ilustraciones

1.1. Arquitectura general del sistema propuesto. Imagen generada por Inteligencia Artificial 5

2.1. Planificación temporal del Trabajo Fin de Máster (TFM). 13

4.1. Arquitectura general del sistema experimental para adquisición de Channel State Information (CSI) en banda ondas milimétricas (mmWave). 27
4.2. Disposición experimental de las antenas MikroTik y el robot Unitree Go1 durante la adquisición del dataset. 29
4.3. Pipeline de construcción del dataset a partir de ficheros `.txt` crudos. 37

5.1. Arquitectura general de la familia CsiNet. La cabeza de dominio y MixStyle se activan únicamente en las variantes con adaptación adversarial de dominio; el modelo principal empleado en inferencia usa la rama multiclase y la cabeza binaria auxiliar. 48
5.2. Flujo general del pipeline de inferencia online. Imagen generada con Inteligencia Artificial 57

6.1. Matrices de confusión de los modelos con mejor rendimiento en partición aleatoria estratificada. DP: `de_pie`; T: `tumbado`; D2: `dos_patas`; B: `baile`. 66
6.2. Matrices de confusión de los principales experimentos de generalización intersesión. DP: `de_pie`; T: `tumbado`; D2: `dos_patas`; B: `baile`. 70
6.3. Escenario del gemelo digital en Isaac Sim con el robot Unitree Go1 y las antenas MikroTik empleadas en la representación del sistema. 79
6.4. Representación de las cuatro clases inferidas en el gemelo digital. 80

Índice de Tablas

2.1. Herramientas y tecnologías empleadas en el desarrollo del TFM. 14

3.1. Comparativa de trabajos relacionados con el presente TFM. 24

4.1. Hardware empleado en el sistema experimental. 28
4.2. Estructura de una línea de medida CSI generada por la plataforma de captura. 30
4.3. Número de ficheros `.txt` disponibles por clase en el dataset crudo. 33
4.4. Distribución de clases en el dataset final empleado para clasificación. 34
4.5. Estadísticas del proceso de limpieza y validación del dataset. 34
4.6. Distribución de muestras válidas por sesión temporal y clase. 35

5.1. Protocolos de partición empleados para entrenamiento, validación y prueba. 44
5.2. Técnicas de aumento de datos empleadas durante el entrenamiento. 53
5.3. Configuraciones principales de entrenamiento empleadas en los experimentos finales. 54
5.4. Diferencias principales entre las dos configuraciones CsiNet más relevantes para el análisis final. 56

6.1. Protocolos experimentales utilizados en la evaluación. 64
6.2. Comparativa de modelos sobre partición aleatoria estratificada. 65
6.3. Resultados de variantes CsiNet sobre partición aleatoria estratificada. . . . 68
6.4. Evaluación intersesión de variantes CsiNet. 69
6.5. Tiempo extremo a extremo del pipeline de inferencia online. 75
6.6. Efecto de la estabilización temporal sobre la salida online. 77
6.7. Correspondencia entre clase inferida y representación en el gemelo digital. . 79

Resumen

Este Trabajo de Fin de Máster se centra en el uso de técnicas de *sensing* inalámbrico en banda *mmWave* para la clasificación de movimientos de un perro robótico a partir de medidas CSI. El objetivo principal es construir un modelo de inteligencia artificial capaz de identificar distintos movimientos predefinidos a partir de secuencias temporales obtenidas del sistema de medida. Para ello, se trabajará con un conjunto de datos compuesto por múltiples archivos de texto, donde cada muestra contiene 120 medidas CSI asociadas a una determinada acción del robot. Como primera aproximación, se plantea estudiar arquitecturas orientadas al procesamiento de series temporales, prestando especial atención a modelos basados en mecanismos de atención, sin descartar la comparación con otras alternativas. El trabajo se centrará en el diseño, entrenamiento y evaluación del clasificador, dejando como posible línea futura su integración en un sistema en tiempo real y su conexión con un gemelo digital del robot.

Palabras clave: *sensing* inalámbrico, *mmWave*, CSI, perro robótico, aprendizaje automático.

Abstract

This Master's Thesis focuses on the use of wireless sensing techniques in the millimeter-wave band for the classification of robotic dog movements from CSI measurements. The main objective is to build an artificial intelligence model capable of identifying different predefined movements from temporal sequences obtained through the measurement system. To this end, the work will use a dataset composed of multiple text files, where each sample contains 120 CSI measurements associated with a specific robot action. As an initial approach, architectures oriented towards time-series processing will be studied, with particular attention to models based on attention mechanisms, while also considering comparisons with alternative approaches. The work will focus on the design, training and evaluation of the classifier, leaving its integration into a real-time system and its connection with a digital twin of the robot as possible future work.

Keywords: wireless sensing, mmWave, CSI, robotic dog, machine learning.

Alberto Martínez Méndez
Máster en Inteligencia Artificial

1. Introducción

La robótica moderna depende cada vez más de la capacidad de percibir, interpretar y representar el entorno físico en tiempo real. Tradicionalmente, esta percepción se ha apoyado en sensores embarcados en el propio robot, como cámaras, Light Detection and Ranging (LiDAR), unidades inerciales u odometría. Estos sensores son esenciales para tareas de navegación, localización, control y toma de decisiones, y han permitido avances muy significativos en robótica móvil y sistemas autónomos (Thrun, Burgard, y Fox, 2005; Cadena y cols., 2016).

Sin embargo, este paradigma también presenta limitaciones. La información generada por el robot depende de la disponibilidad de sus sensores, de la calidad de sus comunicaciones y de la capacidad del sistema para transmitir su estado hacia módulos externos. En escenarios donde existe alta carga de red, pérdida parcial de conectividad, degradación de sensores, o necesidad de supervisión externa, puede resultar útil disponer de mecanismos adicionales capaces de inferir el estado del robot desde fuera, sin depender exclusivamente de la telemetría interna.

En paralelo, las redes inalámbricas están evolucionando hacia un paradigma en el que las señales de radio no se utilizan únicamente para transmitir información, sino también para extraer conocimiento sobre el entorno físico. Esta idea se recoge en el concepto de Integrated Sensing and Communication (ISAC), que plantea la integración de comunicaciones y sensado sobre una misma infraestructura radio (Liu y cols., 2022). Bajo este enfoque, las ondas electromagnéticas que se propagan por un entorno no solo transportan datos, sino que también se ven afectadas por objetos, personas, superficies y movimientos presentes en la escena. Analizar esas perturbaciones permite utilizar el canal inalámbrico como una fuente de percepción.

El presente Trabajo Fin de Máster se sitúa precisamente en esa intersección entre comunicaciones inalámbricas, inteligencia artificial y robótica. El objetivo general es estudiar si es posible clasificar comportamientos predefinidos de un perro robótico a partir de las variaciones que dichos comportamientos generan en un enlace inalámbrico mmWave a 60 GHz. En lugar de observar directamente al robot mediante una cámara o utilizar sus sensores internos, el sistema intenta inferir su comportamiento a partir de medidas de CSI capturadas entre dos antenas comerciales.

Esta aproximación plantea una pregunta central: ¿puede una señal inalámbrica, diseñada originalmente para comunicación, contener suficiente información como para reconocer el estado o movimiento de un robot? Responder a esta pregunta requiere construir una cadena experimental completa: configurar el enlace mmWave, capturar medidas reales, transformar los datos crudos en una representación adecuada, entrenar modelos de clasificación, evaluar su rendimiento y analizar su posible integración en un sistema operativo de inferencia.

1.1. Motivación

El *sensing* inalámbrico basado en CSI ha demostrado su utilidad en tareas como detección de presencia, reconocimiento de actividad humana, localización o monitorización de signos vitales empleando señales Wireless Fidelity (Wi-Fi) convencionales (Wei, Chen, y cols., 2025). En estos sistemas, las variaciones del canal inalámbrico se interpretan como una huella indirecta de los cambios producidos en el entorno.

No obstante, gran parte de estos trabajos se han desarrollado en bandas sub-7 GHz, donde la longitud de onda es mayor y, por tanto, la sensibilidad espacial ante movimientos pequeños es menor. Las bandas milimétricas, y en particular la banda de 60 GHz, ofrecen características físicas diferentes: longitudes de onda del orden de milímetros, alta direccionalidad y mayor sensibilidad ante variaciones geométricas sutiles. Estas propiedades hacen que la banda mmWave sea especialmente interesante para estudiar perturbaciones producidas por objetos dinámicos o robots en movimiento (van Berlo, Elkelany, Özçelebi, y Meratnia, 2021).

En el caso de un robot cuadrúpedo, esta sensibilidad puede ser útil para distinguir entre posturas estáticas y comportamientos dinámicos. Un robot como el Unitree Go1 (Unitree Robotics, s.f.-b) puede permanecer de pie, tumbarse, adoptar una postura elevada sobre dos patas o ejecutar secuencias de movimiento como un baile (Unitree Robotics, s.f.-a). Cada uno de estos comportamientos modifica de forma distinta la propagación multitrayecto entre las antenas, generando variaciones en la fase y la magnitud de la CSI. La hipótesis de partida de este trabajo es que dichas variaciones pueden aprenderse mediante modelos de inteligencia artificial.

La motivación del trabajo no es sustituir los sensores embarcados del robot, sino explorar una fuente de percepción complementaria. Un sistema externo basado en wireless

sensing podría aportar redundancia, permitir monitorización no visual, enriquecer sistemas de supervisión o alimentar representaciones virtuales del entorno sin depender directamente de cámaras ni de la telemetría interna del robot.

Además, esta línea resulta especialmente relevante en el contexto de los gemelos digitales. Un gemelo digital robótico no solo debe representar la geometría del entorno o la posición del robot, sino también su estado semántico: qué está haciendo, cómo evoluciona y cómo interactúa con el entorno (Segovia y Garcia-Alfaro, 2022; Odonkor, Coffelt, Syrbe, y Beetz, 2023). Integrar inferencias obtenidas desde señales inalámbricas permite enriquecer estas representaciones virtuales con información procedente de una modalidad de sensado externa y no visual.

1.2. Planteamiento del problema

El problema abordado en este trabajo consiste en clasificar comportamientos predefinidos de un perro robótico a partir de secuencias temporales de medidas CSI obtenidas mediante un enlace mmWave biestático.

El sistema experimental se basa en dos dispositivos MikroTik wAP 60G configurados como enlace transmisor–receptor en la banda de 60 GHz. El robot se sitúa en la zona de mayor sensibilidad del enlace y ejecuta distintos comportamientos. Las antenas capturan medidas de canal que reflejan las perturbaciones inducidas por el robot en la propagación de la señal.

La versión final del sistema se centra en cuatro clases de comportamiento:

- `de_pie`;
- `tumbado`;
- `dos_patas`;
- `baile`.

Durante la campaña de adquisición también se recopilaron muestras de la condición `no_robot`, utilizadas como condición de control para caracterizar el canal en ausencia del robot. Sin embargo, esta clase no se incorpora al clasificador final, ya que el objetivo principal del trabajo es discriminar comportamientos del robot una vez presente en la zona de sensado, no resolver simultáneamente el problema de detección de presencia.

El problema presenta varios retos técnicos. En primer lugar, las variaciones generadas por el robot sobre el canal son indirectas: el sistema no observa el movimiento en sí, sino sus efectos sobre la señal inalámbrica. En segundo lugar, las medidas CSI son sensibles a pequeñas variaciones del entorno, por lo que el modelo puede degradarse cuando se evalúa en sesiones de captura diferentes (Portolés Colón y Rupp, s.f.). En tercer lugar, el dataset disponible está condicionado por la dificultad práctica de adquirir muestras etiquetadas con un robot físico. Finalmente, si se desea integrar el sistema en un flujo online, el clasificador debe generar predicciones estables sobre ventanas temporales sucesivas y no únicamente sobre ficheros estáticos previamente segmentados.

Por tanto, el trabajo no se limita a entrenar un modelo de clasificación, sino que estudia la cadena completa necesaria para convertir una señal inalámbrica en información semántica sobre el comportamiento del robot.

1.3. Solución propuesta

Para abordar el problema planteado se propone un sistema compuesto por cuatro bloques principales: adquisición mmWave, construcción del dataset, clasificación mediante modelos de aprendizaje automático e integración funcional en un pipeline online con gemelo digital.

El primer bloque corresponde al sistema experimental de adquisición. Se configura un enlace mmWave entre dos antenas MikroTik wAP 60G utilizando herramientas de investigación basadas en OpenWRT que permiten extraer medidas CSI del canal (Mateo y Widmer, 2021). Las capturas se almacenan en ficheros de texto y se organizan por clase de comportamiento.

El segundo bloque transforma las medidas crudas en una representación apta para aprendizaje automático. Cada muestra se construye a partir de una ventana temporal de 120 medidas consecutivas. De cada medida se extraen los valores de fase y magnitud relevantes, se descartan campos de control no utilizados por el clasificador y se aplica un preprocesado específico para representar la fase de forma adecuada.

El tercer bloque corresponde al diseño y evaluación de los modelos de clasificación. Se comparan distintas familias de modelos, incluyendo aproximaciones clásicas, redes neuronales densas, redes convolucionales, modelos basados en atención y una arquitectura convolucional residual denominada CsiNet. Esta última se utiliza como modelo principal

del sistema por su equilibrio entre rendimiento, estructura específica para tensores CSI, extensibilidad y facilidad de integración en inferencia online.

El cuarto bloque corresponde a la validación funcional del pipeline online. El modelo entrenado se integra en un flujo que lee medidas CSI, mantiene una ventana temporal, ejecuta inferencia, estabiliza la predicción y publica el estado inferido mediante un fichero JavaScript Object Notation (JSON). Este estado puede ser consumido por un gemelo digital desarrollado en Isaac Sim, permitiendo visualizar el comportamiento inferido del robot en un entorno virtual.

La Figura 1.1 resume la arquitectura general del sistema propuesto.

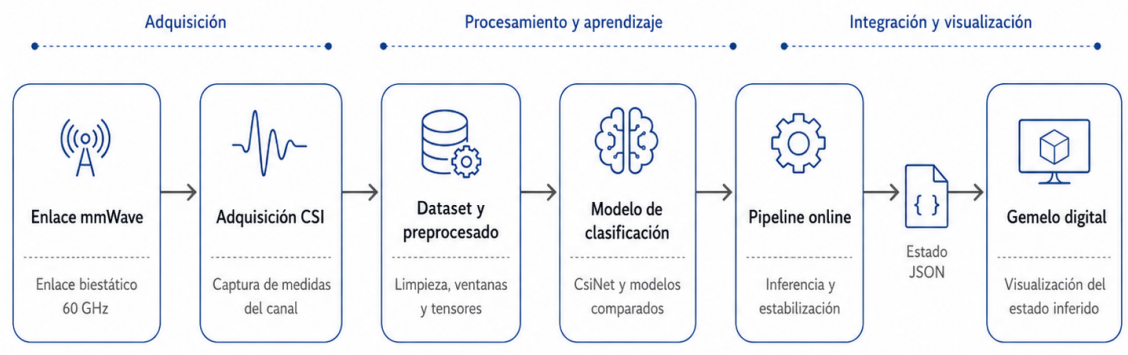


Figura 1.1: Arquitectura general del sistema propuesto. Imagen generada por Inteligencia Artificial

1.4. Contribuciones principales

Las contribuciones principales de este trabajo son las siguientes:

1. **Configuración de un sistema experimental de sensing mmWave**: se configura un enlace inalámbrico biestático a 60 GHz utilizando dispositivos comerciales MikroTik wAP 60G y herramientas de investigación que permiten acceder a medidas CSI del canal.

2. **Construcción de un dataset propio de comportamientos robóticos**: se recopilan y organizan medidas CSI asociadas a comportamientos predefinidos de un robot cuadrúpedo Unitree Go1, incluyendo posturas estáticas y secuencias dinámicas.

3. **Diseño de una representación temporal para señales CSI**: se desarrolla un pipeline de limpieza, validación y transformación de las medidas crudas en tensores

temporales aptos para modelos de aprendizaje automático.

4. **Comparativa de modelos de clasificación**: se evalúan diferentes familias de modelos, incluyendo SVM-RBF, Multilayer Perceptron (MLP), CNN simple, Transformer reducido y CsiNet, analizando su rendimiento en partición aleatoria estratificada.

5. **Análisis de generalización intersesión**: se estudia de forma explícita la degradación de rendimiento cuando el modelo se entrena en una sesión de captura y se evalúa en otra, identificando el *session shift* como uno de los principales retos del sistema.

6. **Diseño e integración de un pipeline de inferencia online**: se implementa una cadena funcional que consume medidas CSI, genera predicciones estabilizadas y publica el estado inferido mediante un fichero JSON.

7. **Demostración con gemelo digital**: se conecta la salida semántica del clasificador con un entorno de Isaac Sim, mostrando la posibilidad de utilizar el gemelo digital como plataforma de visualización e integración de diferentes fuentes de sensado.

Estas contribuciones permiten abordar el problema desde una perspectiva completa: adquisición física, representación de datos, modelado mediante inteligencia artificial, evaluación experimental e integración funcional en un sistema de monitorización.

1.5. Estructura del documento

El resto del documento se organiza de la siguiente forma.

El Capítulo 2 presenta los objetivos del trabajo y la metodología seguida durante el desarrollo del sistema. En él se definen el objetivo principal, los objetivos específicos, la hipótesis de trabajo, las fases de desarrollo y las herramientas utilizadas.

El Capítulo 3 revisa el estado del arte relacionado con *wireless sensing*, CSI, mmWave sensing, aprendizaje automático para señales de canal, generalización entre sesiones y gemelos digitales en robótica.

El Capítulo 4 describe el sistema experimental y el proceso de adquisición del dataset. Se detallan el hardware empleado, la configuración del enlace mmWave, el formato de las medidas CSI, el protocolo de captura y el proceso de limpieza y preprocesado.

El Capítulo 5 presenta el diseño de los modelos de clasificación. Se describe la formulación supervisada del problema, las arquitecturas consideradas, el modelo CsiNet, las funciones de pérdida, las técnicas de aumento de datos y el pipeline de inferencia online.

El Capítulo 6 recoge la evaluación experimental del sistema. Se comparan distintas arquitecturas, se analizan variantes de CsiNet, se estudia la generalización intersesión, se interpretan matrices de confusión y se presenta la validación funcional del pipeline online y del gemelo digital.

Finalmente, el Capítulo 7 resume las conclusiones del trabajo, expone sus principales limitaciones y plantea líneas futuras de investigación.

2. Objetivos y Metodología

Este capítulo define los objetivos del trabajo y describe la metodología seguida para alcanzarlos. El trabajo se plantea como un desarrollo experimental de inteligencia artificial (IA) aplicada: se configura un sistema real de adquisición de medidas CSI en banda mmWave, se construye un conjunto de datos etiquetado a partir de un robot cuadrúpedo, se diseñan modelos de clasificación supervisada, se evalúan cuantitativamente en modo offline y se integran posteriormente en un pipeline de inferencia online como validación funcional del sistema. La parte de comunicaciones inalámbricas constituye la infraestructura de sensado; la contribución principal se sitúa en la construcción y evaluación de un sistema de aprendizaje automático capaz de inferir comportamientos del robot a partir de secuencias temporales de canal.

2.1. Objetivo principal

El objetivo principal de este trabajo es **diseñar, implementar y evaluar un sistema de clasificación de comportamientos predefinidos de un robot cuadrúpedo a partir de secuencias temporales de CSI obtenidas mediante un enlace inalámbrico mmWave a 60 GHz, validando además su integración funcional en un pipeline de inferencia online y gemelo digital**.

Este objetivo combina tres dimensiones. En primer lugar, una dimensión experimental, asociada a la configuración de dos dispositivos MikroTik wAP 60G como enlace biestático y a la captura de medidas de canal en un entorno real. En segundo lugar, una dimensión de IA, centrada en el preprocesado de las señales, la construcción del dataset y el entrenamiento de modelos de aprendizaje supervisado. En tercer lugar, una dimensión de sistema, orientada a trasladar el modelo entrenado a un flujo de inferencia online y a demostrar su integración con una representación de gemelo digital.

La versión final desarrollada se centra en la discriminación de cuatro comportamientos del robot una vez presente en la zona de sensado: `de_pie`, `tumbado`, `dos_patas` y `baile`. Durante la campaña de adquisición también se capturaron muestras de control correspondientes a la condición `no_robot`; sin embargo, dicha condición no se incorpora al clasificador final, al priorizarse la distinción entre estados y movimientos del robot frente

a la detección de presencia/ausencia. Estos datos quedan disponibles para futuras extensiones del sistema hacia una arquitectura jerárquica de dos etapas: detección de presencia del robot y posterior clasificación de comportamiento.

2.2. Hipótesis de trabajo

La hipótesis de partida es que los comportamientos predefinidos del robot cuadrúpedo modifican de forma sistemática la respuesta del canal mmWave, y que dichas modificaciones quedan reflejadas en la evolución temporal de la fase y la magnitud de la CSI. Bajo esta hipótesis, una ventana temporal de medidas de canal puede interpretarse como una muestra multivariante:

$$X \in \mathbb{R}^{T \times B \times C}, \tag{2.1}$$

donde T es el número de medidas consecutivas, B representa los índices útiles de la respuesta de canal y C agrupa los canales de representación generados tras el preprocesado. En el sistema desarrollado se emplean ventanas de $T = 120$ medidas, construidas a partir de los valores de fase y magnitud extraídos de cada lectura CSI. El objetivo del modelo consiste en aprender una función parametrizada:

$$f_\theta : \mathbb{R}^{T \times B \times C} \longrightarrow \mathcal{Y}, \tag{2.2}$$

donde $\mathcal{Y} = \{\texttt{de_pie}, \texttt{tumbado}, \texttt{dos_patas}, \texttt{baile}\}$ es el conjunto de clases de comportamiento y θ son los parámetros aprendidos durante el entrenamiento.

2.3. Objetivos específicos

Para alcanzar el objetivo principal se definen los siguientes objetivos específicos:

OE1 — Configuración de la infraestructura mmWave. Configurar dos dispositivos MikroTik wAP 60G como enlace inalámbrico a 60 GHz, estableciendo una topología Access Point (AP)/Station (STA) estable y habilitando la extracción de medidas CSI mediante herramientas de investigación basadas en OpenWRT.

OE2 — Diseño del protocolo de adquisición. Definir un procedimiento de captura que permita obtener secuencias temporales de CSI asociadas a comportamientos

predefinidos del robot, considerando distintas sesiones de medida y ligeras variaciones experimentales para evaluar la sensibilidad del sistema a cambios del entorno.

OE3 — Construcción del dataset etiquetado. Generar un conjunto de datos compuesto por ficheros de texto con medidas CSI crudas, organizar las muestras por clase, validar la integridad de las capturas y construir una representación tensorial adecuada para modelos de aprendizaje automático.

OE4 — Preprocesado de señales CSI. Implementar un pipeline de limpieza y transformación que descarte medidas inválidas, extraiga los campos relevantes de fase y magnitud, normalice la información de canal y represente la fase de forma adecuada para evitar discontinuidades angulares.

OE5 — Diseño y entrenamiento de modelos de clasificación. Implementar y evaluar modelos de aprendizaje supervisado para clasificar los comportamientos del robot, con especial atención a arquitecturas convolucionales capaces de explotar patrones locales y temporales de la señal CSI.

OE6 — Evaluación offline y análisis de generalización. Medir el rendimiento de los modelos mediante métricas estándar de clasificación y analizar la degradación provocada por el desplazamiento de distribución entre sesiones de adquisición, identificado como uno de los principales retos del *sensing* inalámbrico.

OE7 — Inferencia online y estabilización temporal. Integrar el modelo entrenado en un pipeline de inferencia sobre flujo continuo de CSI, utilizando ventanas deslizantes, suavizado temporal y escritura de resultados en un fichero de estado compartido, con el objetivo de validar funcionalmente el despliegue del sistema más allá de la evaluación offline.

OE8 — Demostración con gemelo digital. Conectar la salida semántica del clasificador con una representación del robot en Isaac Sim, de forma que el comportamiento inferido pueda visualizarse en un entorno de gemelo digital como demostración complementaria del sistema.

2.4. Metodología de trabajo

La metodología seguida se estructura en seis fases. Aunque se presentan de forma secuencial, varias de ellas se desarrollan de manera iterativa: la captura de datos condiciona el preprocesado, los resultados de entrenamiento revelan problemas de generalización y la inferencia online obliga a ajustar tanto el formato de entrada como el mecanismo de estabilización temporal.

2.4.1. Fase 1: estudio previo y formulación del problema

La primera fase comprende la revisión bibliográfica y técnica del problema. Se estudian trabajos sobre ISAC, *sensing* basado en CSI, sensado mmWave, clasificación de series temporales y gemelos digitales. Paralelamente, se analiza la viabilidad de emplear dispositivos comerciales MikroTik a 60 GHz como fuente de información de canal. A partir de este estudio se formula el problema como una tarea de clasificación supervisada sobre ventanas temporales de CSI, en lugar de como una tarea de estimación geométrica o reconstrucción de trayectoria.

2.4.2. Fase 2: configuración del sistema experimental

La segunda fase consiste en configurar el entorno físico de adquisición. Se emplean dos dispositivos MikroTik wAP 60G, uno actuando como AP y otro como STA, directamente enfrentados en el entorno de laboratorio. El robot se sitúa en la zona intermedia del enlace, de forma que sus posturas y movimientos perturban las componentes multitrayecto del canal. La configuración se mantiene controlada, aunque se introducen ligeras variaciones angulares y sesiones de captura independientes para evaluar la robustez del sistema ante cambios realistas.

2.4.3. Fase 3: adquisición y construcción del dataset

La tercera fase se centra en la adquisición de datos. Para cada comportamiento se ejecutan scripts de captura que recopilan medidas CSI y las almacenan en ficheros de texto. Cada muestra se construye a partir de una ventana temporal de 120 medidas consecutivas, lo que corresponde aproximadamente a un intervalo de entre 3 y 4 segundos de observación. El dataset final utilizado para entrenamiento y evaluación contiene las

clases `de_pie`, `tumbado`, `dos_patas` y `baile`; adicionalmente, se conservan capturas de la condición `no_robot` como datos de control no incorporados al clasificador final.

2.4.4. Fase 4: preprocesado y representación de entrada

La cuarta fase transforma las capturas crudas en tensores utilizables por los modelos. Cada línea válida de medida contiene campos de control, metadatos temporales, dirección Media Access Control (MAC), parámetros de medición y valores de fase y magnitud. El pipeline descarta la información no utilizada por el clasificador, extrae los 32 valores de fase y los 32 valores de magnitud, elimina medidas incompletas y genera una representación temporal normalizada. Dado que la fase es una variable circular, se representa mediante sus componentes seno y coseno, evitando discontinuidades artificiales entre valores próximos a 0 y 2π.

2.4.5. Fase 5: diseño, entrenamiento y evaluación de modelos

La quinta fase constituye el núcleo de IA del trabajo. Se entrenan modelos de clasificación supervisada sobre el dataset procesado, empleando particiones de entrenamiento, validación y prueba. El modelo principal se basa en una arquitectura convolucional residual, complementada con técnicas de regularización y estrategias orientadas a mitigar el desplazamiento entre sesiones. La evaluación se realiza mediante métricas estándar de clasificación y análisis de matrices de confusión, descritos en detalle en el Capítulo 6. Además de la evaluación con particiones aleatorias estratificadas, se analiza explícitamente la generalización entre sesiones, ya que la variabilidad temporal del canal constituye uno de los factores más críticos del sistema.

2.4.6. Fase 6: inferencia online e integración con gemelo digital

La sexta fase traslada el modelo entrenado a un escenario operativo. El pipeline online lee medidas CSI de forma continua, mantiene un buffer deslizante de 120 muestras, genera ventanas temporales compatibles con la entrada del modelo, ejecuta inferencia sobre dichas ventanas y aplica estabilización temporal para reducir oscilaciones de predicción. La salida se escribe en un fichero JSON que contiene la clase inferida y metadatos asociados. Este fichero es consumido por una demostración en Isaac Sim, donde el estado semántico del robot se refleja mediante comportamientos predefinidos en el gemelo digital.

2.5. Planificación temporal

La Figura 2.1 resume la planificación temporal seguida durante el desarrollo del TFM. La planificación combina tareas de revisión bibliográfica, configuración experimental, adquisición de datos, desarrollo de modelos, evaluación e integración.

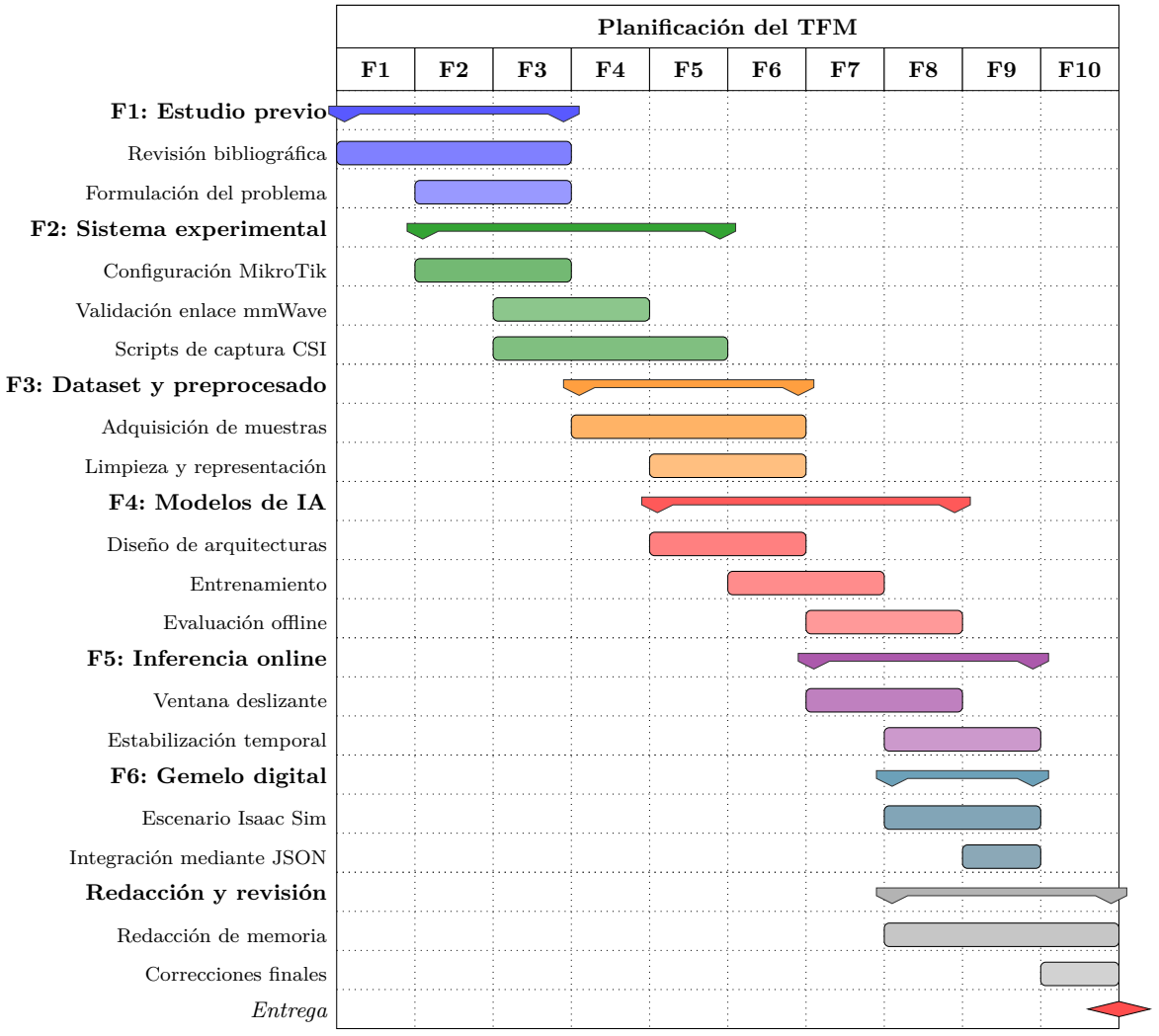


Figura 2.1: Planificación temporal del TFM.

2.6. Herramientas y entorno de desarrollo

La Tabla 2.1 resume las principales herramientas empleadas en el desarrollo del trabajo.

Tabla 2.1: Herramientas y tecnologías empleadas en el desarrollo del TFM.

Componente	**Herramienta / tecnología**	**Uso principal**
Hardware mmWave	MikroTik wAP 60G	Enlace biestático a 60 GHz y captura de medidas de canal.
Firmware y herramientas de captura	Mikrotik-researcher-tools	Extracción de medidas CSI desde los dispositivos configurados para investigación.
Adquisición de datos	Bash, SSH/SCP, scripts Python (Van Rossum y Drake, 2009)	Captura, transferencia, almacenamiento y organización de los ficheros de medidas.
Procesado de datos	Python, NumPy (Harris y cols., 2020)	Parsing, limpieza, validación y generación de tensores de entrada.
Modelos profundos	PyTorch (Paszke y cols., 2019a)	Entrenamiento y evaluación de redes neuronales para clasificación.
Modelos clásicos y métricas	scikit-learn (Pedregosa y cols., 2011), Matplotlib (Hunter, 2007)	Implementación de SVM, PCA, evaluación, matrices de confusión y generación de gráficas.
Robot físico	Unitree Go1	Generación de posturas y comportamientos reales durante la adquisición.
Gemelo digital	NVIDIA Isaac Sim (NVIDIA Corporation, s.f.)	Visualización semántica del comportamiento inferido en un entorno virtual.
Modelado 3D	Blender (Blender Foundation, s.f.)	Preparación de elementos del entorno y recursos visuales.
Intercambio de estado	JSON	Comunicación desacoplada entre el pipeline de inferencia y el entorno de simulación.
Control de versiones	Git/GitHub (Software Freedom Conservancy, s.f.; GitHub, Inc., s.f.)	Gestión del código, experimentos y evolución del proyecto.

2.7. Consideraciones éticas y de seguridad

El sistema desarrollado adquiere información del entorno a partir de la respuesta del canal inalámbrico entre dos dispositivos mmWave. No se recopilan imágenes, audio ni identificadores biométricos de personas (Seifeldin, Saeed, Kosba, El-Keyi, y Youssef, 2013). El objetivo del sistema es clasificar comportamientos de un robot cuadrúpedo y no identificar individuos ni inferir información personal. Por tanto, las muestras utilizadas en el trabajo se consideran datos técnicos de canal asociados al experimento.

Aun así, el *sensing* inalámbrico plantea retos éticos relevantes, ya que las señales de radio pueden reflejar la presencia o movimiento de objetos y personas en el entorno. Por ello, el alcance del presente trabajo se limita a un escenario controlado de laboratorio y a comportamientos ejecutados por un robot. Las capturas de control sin robot no se utilizan para identificar presencia humana, sino como referencia experimental. La potencia de transmisión y el uso de la banda de 60 GHz se mantienen dentro de las condiciones normales de operación del hardware empleado (ETSI, 2021).

2.8. Impacto socioeconómico y científico

El sistema propuesto presenta interés desde una perspectiva científica, tecnológica y socioeconómica. Aunque el trabajo se desarrolla como una prueba de concepto experimental, la línea de investigación en la que se enmarca tiene implicaciones relevantes para futuros sistemas de monitorización, robótica autónoma e infraestructuras inteligentes.

Desde el punto de vista científico, el trabajo explora el uso de CSI en banda mmWave para clasificar comportamientos de un robot cuadrúpedo, un escenario menos tratado que la detección de presencia humana, la localización o el reconocimiento de actividades en entornos domésticos. Esta aportación permite estudiar hasta qué punto una señal inalámbrica de comunicación puede convertirse en una fuente de percepción externa para sistemas robóticos.

Desde el punto de vista de la IA, el proyecto contribuye al estudio de modelos de clasificación sobre señales temporales de canal afectadas por variabilidad entre sesiones. Este aspecto resulta especialmente relevante porque muchos sistemas de aprendizaje automático alcanzan buenos resultados en condiciones controladas, pero se degradan cuando cambian las condiciones de adquisición. En este trabajo, dicha limitación no se oculta, sino

que se analiza explícitamente mediante protocolos de evaluación intersesión.

Desde una perspectiva tecnológica, el sistema abre la posibilidad de complementar los sensores embarcados del robot con una fuente externa de percepción no visual. Esto puede resultar útil en escenarios donde las cámaras no sean deseables por privacidad, donde existan problemas de iluminación, o donde se requiera una capa adicional de redundancia para supervisar el comportamiento de un sistema robótico.

En términos socioeconómicos, este tipo de tecnologías podría tener impacto en sectores como la industria, la logística, la inspección de infraestructuras, la robótica de servicio o la monitorización de entornos críticos. Un sistema capaz de inferir comportamientos o estados de robots mediante señales inalámbricas podría facilitar tareas de supervisión remota, detección de anomalías, mantenimiento predictivo o integración con plataformas de gemelo digital.

Además, el uso de hardware comercial Commercial Off-The-Shelf (COTS) reduce potencialmente la barrera de entrada frente a soluciones basadas en equipamiento radar especializado. Aunque el sistema desarrollado todavía requiere investigación adicional para alcanzar robustez en entornos reales complejos, demuestra que es posible construir una cadena funcional de sensado, clasificación e integración virtual utilizando dispositivos accesibles y herramientas de código abierto o investigación.

Finalmente, la integración con un gemelo digital refuerza el valor aplicado del sistema. La salida del clasificador no se limita a una predicción aislada, sino que puede alimentar una representación virtual del entorno, abriendo la puerta a plataformas donde confluyan distintas fuentes de sensado: mmWave, 5G, cámaras, LiDAR, Inertial Measurement Unit (IMU) u otros sensores. Esta capacidad de integración puede ser especialmente valiosa en sistemas ciberfísicos, donde la toma de decisiones depende de combinar información heterogénea en tiempo real (Orive, López, Estévez, Orive, y Marcos, 2021).

La principal limitación esperada sigue siendo la sensibilidad del canal mmWave a microvariaciones del entorno. El trabajo no solo busca obtener un clasificador con buen rendimiento en condiciones controladas, sino también identificar hasta qué punto dicho rendimiento se mantiene cuando cambian las condiciones de adquisición. Esta distinción entre capacidad discriminativa y robustez real es clave para valorar la madurez tecnológica del sistema.

3. Estado del Arte

Este capítulo revisa los fundamentos y trabajos relacionados que sustentan el trabajo realizado. El objetivo no es presentar una revisión exhaustiva de todas las áreas implicadas, sino justificar las decisiones técnicas adoptadas: el uso de CSI como fuente de percepción, la elección de la banda mmWave a 60 GHz, la formulación del problema como clasificación supervisada de secuencias temporales y la utilización de modelos de aprendizaje profundo para explotar patrones complejos en la señal.

El trabajo se sitúa en la intersección de tres líneas de investigación: el *sensing* inalámbrico y el paradigma ISAC, la percepción externa de sistemas robóticos y los modelos de IA para series temporales multivariantes. La aportación diferencial frente a trabajos previos no reside únicamente en utilizar señales inalámbricas para sensado, sino en aplicarlas a la clasificación de comportamientos predefinidos de un robot cuadrúpedo mediante un sistema experimental completo que incluye adquisición, dataset, entrenamiento offline, inferencia online y demostración con gemelo digital.

3.1. Del sensing inalámbrico al paradigma ISAC

Las redes inalámbricas se han diseñado tradicionalmente para transmitir información entre dispositivos. Sin embargo, toda señal de radio que se propaga por un entorno físico se ve afectada por la geometría, los objetos presentes y sus movimientos. En consecuencia, la señal recibida no solo transporta datos de comunicación, sino también información implícita sobre el entorno. Esta observación fundamenta el paradigma ISAC, que propone reutilizar señales de comunicación para realizar simultáneamente funciones de sensado y transmisión de datos (Liu y cols., 2022).

En una representación simplificada, la respuesta compleja del canal inalámbrico puede modelarse como la suma de múltiples trayectorias de propagación:

$$H(f,t) = \sum_{\ell=1}^{L} \alpha_\ell(t) e^{-j2\pi d_\ell(t)/\lambda}, \tag{3.1}$$

donde L es el número de trayectorias relevantes, $\alpha_\ell(t)$ representa la atenuación compleja de la trayectoria ℓ, $d_\ell(t)$ es su longitud efectiva y λ es la longitud de onda. Cuando un objeto se mueve o modifica su postura, altera alguna de estas trayectorias: puede bloquear

el camino directo, modificar reflexiones existentes o generar nuevas componentes multitrayecto. Estas variaciones se reflejan en la fase y la magnitud de $H(f, t)$, dando lugar a patrones temporales potencialmente discriminativos.

La CSI proporciona precisamente una estimación de esta respuesta de canal. En sistemas Wi-Fi convencionales, la CSI se ha empleado para tareas como detección de presencia, reconocimiento de actividad, estimación de respiración o identificación de personas. La evolución reciente hacia redes 5G/6G y estándares como Institute of Electrical and Electronics Engineers (IEEE) 802.11bf refuerza esta tendencia, al considerar explícitamente el sensado como una funcionalidad integrada en futuras redes inalámbricas (Picazo-Martinez y cols., 2025).

3.2. CSI como representación del entorno

La principal ventaja de la CSI frente a indicadores agregados como la potencia recibida es que conserva información más rica sobre la estructura del canal. Mientras que una métrica escalar puede indicar que la señal se ha atenuado, la CSI permite observar cómo se modifica la respuesta compleja en distintos componentes de la medida. En el contexto de este trabajo, cada lectura proporcionada por la plataforma experimental contiene valores de fase y magnitud asociados a la respuesta del enlace (Ma, Zhou, y Wang, 2019).

La fase resulta especialmente sensible a pequeñas variaciones geométricas porque depende de la relación entre la distancia recorrida por la onda y la longitud de onda. En banda mmWave, donde λ es del orden de milímetros, desplazamientos pequeños pueden producir variaciones apreciables de fase. Esta sensibilidad es una oportunidad para detectar movimientos sutiles, pero también una fuente de dificultad: pequeños cambios de temperatura, posición, orientación o entorno pueden desplazar la distribución de los datos entre sesiones (Quevedo Gutierrez, 2024).

Por esta razón, la CSI no se interpreta en este trabajo como una observación aislada, sino como una secuencia temporal. Una muestra no corresponde a una única medida instantánea, sino a una ventana de medidas consecutivas. Esta decisión permite que el modelo aprenda la evolución del canal durante varios segundos, capturando tanto el estado estático del robot como la firma temporal de movimientos predefinidos.

3.3. Sensing con Wi-Fi en bandas sub-7 GHz

Los primeros sistemas ampliamente estudiados de sensado inalámbrico se apoyaron en dispositivos Wi-Fi operando en las bandas de 2.4 GHz y 5 GHz. Estas bandas ofrecen bajo coste, amplia disponibilidad y mayor capacidad de penetración que las ondas milimétricas, lo que facilita su uso en escenarios interiores. Trabajos recientes han mostrado que la CSI en estas bandas puede emplearse para reconocimiento de actividad humana e identificación, utilizando tanto métodos clásicos como arquitecturas de aprendizaje profundo (Wei, Chen, y cols., 2025).

No obstante, las bandas sub-7 GHz presentan limitaciones cuando se pretende discriminar movimientos de pequeña escala o patrones cinemáticos complejos. Su longitud de onda es mayor que en 60 GHz, por lo que la resolución espacial y la sensibilidad angular son menores. Además, la congestión espectral y la presencia de múltiples redes en entornos interiores pueden introducir interferencias adicionales.

Estas limitaciones no invalidan el *sensing* sub-7 GHz; al contrario, constituye una base metodológica importante para el presente trabajo. De esta línea se heredan la idea de utilizar la CSI como representación del entorno, la formulación como problema supervisado y el uso de redes neuronales para aprender características directamente de la señal. La diferencia principal es que aquí se explota una banda mmWave, más sensible a variaciones geométricas pequeñas pero también más dependiente de la alineación y de la estabilidad del entorno.

3.4. Sensing con ondas milimétricas a 60 GHz

Las bandas mmWave, y en particular la banda de 60 GHz, ofrecen propiedades físicas especialmente atractivas para sensado. La longitud de onda aproximada de 5 mm permite que desplazamientos o cambios posturales pequeños produzcan variaciones medibles en el canal. Además, los grandes anchos de banda disponibles en sistemas 802.11ad/ay facilitan tareas de localización y estimación de trayectorias en comparación con tecnologías de menor frecuencia (van Berlo y cols., 2021).

En el ámbito de la localización y el mapeo, waveSLAM mostró que radios comerciales de 60 GHz pueden emplearse para apoyar tareas de mapeo interior mediante mediciones de canal y distancia (Picazo y cols., 2023). Otros trabajos han explorado la combinación

de información mmWave y sub-7 GHz para detección de presencia, aprovechando la complementariedad entre alta resolución y mayor cobertura (Picazo-Martinez y cols., 2025).

A diferencia de muchos sistemas radar mmWave, el sistema empleado en este trabajo no genera directamente un mapa rango-Doppler ni proporciona una estimación explícita de velocidad radial (Bilik, Longman, Villeval, y Tabrikian, 2019). La información disponible para el clasificador procede de las variaciones temporales de fase y magnitud de la CSI expuesta por la plataforma de investigación. Por tanto, el movimiento debe inferirse de forma indirecta a partir de la evolución temporal de la respuesta del canal. Esta restricción justifica la representación mediante ventanas de $T = 120$ medidas y el uso de modelos capaces de aprender patrones temporales a partir de datos crudos o semiprocesados.

3.5. Plataformas comerciales para extracción de CSI mmWave

Uno de los retos del sensado mmWave es el acceso a información de bajo nivel del canal. Muchos dispositivos comerciales ocultan estas medidas tras el firmware del fabricante, lo que limita su uso en investigación. La plataforma M-Cube (M^3) demostró que es posible reprogramar radios comerciales 802.11ad para exponer información avanzada del array y de la respuesta de canal (Zhao, Woodford, Wei, Qian, y Zhang, 2020).

En la misma línea, las herramientas de investigación para dispositivos MikroTik desarrolladas por IMDEA Networks permiten utilizar modelos como wAP 60G, LHGG-60ad y wAP 60Gx3 AP con una versión modificada de OpenWRT. Estas herramientas habilitan por defecto la recolección de CSI y Fine Time Measurement (FTM), e incluyen utilidades para arrancar los dispositivos como AP/STA y lanzar comandos de medida sobre la interfaz inalámbrica (Mateo y Widmer, 2021).

La salida de CSI proporcionada por esta plataforma incluye campos de control y metadatos, seguidos de 32 valores de fase y 32 valores de magnitud. Esta estructura condiciona directamente el diseño del dataset del presente trabajo: la información útil se extrae de cada línea de medida, se agrupa temporalmente y se transforma en una representación tensorial adecuada para modelos de IA. Aunque la plataforma también puede proporcionar medidas FTM, el presente trabajo no las utiliza como entrada del clasificador, centrándose exclusivamente en la señal CSI.

3.6. Robots móviles y percepción externa

La robótica móvil depende habitualmente de sensores embarcados, como cámaras, LiDAR, IMU u odometría, para estimar el estado del robot y percibir el entorno (Thrun y cols., 2005). Estos sensores son esenciales para navegación y control, pero también presentan limitaciones: pueden verse afectados por oclusiones, iluminación, saturación de red, fallos de comunicación o pérdida de sincronización con sistemas externos.

La percepción inalámbrica externa ofrece una fuente complementaria de información. En lugar de depender exclusivamente de los sensores del robot, un enlace de radio situado en el entorno puede detectar perturbaciones asociadas a su presencia o movimiento. Esta aproximación no pretende sustituir los sensores embarcados, sino añadir redundancia y permitir que sistemas externos, como un gemelo digital, reciban información semántica incluso cuando el robot no publique directamente todos sus estados (Lara Hurtado, Fernández-Lozano, y Vázquez-Martín, 2019).

En el caso de robots cuadrúpedos, la clasificación de comportamientos resulta especialmente interesante porque la misma plataforma puede adoptar posturas estáticas y ejecutar movimientos dinámicos con firmas temporales distintas. No obstante, este trabajo no aborda la estimación continua de trayectoria ni el reconocimiento abierto de cualquier movimiento posible. El alcance se restringe a un conjunto cerrado de comportamientos predefinidos, lo que permite formular el problema como clasificación supervisada.

3.7. Aprendizaje automático para señales CSI

El uso de machine learning (ML) en señales CSI ha evolucionado desde enfoques basados en características manuales hacia arquitecturas de aprendizaje profundo (Wei, Jia, y cols., 2025). Los modelos clásicos, como las Support Vector Machine (SVM) con núcleo Radial Basis Function (RBF), han sido ampliamente utilizados cuando el tamaño del dataset es moderado y las características pueden resumirse mediante estadísticos o representaciones compactas (Cortes y Vapnik, 1995). Su principal ventaja es la simplicidad, pero dependen en gran medida de una buena ingeniería de características.

Las Convolutional Neural Network (CNN) permiten aprender representaciones directamente a partir de estructuras temporales o tiempo-frecuencia. En el caso de señales CSI, las convoluciones pueden capturar patrones locales en la evolución temporal de la

magnitud y la fase. Las arquitecturas residuales, introducidas originalmente para visión artificial, facilitan el entrenamiento de redes más profundas mediante conexiones de salto que reducen problemas de degradación del gradiente (He, Zhang, Ren, y Sun, 2016). Aunque fueron propuestas en otro dominio, su principio de aprendizaje residual es aplicable a señales temporales organizadas como tensores.

Las redes recurrentes, como Long Short-Term Memory (LSTM) o Gated Recurrent Unit (GRU), modelan explícitamente dependencias temporales, pero su entrenamiento es menos paralelizable y puede ser menos eficiente en ventanas largas. Los Transformers y mecanismos de atención han demostrado gran capacidad para representar dependencias de largo alcance en secuencias (Vaswani y cols., 2017), y se han adaptado también a series temporales multivariantes (Zerveas, Jayaraman, Patel, Bhamidipaty, y Eickhoff, 2021; Nie, Nguyen, Sinthong, y Kalagnanam, 2023). Sin embargo, su mayor complejidad y demanda de datos puede no ser siempre la opción más adecuada cuando el dataset es limitado y el objetivo incluye inferencia online ligera.

En este trabajo, el estado del arte en Transformers se considera relevante como referencia metodológica y posible línea futura, pero la implementación principal se orienta hacia modelos convolucionales residuales, más adecuados para el tamaño del dataset y para un despliegue online eficiente. Esta decisión no implica que los modelos de atención sean inadecuados; más bien responde a una restricción práctica del trabajo: construir un clasificador robusto y ejecutable en tiempo real con datos experimentales de tamaño moderado.

3.8. Desplazamiento de dominio y generalización entre sesiones

Uno de los problemas más relevantes del *sensing* basado en CSI es el desplazamiento de dominio. El canal inalámbrico es altamente dependiente del entorno: pequeñas variaciones en la posición de los dispositivos, orientación de antenas, temperatura, presencia de objetos o configuración del laboratorio pueden modificar la distribución de las medidas. Como consecuencia, un modelo entrenado con datos de una sesión puede degradarse al evaluarse en una sesión posterior.

Este fenómeno es especialmente crítico en banda mmWave. La alta sensibilidad de la fase a cambios milimétricos permite detectar movimientos sutiles, pero también hace que

el modelo pueda aprender patrones específicos de una sesión en lugar de características invariantes del comportamiento. Por esta razón, la evaluación mediante particiones aleatorias no es suficiente. Es necesario analizar también escenarios de evaluación entre sesiones, donde el conjunto de prueba procede de una distribución temporalmente separada del entrenamiento.

En aprendizaje profundo se han propuesto distintas estrategias para mejorar la robustez ante cambios de dominio. Las redes adversariales de dominio buscan aprender representaciones útiles para la tarea principal pero poco informativas respecto al dominio de procedencia, normalmente mediante una capa de inversión de gradiente (Ganin y cols., 2016). Otras técnicas, como MixStyle, introducen diversidad artificial mezclando estadísticas de características entre muestras para favorecer la generalización fuera de distribución (Zhou, Yang, Qiao, y Xiang, 2021). Estas ideas inspiran parte del análisis experimental del presente trabajo, donde la generalización entre sesiones se considera una métrica central y no un aspecto secundario.

3.9. Gemelos digitales en robótica

El concepto de gemelo digital describe una réplica virtual de un sistema físico que se actualiza mediante datos procedentes del sistema real (Grieves, 2014). En robótica, los gemelos digitales permiten visualizar el estado del robot, validar comportamientos, reproducir escenarios y facilitar la supervisión de sistemas complejos. Herramientas como Isaac Sim proporcionan entornos 3D de alta fidelidad que pueden integrarse con robots físicos y pipelines externos.

La aportación del presente trabajo en este ámbito es acotada pero relevante: el gemelo digital se emplea como demostración de consumo de la salida semántica del clasificador. El sistema no pretende resolver todos los problemas de sincronización física ni reproducir con precisión dinámica cada movimiento real del robot. En su lugar, muestra que una predicción generada a partir de CSI puede escribirse en un fichero JSON y ser utilizada para actualizar comportamientos predefinidos del robot virtual. De esta forma, el gemelo digital actúa como interfaz de visualización y validación cualitativa del pipeline online.

3.10. Síntesis crítica y posicionamiento del trabajo

La literatura revisada muestra que el *sensing* inalámbrico basado en CSI es una línea consolidada para detección de presencia y reconocimiento de actividad, especialmente en escenarios humanos. También muestra que la banda mmWave aporta una sensibilidad espacial superior y que existen plataformas comerciales modificadas capaces de exponer medidas de canal útiles para investigación. Sin embargo, se identifican tres huecos relevantes.

En primer lugar, la mayoría de trabajos se centran en personas, localización o mapeo, y no en la clasificación de comportamientos de robots cuadrúpedos. En segundo lugar, muchas soluciones reportan buenos resultados en escenarios controlados, pero no siempre analizan con suficiente profundidad la generalización entre sesiones, aspecto crítico en CSI mmWave. En tercer lugar, la conexión entre clasificación semántica mediante canal inalámbrico e integración con un gemelo digital robótico continúa siendo una línea poco explorada.

El presente trabajo se posiciona precisamente en ese espacio. Aporta un sistema experimental que captura CSI de un enlace 60 GHz con un robot cuadrúpedo en la zona de sensado, construye un dataset de comportamientos predefinidos, entrena modelos de clasificación supervisada y evalúa tanto el rendimiento offline como la viabilidad de inferencia online. La Tabla 3.1 resume la relación entre algunos trabajos representativos y el enfoque desarrollado.

Tabla 3.1: Comparativa de trabajos relacionados con el presente TFM.

Trabajo	**Banda**	**Objetivo**	**Modelo**	**Online**	**Robot/DT**
(Picazo-Martinez y cols., 2025)	60 GHz + sub-7 GHz	Presencia humana	CNN	No	No
(Picazo y cols., 2023)	60 GHz	Mapeo interior	SLAM + canal	Parcial	Robot móvil
(Zhao y cols., 2020)	60 GHz	Plataforma radio	MIMO software radio	No	No
(Wei, Chen, y cols., 2025)	sub-7 GHz	Actividad/identificación	DL variado	Parcial	No
Este trabajo	**60 GHz**	**Comportamiento robótico**	**CNN residual**	**Sí**	**Sí**

En síntesis, el trabajo no asume que la CSI mmWave sea una solución universal ni que

sustituya a sensores embarcados como cámaras, LiDAR u odometría. Su planteamiento es más específico: evaluar si las perturbaciones inducidas por un conjunto cerrado de comportamientos del robot son suficientemente discriminativas para construir un sistema de IA capaz de clasificarlas, y analizar de forma crítica qué limitaciones aparecen al trasladar esta idea desde una evaluación offline hacia un pipeline online.

4. Sistema Experimental y Adquisición del Dataset

Este capítulo describe el sistema experimental empleado para la adquisición de las medidas de CSI y la construcción del dataset utilizado posteriormente para el entrenamiento y evaluación de los modelos de clasificación. A diferencia del capítulo anterior, centrado en el estado del arte, aquí se documenta el trabajo desarrollado en el proyecto: la configuración del enlace mmWave, el protocolo de captura, el formato de las medidas obtenidas, la organización de los ficheros y el proceso de transformación de los datos crudos en muestras temporales aptas para aprendizaje automático.

El objetivo del sistema experimental no es únicamente registrar medidas de canal, sino construir una cadena completa que permita relacionar las perturbaciones inducidas por un robot cuadrúpedo en un enlace inalámbrico de 60 GHz con una etiqueta semántica de comportamiento. Por ello, el capítulo se estructura en tres bloques: primero se presenta la arquitectura física del sistema, después se describe la adquisición de CSI y finalmente se detalla la construcción del dataset utilizado por los modelos de clasificación.

4.1. Visión general del sistema experimental

El sistema experimental se basa en un enlace inalámbrico biestático formado por dos dispositivos MikroTik wAP 60G operando en la banda de 60 GHz. Uno de los dispositivos se configura como punto de acceso (AP) y el otro como estación (STA). Entre ambos se sitúa el perro robótico Unitree Go1, que ejecuta distintos estados o movimientos predefinidos dentro de la zona de mayor sensibilidad del enlace.

La hipótesis operacional es que cada comportamiento del robot modifica de forma diferente la propagación multitrayecto entre las antenas. Estas modificaciones afectan a la fase y a la magnitud de la CSI, generando patrones temporales que pueden ser aprendidos por un modelo supervisado. En lugar de depender de una cámara o de la odometría interna del robot, el sistema infiere el comportamiento a partir de la respuesta del canal inalámbrico.

La Figura 4.1 muestra la arquitectura general del sistema experimental.

Desde el punto de vista funcional, el sistema se divide en cinco etapas:

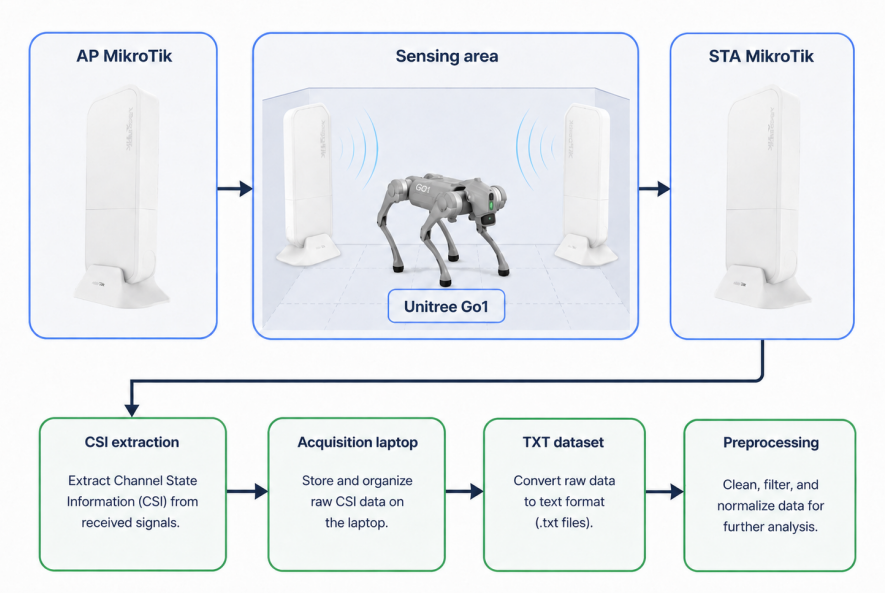


Figura 4.1: Arquitectura general del sistema experimental para adquisición de CSI en banda mmWave.

1. **Configuración del enlace mmWave**: establecimiento de una conexión AP/STA entre las dos antenas MikroTik.
2. **Ejecución del comportamiento del robot**: colocación del Unitree Go1 en la zona de sensing y ejecución de una postura o movimiento predefinido.
3. **Captura de medidas CSI**: extracción periódica de medidas de fase y magnitud del canal mediante las herramientas de investigación instaladas en las antenas.
4. **Almacenamiento en ficheros**: guardado de cada muestra temporal en un archivo `.txt`.
5. **Construcción del dataset**: limpieza, filtrado, normalización y transformación de los datos en tensores de entrada para los modelos de aprendizaje automático.

4.2. Hardware empleado

La Tabla 4.1 resume los principales elementos hardware utilizados en el sistema experimental.

Las antenas MikroTik se colocan enfrentadas, apuntando directamente una hacia la

Tabla 4.1: Hardware empleado en el sistema experimental.

Elemento	**Función en el sistema**
MikroTik wAP 60G	Dispositivos mmWave empleados para formar el enlace inalámbrico biestático a 60 GHz. Uno se configura como AP y el otro como STA.
Unitree Go1	Robot cuadrúpedo utilizado como objeto dinámico de estudio. Ejecuta los comportamientos que se desean clasificar.
Portátil de adquisición	Equipo desde el que se lanzan los scripts de captura, se almacenan los ficheros `.txt` y se ejecutan posteriormente los scripts de procesado y entrenamiento.
Red Ethernet local	Canal de control utilizado para acceder a las antenas, ejecutar comandos remotos y recuperar los ficheros de medidas.

otra, de forma que el robot queda situado aproximadamente en la región central del enlace. La distancia entre antenas se mantiene en torno a unos pocos metros, aproximadamente 2.7 m en la configuración experimental empleada. Esta geometría busca maximizar la sensibilidad del enlace a los cambios introducidos por el robot, manteniendo al mismo tiempo un enlace suficientemente estable para la adquisición continua de medidas.

Durante la captura se introducen pequeñas variaciones de orientación de las antenas, del orden de $\pm 2^{\circ}$, con el objetivo de evitar que el modelo aprenda únicamente una configuración geométrica extremadamente concreta. No obstante, estas variaciones se mantienen deliberadamente pequeñas (IEEE, 2012). En enlaces de 60 GHz, la alta direccionalidad del haz hace que giros mayores puedan modificar de forma significativa la trayectoria dominante, favoreciendo incluso reflexiones en paredes u otros elementos del entorno. En ese caso, la señal medida dejaría de representar de forma consistente la interacción directa entre el robot y el enlace principal.

La Figura 4.2 muestra la disposición experimental que se debe generar posteriormente como esquema.

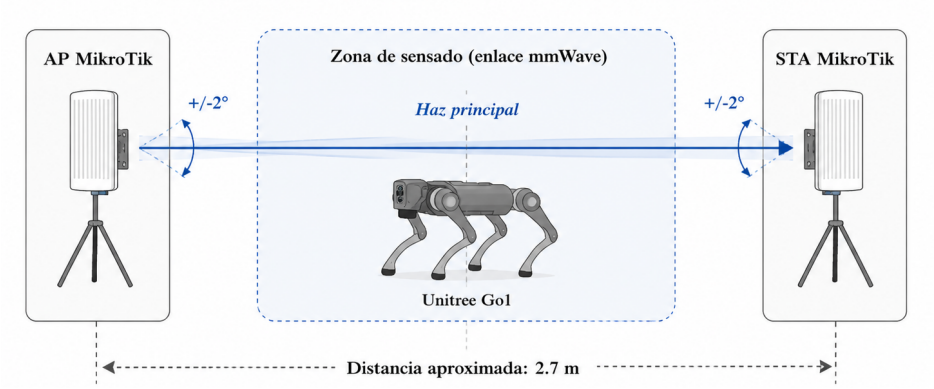


Figura 4.2: Disposición experimental de las antenas MikroTik y el robot Unitree Go1 durante la adquisición del dataset.

4.3. Configuración del enlace mmWave

Para acceder a medidas internas del canal, las antenas MikroTik se configuraron con una versión de investigación de OpenWrt para dispositivos wAP 60G, LHGG-60ad y wAP 60Gx3 AP. Esta versión, desarrollada por IMDEA Networks, habilita la extracción de medidas CSI y FTM, además de permitir la ejecución de herramientas de red estándar sobre el dispositivo (Mateo y Widmer, 2021).

Aunque la plataforma permite obtener tanto CSI como FTM, en este trabajo la clasificación se basa únicamente en la CSI. El FTM queda fuera de la entrada del modelo porque el objetivo no es estimar distancia ni localizar el robot, sino clasificar comportamientos a partir de la evolución temporal de la fase y la magnitud del canal.

La configuración básica del enlace consta de dos roles:

- **AP**: dispositivo encargado de anunciar la red inalámbrica de 60 GHz.
- **STA**: dispositivo que se asocia al punto de acceso y desde el que se solicitan las medidas de canal hacia el peer correspondiente.

De forma esquemática, el arranque del enlace se realiza mediante los servicios habituales de Linux para este tipo de configuración:

```
# En el dispositivo configurado como AP
hostapd -B /etc/hostapd.conf
```

```
# En el dispositivo configurado como STA
wpa_supplicant -D nl80211 -i wlan0 -c /etc/wpa_supplicant.conf -B
```

Una vez establecida la conexión, la extracción de medidas se realiza mediante comandos específicos del driver, que generan salidas en el registro del sistema. Estas salidas contienen una línea por medida y son recuperadas mediante scripts de captura. El flujo general consiste en solicitar medidas desde la antena, leer la salida generada, filtrar las líneas válidas y almacenarlas en ficheros de texto.

4.4. Formato de las medidas CSI

Cada medida válida aparece en el fichero de captura como una línea que contiene el marcador `Measurement`. La salida cruda incluye metadatos de control, marca temporal, dirección MAC del peer, parámetros internos de la medida y, finalmente, los valores de fase y magnitud que se utilizan para construir la entrada del modelo.

Un ejemplo simplificado de línea de medida es el siguiente:

```
[3923.520639] [AOA] Measurement:
0,1619025082.466828,48:8f:5a:df:02:3b,2,1,2,0,128,225,55,292,...,53
```

La estructura de la línea se resume en la Tabla 4.2.

Tabla 4.2: Estructura de una línea de medida CSI generada por la plataforma de captura.

Bloque	**Número de campos**	**Descripción**
Metadatos	8	Código de retorno, timestamp, dirección MAC, canal, tipo de medida, máscara RF, estado y longitud.
Fase	32	Valores cuantificados de fase asociados a la respuesta del canal.
Magnitud	32	Valores de magnitud asociados a la respuesta del canal.

Por tanto, aunque cada línea contiene 72 campos separados por comas, solo los 64 últimos corresponden directamente a información de canal: 32 valores de fase y 32 valores

de magnitud. Los campos anteriores se emplean para control, validación y trazabilidad de la muestra, pero no se introducen directamente en el modelo de clasificación.

La fase se entrega como un valor cuantificado entre 0 y 1024. Para evitar discontinuidades artificiales debidas al carácter circular de la fase, durante el preprocesado se transforma cada valor de fase $\phi_i(t)$ en un ángulo:

$$\theta_i(t) = \frac{2\pi\phi_i(t)}{1024}, \tag{4.1}$$

y posteriormente se representa mediante sus componentes seno y coseno:

$$\mathbf{p}_i(t) = \left[\sin\left(\theta_i(t)\right), \cos\left(\theta_i(t)\right)\right]. \tag{4.2}$$

Esta transformación evita que dos fases físicamente próximas, pero situadas a ambos lados del salto circular, sean interpretadas por el modelo como valores muy alejados.

4.5. Protocolo de adquisición del dataset

El dataset se construye a partir de múltiples capturas independientes. Cada captura se almacena en un fichero `.txt` y representa una secuencia temporal asociada a una única clase. La duración de cada muestra viene determinada por la captura de 120 medidas CSI, que corresponden aproximadamente a un intervalo de entre 3 s y 4 s, aunque existe cierta variabilidad debido al proceso de adquisición y al retardo de los comandos remotos.

Las clases consideradas inicialmente durante la captura fueron cinco:

- `de_pie`: robot en postura estática de pie.
- `tumbado`: robot en postura estática tumbada.
- `gesto1`: comportamiento dinámico posteriormente etiquetado como `dos_patas`.
- `gesto2`: comportamiento dinámico posteriormente etiquetado como `baile`.
- `no_robot`: condición de control sin el robot situado en la zona de sensing.

La clase `no_robot` se capturó para disponer de información de referencia sobre el canal en ausencia del robot. Sin embargo, el clasificador final desarrollado en este trabajo se centra en discriminar comportamientos del robot cuando este se encuentra presente en la zona de sensing. Por esta razón, `no_robot` se conserva como condición de control

disponible, pero no se incorpora a la versión final del modelo de cuatro clases. Esta decisión evita mezclar dos problemas distintos: detección de presencia del robot y clasificación de su comportamiento.

Las cuatro clases finales del problema de clasificación son:

$$\mathcal{Y} = \{\texttt{de_pie}, \texttt{tumbado}, \texttt{dos_patas}, \texttt{baile}\}. \tag{4.3}$$

La captura se realizó en varias sesiones temporales con el objetivo de introducir variabilidad natural en el dataset. Esta decisión es importante porque las señales mmWave son muy sensibles a pequeños cambios del entorno: variaciones en la posición exacta del robot, orientación de las antenas, temperatura, presencia de objetos cercanos o modificaciones mínimas de la escena pueden alterar la distribución de la CSI. En consecuencia, realizar capturas en varias sesiones permite estudiar posteriormente hasta qué punto el modelo generaliza más allá de las condiciones concretas de una única adquisición.

4.6. Organización del dataset

Los ficheros de captura se organizaron en una estructura de carpetas por clase. Cada fichero `.txt` contiene las líneas crudas generadas durante una adquisición de 120 medidas. La nomenclatura de los archivos permite identificar la clase y el número de muestra, por ejemplo:

```
all_120/
    de_pie/
        txt/
            csi_de_pie_120_001.txt
            csi_de_pie_120_002.txt
            ...
    tumbado/
        txt/
            csi_tumbado_120_001.txt
            ...
    gesto1/
        txt/
            csi_gesto1_120_001.txt
```

```
        ...
gesto2/
    txt/
        csi_gesto2_120_001.txt
        ...
no_robot/
    txt/
        csi_no_robot_120_001.txt
        ...
```

La Tabla 4.3 resume el número de ficheros disponibles en la estructura inicial de captura.

Tabla 4.3: Número de ficheros `.txt` disponibles por clase en el dataset crudo.

Clase cruda	**Uso**	**Número de ficheros**
`de_pie`	Clase final estática	696
`tumbado`	Clase final estática	695
`gesto1`	Clase final dinámica, renombrada como `dos_patas`	212
`gesto2`	Clase final dinámica, renombrada como `baile`	217
`no_robot`	Condición de control, no usada en el modelo final	498
Total		**2318**

Tras el proceso de limpieza y validación, el dataset final utilizado para el entrenamiento del clasificador queda formado por 1806 muestras válidas distribuidas en cuatro clases. La Tabla 4.4 muestra la distribución final de clases.

Tabla 4.4: Distribución de clases en el dataset final empleado para clasificación.

Clase final	**Tipo de comportamiento**	**Muestras válidas**
`de_pie`	Estático	692
`tumbado`	Estático	693
`dos_patas`	Dinámico	205
`baile`	Dinámico	216
Total		**1806**

La diferencia entre el número de ficheros crudos y el número final de muestras válidas se debe principalmente a tres factores: exclusión de la clase `no_robot`, descarte de ficheros sin suficientes medidas válidas y eliminación de líneas malformadas. Para evitar introducir secuencias demasiado incompletas, se exige un mínimo de 100 medidas válidas por fichero. Las muestras que superan este umbral se interpolan temporalmente hasta longitud fija $T = 120$, permitiendo que todos los ejemplos tengan la misma dimensión de entrada.

La Tabla 4.5 resume las principales estadísticas del proceso de construcción del dataset.

Tabla 4.5: Estadísticas del proceso de limpieza y validación del dataset.

Concepto	**Valor**
Ficheros crudos totales	2318
Ficheros de control `no_robot` no usados en el modelo final	498
Muestras finales válidas	1806
Ficheros descartados por longitud insuficiente	13
Ficheros descartados por ausencia de medidas válidas	1
Líneas malformadas detectadas durante el parsing	165
Longitud temporal objetivo por muestra	120 medidas CSI
Mínimo de medidas válidas exigido	100 medidas CSI

4.7. Distribución temporal de las capturas

Además de la distribución por clases, es relevante analizar la procedencia temporal de las muestras. La fecha de cada muestra se obtiene a partir del timestamp de la primera medida válida del fichero. Esta información permite estudiar el desplazamiento de dominio entre sesiones, uno de los principales retos detectados en el trabajo.

La Tabla 4.6 muestra la distribución de muestras válidas por día de adquisición.

Tabla 4.6: Distribución de muestras válidas por sesión temporal y clase.

Día	`de_pie`	`tumbado`	`dos_patas`	`baile`
2026-04-21	197	191	174	176
2026-04-24	100	103	31	40
2026-05-05	395	399	0	0

Esta distribución muestra dos aspectos importantes. En primer lugar, el dataset no es completamente equilibrado: las clases estáticas tienen más muestras que las clases dinámicas. En segundo lugar, no todas las sesiones contienen todas las clases. Esta característica condiciona el diseño de las evaluaciones posteriores, especialmente las pruebas de generalización entre días, ya que solo las sesiones que contienen todas las clases permiten realizar una evaluación intersesión completa.

El análisis de esta variabilidad temporal se retoma en el Capítulo 6, donde se comparan resultados con partición aleatoria estratificada y resultados con separación por día de captura.

4.8. Modelo matemático de la señal de entrada

La señal medida puede interpretarse a partir de un modelo de canal multitrayecto. En un instante t, la respuesta compleja del canal puede expresarse como:

$$H(t) = \sum_{\ell=1}^{L} \alpha_\ell(t) e^{-j2\pi d_\ell(t)/\lambda}, \tag{4.4}$$

donde L representa el número de trayectorias relevantes, $\alpha_\ell(t)$ es la atenuación compleja asociada a la trayectoria ℓ, $d_\ell(t)$ es su longitud efectiva y λ es la longitud de onda. En la banda de 60 GHz, la longitud de onda es aproximadamente:

$$\lambda = \frac{c}{f} \approx \frac{3 \cdot 10^8}{60 \cdot 10^9} = 5\,\text{mm}. \tag{4.5}$$

Esta longitud de onda reducida explica la alta sensibilidad del sistema ante pequeños desplazamientos del robot o cambios sutiles en la escena. Un movimiento de pocos milímetros puede inducir variaciones apreciables en la fase del canal, lo que convierte la CSI en una fuente de información adecuada para detectar patrones de comportamiento, pero también hace que el sistema sea vulnerable a variaciones no controladas del entorno.

En cada instante t, la plataforma proporciona 32 valores de fase y 32 valores de magnitud. Sea $m_i(t)$ la magnitud asociada al índice i y $\phi_i(t)$ el valor de fase cuantificado, con $i = 1, \ldots, 32$. La representación cruda de una medida puede escribirse como:

$$\mathbf{r}(t) = [\phi_1(t), \ldots, \phi_{32}(t), m_1(t), \ldots, m_{32}(t)]\,. \tag{4.6}$$

Sin embargo, la entrada final del modelo no utiliza directamente esta concatenación cruda. La magnitud se transforma mediante una escala logarítmica y se normaliza, mientras que la fase se representa mediante seno y coseno. Así, para cada índice i se construye un vector de tres componentes:

$$\mathbf{z}_i(t) = [\widetilde{m}_i(t), \sin(\theta_i(t)), \cos(\theta_i(t))]\,, \tag{4.7}$$

donde $\widetilde{m}_i(t)$ representa la magnitud transformada y normalizada, y $\theta_i(t)$ es el ángulo definido en la Ecuación 4.1.

Por tanto, cada muestra del dataset se representa como un tensor tridimensional:

$$X = \{\mathbf{z}(1), \mathbf{z}(2), \ldots, \mathbf{z}(T)\} \in \mathbb{R}^{T \times 32 \times 3}, \qquad T = 120. \tag{4.8}$$

Esta formulación es especialmente adecuada para modelos convolucionales, ya que conserva simultáneamente la estructura temporal, la estructura interna de los 32 valores de canal y la separación entre magnitud y fase circular.

4.9. Proceso de limpieza y preprocesado

El proceso de transformación desde los ficheros `.txt` crudos hasta el tensor de entrada se resume en la Figura 4.3.

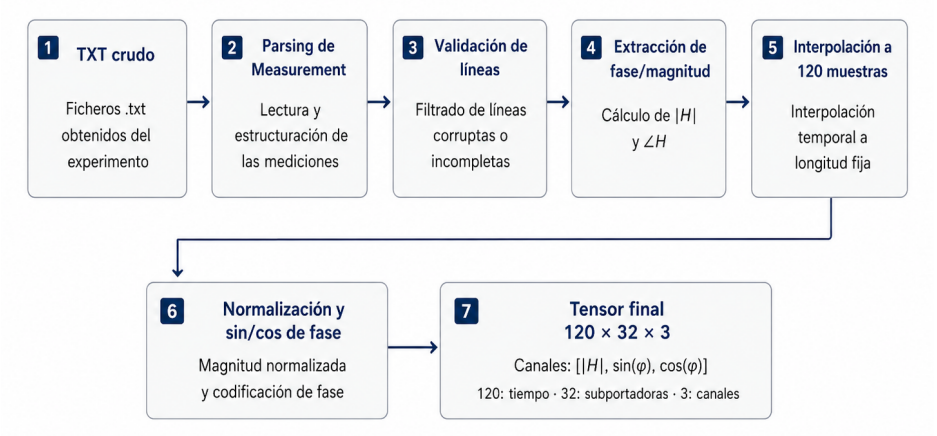


Figura 4.3: Pipeline de construcción del dataset a partir de ficheros `.txt` crudos.

El procedimiento aplicado es el siguiente:

1. **Lectura del fichero**: se recorren todas las líneas del archivo `.txt`.
2. **Filtrado por marcador**: se conservan únicamente las líneas que contienen el marcador `Measurement`.
3. **Validación de longitud**: se comprueba que la línea contenga exactamente 72 campos separados por comas.
4. **Extracción de timestamp**: se utiliza la marca temporal de la medida para ordenar y caracterizar temporalmente la muestra.
5. **Extracción de fase y magnitud**: se descartan los 8 campos de metadatos y se conservan los 32 valores de fase y los 32 valores de magnitud.
6. **Control de longitud mínima**: se descartan los ficheros con menos de 100 medidas válidas.
7. **Interpolación temporal**: las secuencias válidas se ajustan a una longitud fija de 120 medidas.
8. **Transformación de fase**: la fase se convierte a representación circular mediante seno y coseno.
9. **Transformación de magnitud**: la magnitud se transforma mediante $\log(1 + x)$ y se normaliza.

10. **Construcción del tensor**: se genera una muestra final de dimensión $120 \times 32 \times 3$.

La normalización de la magnitud se realiza sobre la versión logarítmica de los valores originales:

$$m_i^{log}(t) = \log(1 + m_i(t)). \tag{4.9}$$

Posteriormente se aplica una normalización temporal por componente:

$$\widetilde{m}_i(t) = \frac{m_i^{log}(t) - \mu_i}{\sigma_i + \epsilon}, \tag{4.10}$$

donde μ_i y σ_i representan la media y desviación estándar de la magnitud logarítmica para el índice i dentro de la muestra, y ϵ es una constante pequeña que evita divisiones por cero.

Esta normalización reduce la dependencia del modelo respecto a la potencia absoluta recibida, que puede variar entre sesiones, y favorece que el clasificador aprenda patrones relativos de evolución temporal.

4.10. Particularidades y retos de la adquisición

La adquisición de CSI en banda mmWave presenta varios retos que condicionan tanto el diseño del dataset como la interpretación de los resultados.

El primero es la alta sensibilidad del canal a microvariaciones del entorno. La longitud de onda de aproximadamente 5 mm permite capturar variaciones sutiles del movimiento del robot, pero también hace que el sistema sea sensible a cambios no relacionados con la clase: pequeños desplazamientos de las antenas, variaciones de orientación, presencia de objetos cercanos o cambios de temperatura pueden modificar la distribución de las medidas.

El segundo reto es el desplazamiento de sesión, o *session shift*. Aunque dos capturas correspondan a la misma clase, si se realizan en días, configuraciones o condiciones ambientales ligeramente distintas, la distribución estadística de la CSI puede cambiar de forma significativa. Este fenómeno ha sido identificado de forma recurrente en sistemas de *Wi-Fi sensing*. Por ejemplo, Ding y Wang (2021) y Strohmayer, Sterzinger, Wödlinger, y Kampel (2024) señalan que los cambios de entorno, orientación, calidad de la CSI, propiedades físicas del sujeto u objeto sensado y pequeñas variaciones en el patrón de movimiento pueden

degradar el rendimiento de los modelos entrenados en un dominio y evaluados en otro. Por esta razón, una partición aleatoria puede sobreestimar el rendimiento real al mezclar muestras muy similares entre entrenamiento y prueba, mientras que una evaluación por día de captura mide de forma más exigente la capacidad de generalización del sistema ante condiciones no vistas.

El tercer reto es la ausencia de información Doppler explícita en la representación utilizada por el sistema. En radares Frequency-Modulated Continuous Wave (FMCW), la velocidad radial de un objetivo puede estimarse a partir del desplazamiento Doppler y de la evolución de fase entre chirps consecutivos, dando lugar a representaciones de tipo rango-Doppler ampliamente utilizadas en detección y seguimiento de objetos (Bilik, Longman, Villeval, y Tabrikian, 2020; Li, Du, Liu, Müller, y Waldschmidt, 2020). En cambio, la plataforma empleada en este trabajo no entrega directamente un mapa rango-Doppler ni una estimación explícita de velocidad, sino líneas de medida CSI con 32 valores de fase y 32 valores de magnitud. Por tanto, los comportamientos dinámicos no se clasifican mediante velocidad radial instantánea, sino mediante la evolución temporal de fase y magnitud en ventanas de 120 medidas. Esta limitación justifica el uso de modelos de aprendizaje automático capaces de extraer patrones temporales a partir de secuencias completas.

El cuarto reto es el desequilibrio entre clases. Las clases estáticas, `de_pie` y `tumbado`, tienen un número de muestras mayor que las clases dinámicas, `dos_patas` y `baile`. Este desequilibrio se debe a la mayor facilidad de capturar posturas estáticas de forma repetida, frente a la mayor complejidad de sincronizar y registrar comportamientos dinámicos del robot.

Estos retos no invalidan el dataset, pero sí condicionan la forma en que debe evaluarse el sistema. Por ello, en los capítulos posteriores se distinguen claramente dos niveles de evaluación: rendimiento offline con partición aleatoria estratificada y evaluación de generalización entre sesiones.

4.11. Relación con los modelos de clasificación

El resultado final de este capítulo es un dataset estructurado de muestras temporales de CSI, donde cada ejemplo queda representado por un tensor:

$$X_n \in \mathbb{R}^{120\times 32\times 3}, \qquad y_n \in \{0,1,2,3\}. \tag{4.11}$$

Cada etiqueta y_n corresponde a una de las cuatro clases finales del problema. A partir de esta representación, el Capítulo 5 plantea el problema como una tarea de clasificación supervisada y describe las arquitecturas de aprendizaje automático desarrolladas para resolverlo.

La principal decisión metodológica derivada de este capítulo es que el problema no debe abordarse como una clasificación de medidas aisladas, sino como una clasificación de secuencias temporales. Una única medida CSI puede estar afectada por ruido, fluctuaciones instantáneas o condiciones locales del canal. En cambio, una ventana de 120 medidas permite capturar la evolución temporal asociada a cada comportamiento, especialmente en las clases dinámicas.

En consecuencia, el dataset construido constituye la base sobre la que se evalúan los modelos offline y el pipeline de inferencia online descritos en los siguientes capítulos.

5. Diseño de los Modelos de Clasificación

Este capítulo describe el diseño de los modelos de aprendizaje automático desarrollados para clasificar comportamientos del perro robótico a partir de secuencias temporales de CSI. Mientras que el Capítulo 4 se centró en el sistema experimental y en la construcción del dataset, este capítulo aborda la formulación del problema de clasificación supervisada, la representación de entrada empleada por los modelos, la arquitectura neuronal propuesta, las estrategias de regularización y el procedimiento de entrenamiento.

El enfoque adoptado se ajusta al carácter aplicado del trabajo: el objetivo no es únicamente obtener una buena métrica offline, sino diseñar un modelo que pueda integrarse posteriormente en un pipeline de inferencia online. Por ello, se prioriza una arquitectura con coste computacional moderado, capaz de procesar ventanas temporales de CSI y suficientemente robusta ante variaciones entre sesiones de adquisición.

5.1. Formulación del problema de clasificación supervisada

El problema se formula como una tarea de clasificación supervisada multiclase. Cada muestra del dataset se corresponde con una ventana temporal de medidas CSI y una etiqueta asociada al comportamiento ejecutado por el robot. Tras el preprocesado descrito en el capítulo anterior, cada muestra queda representada como un tensor:

$$X_n \in \mathbb{R}^{T \times F \times C}, \qquad T = 120, \quad F = 32, \quad C = 3, \tag{5.1}$$

donde T es el número de instantes temporales de la ventana, F es el número de valores de canal por medida y C son los tres canales de representación empleados: magnitud normalizada, seno de fase y coseno de fase.

El conjunto de clases finales queda definido como:

$$\mathcal{Y} = \{\texttt{de_pie}, \texttt{tumbado}, \texttt{dos_patas}, \texttt{baile}\}. \tag{5.2}$$

La tarea consiste en aprender una función parametrizada f_θ tal que:

$$f_\theta : \mathbb{R}^{120 \times 32 \times 3} \longrightarrow \mathbb{R}^4, \tag{5.3}$$

donde la salida es un vector de *logits* asociado a las cuatro clases. La predicción final se obtiene aplicando la función *softmax* y seleccionando la clase de mayor probabilidad:

$$\hat{y} = \arg \max_{c \in \{1,\dots,4\}} \frac{\exp(z_c)}{\sum_{j=1}^{4} \exp(z_j)}. \tag{5.4}$$

Este planteamiento asume que la información discriminativa no se encuentra necesariamente en una única medida aislada, sino en la evolución temporal de la fase y la magnitud durante la ventana completa. Esta decisión es especialmente relevante para las clases dinámicas, como `dos_patas` y `baile`, donde el patrón temporal del movimiento es parte esencial de la clase.

5.2. Representación de entrada

Las medidas crudas proporcionadas por la plataforma contienen 32 valores de fase y 32 valores de magnitud. Sin embargo, introducir directamente la fase cuantificada como un valor escalar presenta un problema: la fase es una variable circular. Dos valores cercanos físicamente pueden aparecer alejados numéricamente si se encuentran a ambos lados del salto de 2π.

Por esta razón, la fase se transforma a una representación circular mediante seno y coseno. Dado un valor de fase cuantificado $\phi_i(t)$, se obtiene primero el ángulo:

$$\theta_i(t) = \frac{2\pi\phi_i(t)}{1024}, \tag{5.5}$$

y posteriormente se representa mediante:

$$\left[\sin(\theta_i(t)), \cos(\theta_i(t))\right]. \tag{5.6}$$

La magnitud se transforma mediante $\log(1 + x)$ y se normaliza temporalmente por componente. Para cada índice i:

$$m_i^{log}(t) = \log(1 + m_i(t)), \tag{5.7}$$

$$\widetilde{m}_i(t) = \frac{m_i^{log}(t) - \mu_i}{\sigma_i + \epsilon}, \tag{5.8}$$

donde μ_i y σ_i son, respectivamente, la media y desviación estándar temporal de la magnitud logarítmica para el índice i, y ϵ evita divisiones por cero.

Así, para cada instante temporal t y cada índice i se construye el vector:

$$\mathbf{z}_i(t) = [\widetilde{m}_i(t), \sin(\theta_i(t)), \cos(\theta_i(t))]\,. \tag{5.9}$$

En la implementación con PyTorch, el tensor se reorganiza al formato habitual de entrada de una red convolucional bidimensional:

$$X_n \in \mathbb{R}^{C\times T\times F}, \tag{5.10}$$

de forma que los tres canales actúan como canales de entrada y las dimensiones T y F se procesan como una matriz temporal-canal.

5.3. Partición del dataset y protocolos de evaluación

Para entrenar y evaluar los modelos se definieron dos protocolos complementarios. El primero es una partición aleatoria estratificada, cuyo objetivo es medir la capacidad del modelo para clasificar muestras procedentes de la misma distribución global del dataset. El segundo es una evaluación con separación por día de captura, cuyo objetivo es analizar la generalización ante desplazamiento de sesión.

En la partición aleatoria estratificada se reserva un 15 % de las muestras para validación, un 15 % para prueba y el 70 % restante para entrenamiento. Esta partición mantiene aproximadamente la distribución de clases en los tres subconjuntos. En el dataset final, esta división produce:

$$N_{train} = 1264, \qquad N_{val} = 271, \qquad N_{test} = 271. \tag{5.11}$$

Adicionalmente, se definió una evaluación de tipo *day-holdout*. En este caso, el modelo se entrena con muestras de un día y se evalúa sobre otro día distinto. Esta configuración es más exigente, ya que impide que muestras muy similares de una misma sesión aparezcan simultáneamente en entrenamiento y prueba.

Se consideraron dos direcciones de evaluación:

- Entrenamiento en la sesión del 21/04/2026 y prueba en la sesión del 24/04/2026.
- Entrenamiento en la sesión del 24/04/2026 y prueba en la sesión del 21/04/2026.

La Tabla 5.1 resume los tamaños de los subconjuntos empleados en cada protocolo.

Tabla 5.1: Protocolos de partición empleados para entrenamiento, validación y prueba.

Protocolo	**Entrenamiento**	**Validación**	**Prueba**
Aleatorio estratificado	1264	271	271
Día 21/04/2026 → 24/04/2026	627	111	274
Día 24/04/2026 → 21/04/2026	232	42	738

La partición aleatoria proporciona una estimación útil del rendimiento en condiciones similares a las observadas durante el entrenamiento. Sin embargo, la evaluación por día es más representativa del uso en condiciones no vistas, ya que captura el efecto de variaciones en la distribución de la CSI entre sesiones.

5.4. Arquitecturas consideradas

Durante el diseño del sistema se consideraron varias familias de modelos para abordar la clasificación de secuencias CSI. Esta comparativa es una parte central del trabajo, ya que permite comprobar si la señal contiene información discriminativa suficiente y, al mismo tiempo, justificar la elección del modelo integrado posteriormente en el pipeline de inferencia.

Los resultados cuantitativos de esta comparativa se presentan en el Capítulo 6. En este capítulo se describen las alternativas consideradas y el papel que desempeña cada una dentro del diseño experimental.

5.4.1. SVM con núcleo RBF

Como primera referencia se considera una SVM con núcleo RBF, modelo clásico ampliamente utilizado en problemas de clasificación supervisada con conjuntos de datos de tamaño moderado (Cortes y Vapnik, 1995). En este caso, la secuencia CSI no se introduce como tensor, sino que se transforma en una representación vectorial de características estadísticas y, posteriormente, se reduce mediante Principal Component Analysis (PCA) antes de entrenar el clasificador.

Este modelo cumple una función de línea base clásica. Si una SVM alcanza un rendimiento elevado, ello indica que las clases presentan diferencias estadísticas claras en la

señal. Sin embargo, su principal limitación es que no explota de forma explícita la estructura temporal completa de la ventana ni la organización bidimensional tiempo–índice CSI. Por tanto, aunque puede ofrecer buenas métricas offline, resulta menos adecuada para extender el sistema hacia arquitecturas multitarea o mecanismos de adaptación de dominio.

5.4.2. Perceptrón multicapa

Como línea base neuronal sencilla se incluye también un perceptrón multicapa, o MLP. En este enfoque, el tensor de entrada $X \in \mathbb{R}^{120\times32\times3}$ se aplana en un único vector y se procesa mediante capas densas (Yang y cols., 2023). A diferencia de las redes convolucionales o de atención, el MLP no incorpora ninguna hipótesis estructural sobre la relación entre instantes temporales próximos o entre índices CSI consecutivos.

Su utilidad es servir como comparación intermedia entre los modelos clásicos y las arquitecturas profundas estructuradas. Si el MLP queda por debajo de modelos convolucionales o de atención, ello sugiere que preservar la estructura de la señal resulta beneficioso para la clasificación.

5.4.3. CNN simple

La tercera alternativa considerada es una red convolucional bidimensional simple (Fard Moshiri, Nabati, Shahbazian, y Ghorashi, 2021). Este modelo ya interpreta la entrada como una matriz tiempo–índice CSI con tres canales, pero utiliza una arquitectura más ligera que CsiNet, sin bloques residuales, sin cabeza binaria auxiliar y sin mecanismos de adaptación de dominio.

La CNN simple permite aislar el efecto de introducir convoluciones sobre la representación estructurada de la señal. Su comparación con CsiNet permite evaluar si los bloques residuales, la cabeza auxiliar y las estrategias de regularización aportan una mejora relevante frente a una red convolucional básica.

5.4.4. Modelo basado en atención

También se considera un modelo ligero basado en atención, implementado mediante un codificador Transformer compacto. En este caso, cada instante de la ventana temporal se proyecta a un embedding y el mecanismo de autoatención aprende relaciones entre posiciones temporales de la secuencia (Vaswani y cols., 2017; Zerveas y cols., 2021).

Este modelo es relevante porque los comportamientos dinámicos del robot no afectan necesariamente a toda la ventana temporal de forma uniforme: ciertos instantes pueden ser más informativos que otros. Un mecanismo de atención puede aprender a ponderar esas posiciones. No obstante, este tipo de arquitectura puede requerir más datos y una regularización cuidadosa para evitar sobreajuste, especialmente cuando el número de sesiones de adquisición es limitado.

5.4.5. CsiNet residual

La familia seleccionada como base del modelo principal es la de redes convolucionales residuales. Esta elección responde a cuatro razones. En primer lugar, la entrada del problema puede interpretarse de forma natural como una matriz tiempo–índice CSI con tres canales: magnitud normalizada, seno de fase y coseno de fase. En segundo lugar, las convoluciones bidimensionales permiten capturar patrones locales tanto en la dimensión temporal como en la dimensión de canal. En tercer lugar, las conexiones residuales facilitan el entrenamiento de redes más profundas al mitigar problemas de degradación del gradiente (He y cols., 2016). Finalmente, esta arquitectura permite incorporar de forma sencilla cabezas auxiliares, como la clasificación estático/dinámico o la adaptación adversarial de dominio.

A partir de esta familia se diseña el modelo *CsiNet*, descrito en detalle en la siguiente sección. La comparación experimental del Capítulo 6 muestra que otros modelos pueden alcanzar rendimientos muy competitivos en partición aleatoria, pero CsiNet se mantiene como modelo principal del sistema por su equilibrio entre rendimiento, estructura específica para tensores CSI, extensibilidad y facilidad de integración en inferencia online.

5.5. Modelo propuesto: CsiNet

El modelo desarrollado para la integración final recibe el nombre de *CsiNet*. Se trata de una red convolucional residual diseñada específicamente para procesar tensores de CSI con forma $3 \times 120 \times 32$. La arquitectura incluye una cabeza principal de clasificación multiclase y una cabeza auxiliar binaria para distinguir comportamientos estáticos y dinámicos. Además, se implementa una variante opcional con cabeza adversarial de dominio, utilizada en los experimentos de robustez entre sesiones.

La versión seleccionada como modelo principal para el sistema es *CsiNet baseline-less-aug*: una variante sin cabeza de dominio, con aumento de datos suavizado. Esta configuración se justifica cuantitativamente en el Capítulo 6, donde obtiene el mejor equilibrio entre rendimiento offline y robustez intersesión. La variante adversarial *domain-adv-soft* se conserva como análisis complementario, ya que permite estudiar si una regularización de dominio conservadora mejora la generalización entre días.

La Figura 5.1 muestra la arquitectura general de la familia CsiNet.

La entrada del modelo tiene tres canales: magnitud normalizada, seno de fase y coseno de fase. El primer bloque, denominado *stem*, aplica una convolución bidimensional seguida de normalización por grupos y activación SiLU. A continuación, se emplean tres grupos de bloques residuales. Cada grupo incrementa progresivamente el número de canales y reduce parcialmente la resolución temporal o la dimensión de canal mediante convoluciones con *stride*.

Los anchos de canal utilizados son:

$$(32,\ 48,\ 72,\ 96). \tag{5.12}$$

Tras los bloques convolucionales, se aplica un *adaptive average pooling* hasta una salida espacial 1×1. El vector resultante se proyecta a un espacio latente de 192 dimensiones:

$$\mathbf{h}_n \in \mathbb{R}^{192}. \tag{5.13}$$

Sobre este embedding se construyen las distintas cabezas de salida.

5.5.1. Bloques residuales

Cada bloque residual aplica dos transformaciones convolucionales y suma la entrada del bloque a su salida transformada. De forma abstracta, un bloque residual puede escribirse como:

$$\mathbf{x}_{out} = \sigma\left(\mathcal{F}(\mathbf{x}_{in}; \theta) + \mathcal{S}(\mathbf{x}_{in})\right), \tag{5.14}$$

donde $\mathcal{F}$ representa la rama convolucional principal, $\mathcal{S}$ es la conexión de salto, que puede ser la identidad o una proyección 1×1 cuando cambia la dimensión, y σ es la activación no lineal.

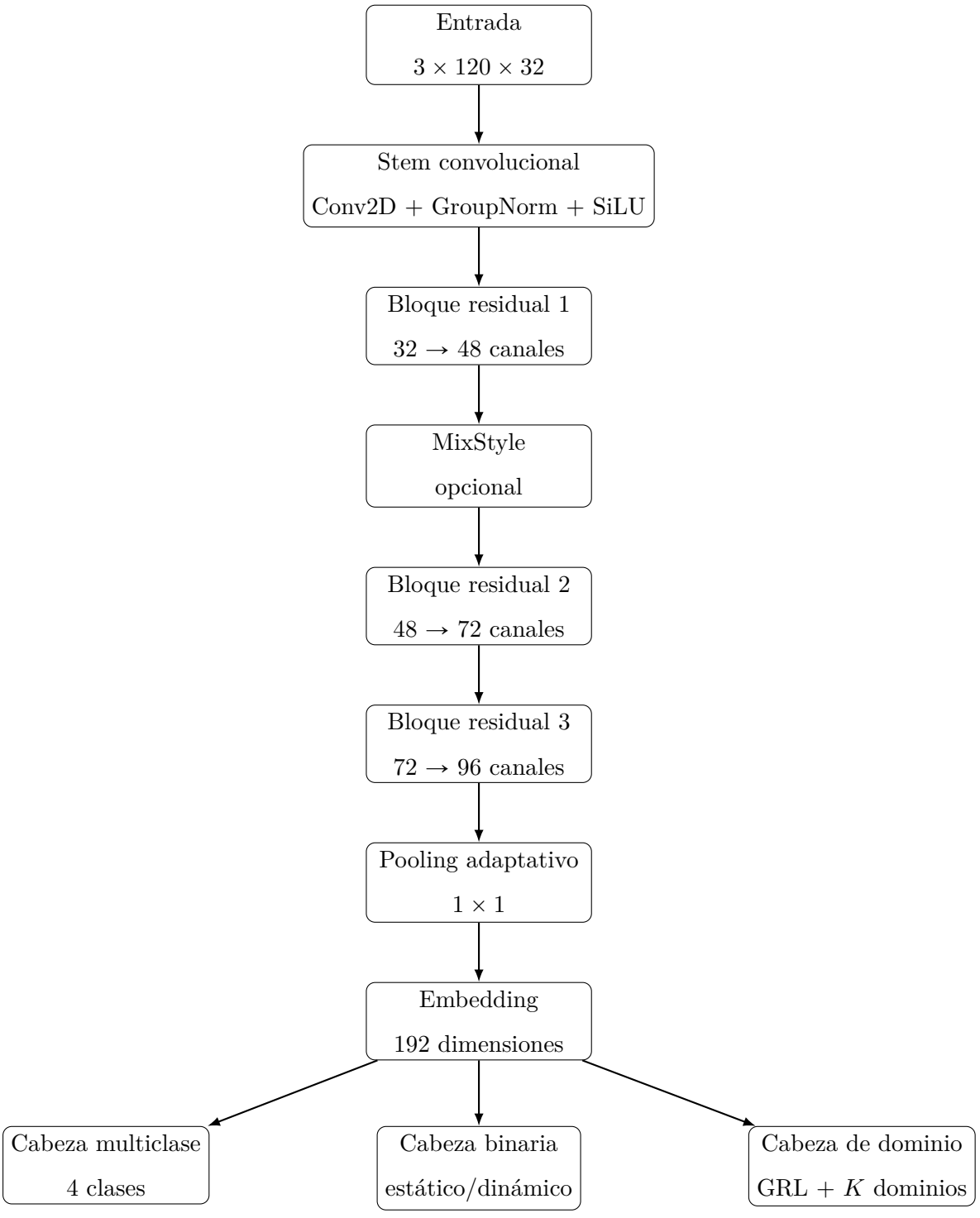


Figura 5.1: Arquitectura general de la familia CsiNet. La cabeza de dominio y MixStyle se activan únicamente en las variantes con adaptación adversarial de dominio; el modelo principal empleado en inferencia usa la rama multiclase y la cabeza binaria auxiliar.

En este trabajo se emplean convoluciones con kernels $(5, 3)$ y $(3, 3)$, lo que permite capturar patrones locales relativamente extendidos en el eje temporal y correlaciones entre valores próximos de la dimensión de canal. La normalización se realiza mediante *Group Normalization*, lo que reduce la dependencia del tamaño de batch y resulta adecuado para datasets de tamaño moderado.

5.5.2. Cabeza principal de clasificación

La cabeza principal toma el embedding $\mathbf{h}_n$ y produce cuatro *logits*, uno por clase:

$$\mathbf{z}_n^{class} = W_2\,\sigma\left(W_1\mathbf{h}_n + \mathbf{b}_1\right) + \mathbf{b}_2, \tag{5.15}$$

donde $\mathbf{z}_n^{class} \in \mathbb{R}^4$. La predicción se obtiene mediante la función *softmax*, como se indicó en la Ecuación 5.4.

Esta cabeza es la salida principal del modelo y la que se utiliza para evaluar la clasificación de las cuatro clases finales.

5.5.3. Cabeza auxiliar estático/dinámico

Además de la clasificación multiclase, se añade una cabeza auxiliar binaria cuyo objetivo es distinguir entre comportamientos estáticos y dinámicos:

$$\mathcal{Y}_{bin} = \{\texttt{estatico}, \texttt{dinamico}\}. \tag{5.16}$$

Las clases `de_pie` y `tumbado` se asignan al grupo estático, mientras que `dos_patas` y `baile` se asignan al grupo dinámico. Esta salida auxiliar introduce una señal de supervisión adicional que fuerza al embedding a separar, al menos parcialmente, patrones de postura frente a patrones de movimiento. En problemas con clases heterogéneas y dataset moderado, este tipo de aprendizaje multitarea puede actuar como regularizador al imponer una estructura semántica adicional sobre la representación latente (Ruder, 2017).

5.5.4. Cabeza adversarial de dominio

La variante adversarial de CsiNet incorpora una cabeza de dominio inspirada en el aprendizaje adversarial de dominio (Ganin y cols., 2016). Su objetivo es reducir la dependencia del embedding respecto a variaciones de sesión o estilo de canal. Aunque se dispone

de información temporal de sesión, esta no captura necesariamente toda la variabilidad física del canal. Por ello, en la variante adversarial se construyen pseudo-dominios mediante agrupamiento no supervisado sobre características de estilo de la señal.

Para ello, se extraen características de estilo a partir de la magnitud logarítmica, incluyendo estadísticas por índice de canal y estadísticas globales. Sobre estas características se aplica *k-means* (MacQueen, 1967), obteniendo K pseudo-dominios. Estos pseudo-dominios no representan clases semánticas del robot, sino agrupaciones de condiciones de canal. En la variante conservadora utilizada para el análisis final se emplea $K = 3$.

La cabeza de dominio se conecta al embedding mediante una capa de inversión de gradiente (*Gradient Reversal Layer*, GRL). Durante el paso hacia delante, esta capa actúa como la identidad:

$$GRL(\mathbf{h}) = \mathbf{h}, \tag{5.17}$$

pero durante la retropropagación invierte el signo del gradiente que llega al extractor de características:

$$\frac{\partial GRL}{\partial \mathbf{h}} = -\lambda_{grl} I. \tag{5.18}$$

Así, la cabeza de dominio aprende a predecir el pseudo-dominio, mientras que el extractor de características recibe una señal opuesta que le empuja a producir embeddings menos informativos respecto al dominio. El objetivo práctico es que la representación conserve información útil para clasificar comportamientos, pero reduzca la dependencia respecto a condiciones específicas de sesión.

En las primeras pruebas se empleó una programación sigmoidal relativamente agresiva para λ_{grl}. Sin embargo, los experimentos finales muestran que la regularización adversarial debe aplicarse de forma conservadora para no eliminar información útil de clase. Por ello, la variante *domain-adv-soft* utiliza un peso de dominio reducido y $\lambda_{max} = 0{,}1$, frente a valores mayores empleados en configuraciones preliminares.

Esta variante no se plantea como sustitución directa del modelo principal, sino como mecanismo experimental para estudiar la robustez frente al desplazamiento de sesión. Como se analiza en el Capítulo 6, la regularización adversarial suave mejora parcialmente algunas métricas de equilibrio entre clases, pero no supera de forma consistente a `baseline_less_aug` en macro-F1.

5.5.5. MixStyle

En algunas variantes con adaptación de dominio se incorpora también MixStyle, una técnica de regularización que mezcla estadísticas de estilo entre muestras de un mismo batch (Zhou y cols., 2021). Sea $\mu(\mathbf{x})$ y $\sigma(\mathbf{x})$ la media y desviación estándar de una activación intermedia. MixStyle normaliza la activación y reconstruye su estilo combinando estadísticas propias con las de otra muestra del batch:

$$\mu_{mix} = \lambda\mu(\mathbf{x}) + (1-\lambda)\mu(\mathbf{x}_{perm}), \tag{5.19}$$

$$\sigma_{mix} = \lambda\sigma(\mathbf{x}) + (1-\lambda)\sigma(\mathbf{x}_{perm}), \tag{5.20}$$

donde λ se muestrea de una distribución beta. En este trabajo MixStyle no forma parte de la configuración final seleccionada, sino de una variante experimental con adaptación de dominio suave. En dicha variante se aplica tras el primer grupo de bloques residuales con una probabilidad reducida, $p = 0{,}05$, para evitar una perturbación excesiva del estilo de canal.

5.6. Funciones de pérdida

El entrenamiento combina hasta tres términos de pérdida: clasificación multiclase, clasificación binaria auxiliar y clasificación de dominio. La pérdida principal es la entropía cruzada multiclase con pesos por clase y suavizado de etiquetas:

$$\mathcal{L}_{class} = -\sum_{c=1}^{4} w_c \, y_c \log(\hat{p}_c), \tag{5.21}$$

donde w_c es el peso asociado a la clase c, y_c es la etiqueta real codificada en formato *one-hot* y $\hat{p}_c$ es la probabilidad predicha por el modelo. Los pesos de clase se calculan de forma inversamente proporcional a la frecuencia de cada clase en el conjunto de entrenamiento:

$$w_c \propto \frac{1}{N_c}, \tag{5.22}$$

y posteriormente se normalizan para que su media sea uno. Esta ponderación compensa parcialmente el desequilibrio entre clases estáticas y dinámicas.

La pérdida binaria auxiliar se define como:

$$\mathcal{L}_{bin} = -\sum_{b=1}^{2} y_b^{bin} \log(\hat{p}_b^{bin}). \tag{5.23}$$

Cuando se activa la cabeza de dominio, se añade una tercera pérdida:

$$\mathcal{L}_{dom} = -\sum_{k=1}^{K} y_k^{dom} \log(\hat{p}_k^{dom}), \qquad K \in \{3, 5\}. \tag{5.24}$$

La función de pérdida total implementada durante el entrenamiento es:

$$\mathcal{L} = \mathcal{L}_{class} + \alpha\mathcal{L}_{bin} + \beta\mathcal{L}_{dom}, \tag{5.25}$$

con $\alpha = 0{,}15$ y $\beta = 0{,}4$ en los experimentos con adaptación de dominio. En la variante sin adaptación de dominio se anula el término $\mathcal{L}_{dom}$. Debido a la capa de inversión de gradiente, el término de dominio actúa de forma adversarial sobre el extractor de características, aunque se implemente como una entropía cruzada convencional en la cabeza de dominio.

5.7. Aumento de datos

Dado que el dataset contiene un número limitado de sesiones y que la señal mmWave es altamente sensible a variaciones de entorno, se aplican técnicas de aumento de datos durante el entrenamiento. Estas transformaciones se aplican únicamente al conjunto de entrenamiento, nunca a validación ni prueba (Iwana y Uchida, 2021).

La probabilidad global de aplicar aumento de datos a una muestra es $p = 0{,}85$. Si se activa, se aplican de forma probabilística las transformaciones de la Tabla 5.2.

Tabla 5.2: Técnicas de aumento de datos empleadas durante el entrenamiento.

Transformación	**Probabilidad**	**Descripción**
Rotación global de fase	0,70	Aplica una rotación angular común a los canales seno/coseno para simular desplazamientos de fase.
Ruido en magnitud	0,60	Añade ruido gaussiano de desviación estándar 0.05 sobre el canal de magnitud normalizada.
Jitter de fase	0,20	Añade ruido angular local sobre la fase reconstruida a partir de seno y coseno.
Enmascaramiento temporal	0,50	Anula un segmento temporal aleatorio de entre 6 y 20 muestras.
Enmascaramiento de canal	0,50	Anula un bloque aleatorio de entre 2 y 6 índices de canal.
Repetición de frames	0,35	Sustituye entre 1 y 6 frames por el frame anterior, simulando pérdidas puntuales.
Recorte y redimensionado temporal	0,15	Recorta una subsecuencia temporal y la interpola de nuevo a longitud 120.

Estas transformaciones buscan mejorar la robustez del modelo ante fluctuaciones realistas de la adquisición: cambios de fase, variaciones de potencia, pérdidas de medidas y pequeñas diferencias de velocidad en la ejecución de los movimientos.

5.8. Configuración de entrenamiento

La implementación se realizó en Python utilizando PyTorch (Paszke y cols., 2019b). Los modelos neuronales se entrenan con AdamW, decaimiento de pesos, *scheduler* cosenoidal y parada temprana basada en macro-*F1* de validación. La Tabla 5.3 resume las configuraciones más relevantes empleadas en los experimentos finales.

Tabla 5.3: Configuraciones principales de entrenamiento empleadas en los experimentos finales.

Parámetro	**CsiNet baseline-less-aug**	**CsiNet domain-adv-soft**
Framework	PyTorch	PyTorch
Épocas máximas	40	40
Tamaño de batch	64	64
Optimizador	AdamW	AdamW
Tasa de aprendizaje inicial	10^{-3}	10^{-3}
Decaimiento de pesos	10^{-4}	10^{-4}
Scheduler	Cosine Annealing	Cosine Annealing
Paciencia de parada temprana	10 épocas	10 épocas
Métrica de selección	Macro-*F1* validación	Macro-*F1* validación
Suavizado de etiquetas	0,03	0,03
Peso pérdida binaria α	0,15	0,15
Peso pérdida dominio β	–	0,10
$\lambda_{\text{máx}}$ de GRL	–	0,10
Número de pseudo-dominios	–	3
Probabilidad MixStyle	–	0,00
Probabilidad global de aumento	0,45	0,85
Recorte de gradiente	Norma máxima 5,0	Norma máxima 5,0

Durante el entrenamiento, el mejor punto de cada modelo se selecciona según el valor de macro-*F1* en el conjunto de validación. Posteriormente, la selección del modelo final para el pipeline online se realiza en el Capítulo 6, comparando los resultados obtenidos por las distintas arquitecturas y configuraciones sobre los conjuntos de prueba. Esta elección es más adecuada que la exactitud global en presencia de cierto desequilibrio entre clases, ya que da el mismo peso a cada clase independientemente de su frecuencia.

5.9. Variantes experimentales

Sobre la arquitectura CsiNet se evaluaron varias configuraciones. El objetivo no era únicamente maximizar la métrica en una partición aleatoria, sino analizar qué decisiones

de entrenamiento afectaban a la robustez del modelo frente a cambios de sesión.

baseline-original-like. Configuración de referencia sin cabeza de dominio, con aumento de datos global $p = 0{,}85$ y parámetros equivalentes a los entrenamientos preliminares del sistema.

baseline-long. Variante sin cabeza de dominio con más épocas máximas y mayor paciencia, utilizada para comprobar si prolongar el entrenamiento mejoraba el rendimiento o inducía sobreajuste.

baseline-less-aug. Variante sin cabeza de dominio y con aumento de datos suavizado ($p = 0{,}45$). Esta configuración reduce la intensidad de las perturbaciones sintéticas y se utiliza como candidata principal para el sistema final.

domain-adv-soft. Variante con cabeza adversarial de dominio, $K = 3$ pseudo-dominios, $\beta = 0{,}10$ y $\lambda_{max} = 0{,}10$. Su objetivo es evaluar si una regularización de dominio conservadora mejora la generalización entre sesiones sin degradar en exceso la clasificación principal.

domain-adv-mixstyle-soft. Variante adversarial suave que añade MixStyle con probabilidad reducida. Permite analizar si mezclar estilos de canal aporta robustez adicional.

domain-adv-delayed. Variante adversarial con activación más conservadora o retardada de la inversión de gradiente, diseñada para evitar que la señal adversarial domine las primeras etapas del entrenamiento.

La Tabla 5.4 resume las diferencias principales entre la configuración seleccionada como modelo principal y la variante adversarial conservadora.

La comparación de estas variantes se presenta en el Capítulo 6. Como se discutirá allí, la configuración *baseline-less-aug* ofrece el mejor equilibrio global para el sistema final, mientras que *domain-adv-soft* resulta útil para analizar el efecto de la regularización de dominio sobre la evaluación intersesión.

5.10. Métricas de evaluación

Aunque los resultados cuantitativos se presentan en el Capítulo 6, las métricas se definen aquí por formar parte del diseño experimental.

Tabla 5.4: Diferencias principales entre las dos configuraciones CsiNet más relevantes para el análisis final.

Componente	**baseline-less-aug**	**domain-adv-soft**
Cabeza multiclase	Sí	Sí
Cabeza binaria estático/dinámico	Sí	Sí
Cabeza de dominio	No	Sí
GRL	No	Sí
MixStyle	No	No
Pseudo-dominios	–	3
β dominio	–	0,10
λ_{max} GRL	–	0,10
Probabilidad global de aumento	0,45	0,85

La exactitud global mide la proporción de muestras correctamente clasificadas:

$$Accuracy = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}(\hat{y}_n = y_n). \tag{5.26}$$

La precisión y el recall por clase se definen como:

$$Precision_c = \frac{TP_c}{TP_c + FP_c}, \qquad Recall_c = \frac{TP_c}{TP_c + FN_c}. \tag{5.27}$$

A partir de ellos se calcula el *F1-score* por clase:

$$F1_c = 2\frac{Precision_c \cdot Recall_c}{Precision_c + Recall_c}. \tag{5.28}$$

La métrica principal de comparación es el macro-*F1*:

$$MacroF1 = \frac{1}{C}\sum_{c=1}^{C} F1_c, \qquad C = 4. \tag{5.29}$$

También se emplea la exactitud balanceada:

$$BalancedAccuracy = \frac{1}{C}\sum_{c=1}^{C} Recall_c. \tag{5.30}$$

Estas métricas son necesarias porque el dataset no está perfectamente equilibrado. Las clases `de_pie` y `tumbado` tienen más muestras que las clases dinámicas, por lo que la exactitud global puede ocultar errores relevantes en clases minoritarias.

5.11. Pipeline de inferencia online

Además del entrenamiento y evaluación offline de los modelos, durante el desarrollo del trabajo se diseñó e implementó un pipeline de inferencia online orientado a validar la viabilidad práctica del sistema en tiempo real.

El objetivo de este pipeline consiste en transformar el flujo continuo de medidas CSI capturado por las antenas mmWave en predicciones temporales de comportamiento del robot, integrando posteriormente dichas predicciones con un gemelo digital desarrollado en Isaac Sim.

A diferencia del escenario offline, donde las muestras se encuentran previamente segmentadas y etiquetadas, el pipeline online debe operar sobre un flujo continuo de datos, gestionando problemas adicionales como sincronización temporal, acumulación de ventanas, estabilización de predicciones y actualización continua del estado del sistema.

La Figura 5.2 muestra el flujo general del sistema de inferencia online diseñado.

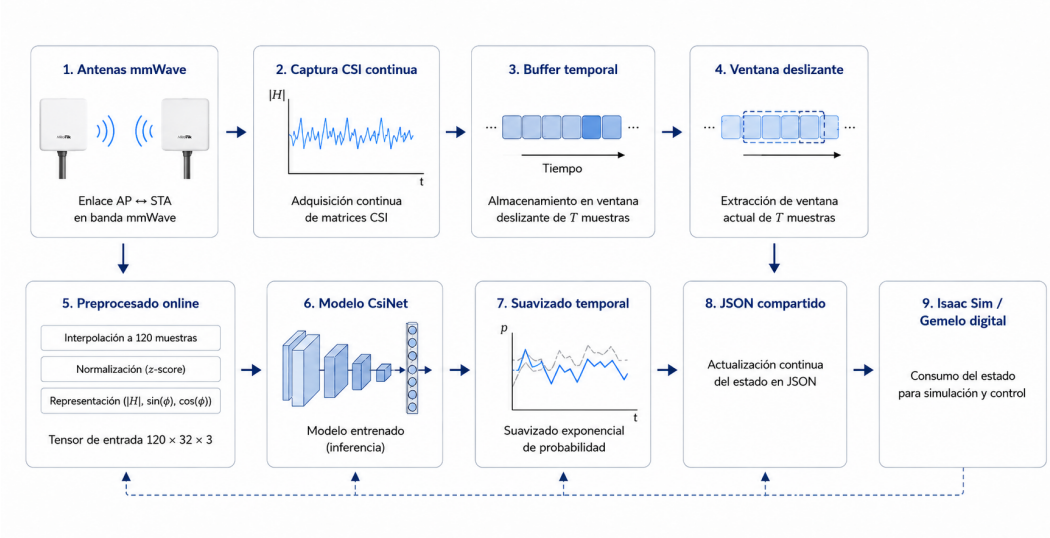


Figura 5.2: Flujo general del pipeline de inferencia online. Imagen generada con Inteligencia Artificial

5.11.1. Captura continua de medidas CSI

El sistema online parte de la captura continua de medidas CSI generadas por el enlace inalámbrico entre las dos antenas MikroTik configuradas para extracción de información del canal.

Durante la ejecución del sistema, las antenas transmiten y reciben tráfico de manera continua mientras un script de captura almacena las medidas generadas en un fichero compartido. Cada línea contiene la información *raw* asociada a una medida CSI, incluyendo los distintos valores complejos correspondientes al perfil de canal estimado.

A diferencia del pipeline offline, donde cada muestra ya se encontraba almacenada en un fichero independiente, en el sistema online las medidas se generan de forma continua y deben procesarse dinámicamente.

5.11.2. Buffer temporal y ventana deslizante

Dado que el modelo CsiNet no opera sobre medidas individuales sino sobre secuencias temporales completas, el sistema mantiene un buffer dinámico que almacena las últimas medidas disponibles.

El modelo principal fue entrenado utilizando ventanas temporales de 120 medidas consecutivas. Por tanto, el sistema online debe acumular continuamente nuevas muestras hasta completar una ventana válida antes de ejecutar la inferencia.

Una vez completada la primera ventana, el sistema adopta una estrategia de ventana deslizante (*sliding window*). Cada nueva medida recibida provoca: la eliminación de la muestra más antigua del buffer, la incorporación de la nueva medida y la generación de una nueva ventana temporal para inferencia.

Este mecanismo permite obtener predicciones continuas manteniendo simultáneamente contexto temporal suficiente para la clasificación.

No obstante, el uso de ventanas largas introduce una latencia inherente, ya que el modelo necesita observar un número elevado de medidas antes de generar una predicción estable.

5.11.3. Preprocesado online

Una vez generada la ventana temporal correspondiente, el sistema aplica el mismo pipeline de preprocesado utilizado durante el entrenamiento offline.

Las medidas CSI se reorganizan en forma tensorial preservando la dimensión temporal, estructura del canal y representación compleja mediante canales separados.

Posteriormente, el tensor resultante se normaliza utilizando los mismos criterios empleados durante entrenamiento, garantizando compatibilidad entre inferencia online y distribución de entrenamiento.

Finalmente, el tensor se convierte al formato de entrada esperado por el modelo PyTorch antes de ejecutar la inferencia.

5.11.4. Inferencia mediante CsiNet

El modelo seleccionado para inferencia online corresponde a la variante *baseline-less-aug* de CsiNet, elegida tras el proceso de evaluación experimental descrito en el Capítulo 6.

Esta configuración fue seleccionada debido a que ofrecía el mejor equilibrio entre precisión offline, estabilidad, robustez intersesión y viabilidad para despliegue online.

Durante la ejecución online, cada ventana temporal procesada se introduce en el modelo, obteniendo como salida un vector de probabilidades asociado a las distintas clases de comportamiento siendo la clase con mayor probabilidad se considera la predicción instantánea del sistema.

5.11.5. Estabilización temporal de predicciones

Uno de los principales problemas observados durante la inferencia online es la fluctuación temporal de las predicciones.

Debido a la elevada sensibilidad del canal inalámbrico, pequeñas perturbaciones o variaciones instantáneas pueden provocar cambios abruptos entre clases consecutivas, incluso cuando el comportamiento real del robot permanece estable.

Para reducir este efecto se incorporó un mecanismo de estabilización temporal basado en suavizado de predicciones.

El sistema mantiene un historial reciente de inferencias y aplica mecanismos de consenso temporal antes de actualizar el estado final del sistema. Entre las estrategias consideradas se incluyen votación mayoritaria sobre varias inferencias consecutivas, suavizado temporal de probabilidades e histéresis de cambio de estado.

Este tipo de técnicas resulta especialmente importante en sistemas de *wireless sensing*, donde las predicciones instantáneas suelen presentar un mayor nivel de ruido que en sistemas basados en visión artificial.

5.11.6. Modelo auxiliar de baja latencia

Durante las pruebas realizadas se observó que las ventanas de 120 medidas proporcionaban una buena capacidad de clasificación, pero introducían una latencia relativamente elevada para determinados escenarios dinámicos.

Por este motivo se exploró también el uso de modelos auxiliares basados en ventanas temporales más cortas, orientados principalmente a detección rápida de cambios de estado.

La idea general consiste en combinar un modelo largo y estable para clasificación robusta junto con un modelo corto de menor latencia para detección rápida.

Aunque esta línea quedó en fase experimental, los resultados preliminares sugieren que una arquitectura jerárquica de inferencia podría mejorar significativamente la capacidad de respuesta temporal del sistema.

5.11.7. Comunicación mediante JSON compartido

Una vez estabilizada la predicción final, el sistema genera un fichero JSON compartido utilizado como mecanismo de comunicación entre el pipeline de inferencia y el gemelo digital.

Este fichero contiene información como:

- clase predicha,
- probabilidad asociada,
- timestamp de actualización,
- estado actual del sistema.

Un ejemplo simplificado de la estructura utilizada se muestra en el Listado 5.1.

Listing 5.1: Ejemplo simplificado del JSON de intercambio online.

```
{
    "predicted_class": "dos_patas",
    "confidence": 0.91,
    "timestamp": 1715189200
}
```

Este mecanismo desacopla completamente el sistema de inferencia del entorno de visualización, permitiendo que distintos módulos consuman el estado del sistema de forma independiente.

5.11.8. Integración con el gemelo digital

Como demostración complementaria del sistema desarrollado, el pipeline online se integró con un gemelo digital del robot implementado en Isaac Sim[1].

El objetivo principal de esta integración consiste en representar visualmente el estado inferido del robot a partir de las perturbaciones detectadas sobre el canal inalámbrico.

Durante la ejecución del sistema, Isaac Sim monitoriza continuamente el fichero JSON generado por el pipeline online y actualiza dinámicamente el comportamiento representado en el entorno virtual.

De este modo, el gemelo digital reproduce de forma aproximada el comportamiento inferido por el clasificador, permitiendo visualizar en tiempo real las predicciones obtenidas a partir de las medidas CSI.

Aunque esta integración se planteó principalmente como una validación funcional y una demostración tecnológica, el resultado obtenido abre la puerta a utilizar el gemelo digital como una plataforma común de integración de múltiples fuentes de sensado. En este sentido, la arquitectura no tendría por qué limitarse exclusivamente a medidas CSI en banda mmWave, sino que podría ampliarse para incorporar información procedente de otras tecnologías, como sensado basado en 5G, cámaras con detección de objetos mediante YOLO (Redmon, Divvala, Girshick, y Farhadi, 2016), sensores LiDAR, unidades inerciales o cualquier otra fuente capaz de proporcionar información semántica o espacial sobre el entorno. De esta forma, el gemelo digital podría actuar como una capa de fusión y visualización donde diferentes mecanismos de percepción alimentan una representación virtual sincronizada del sistema físico.

[1]Isaac Sim es una plataforma de simulación robótica desarrollada por NVIDIA sobre Omniverse, orientada a la creación, entrenamiento, prueba y validación de robots en entornos virtuales. Véase: `https://developer.nvidia.com/isaac/sim`.

6. Evaluación experimental y resultados

6.1. Introducción

En este capítulo se presentan los resultados experimentales obtenidos durante el desarrollo y evaluación del sistema propuesto para la clasificación de comportamientos de un robot cuadrúpedo mediante medidas CSI en banda mmWave. El objetivo principal de esta evaluación es determinar si las perturbaciones introducidas por el robot sobre el canal inalámbrico contienen información suficientemente discriminativa para identificar sus comportamientos, así como analizar la robustez de los modelos ante cambios de sesión y variaciones en las condiciones de captura.

La evaluación se organiza en tres niveles. En primer lugar, se comparan distintas familias de modelos sobre una partición aleatoria estratificada del dataset, con el fin de estimar la capacidad discriminativa de las señales CSI cuando entrenamiento y prueba comparten condiciones similares. En segundo lugar, se estudian distintas variantes del modelo CsiNet, modificando la intensidad del aumento de datos, la duración del entrenamiento y la presencia de mecanismos adversariales de dominio. En tercer lugar, se analiza la generalización intersesión, entrenando en un día de captura y evaluando en otro.

Es importante destacar que los resultados presentados en este capítulo no buscan únicamente maximizar una métrica de clasificación, sino caracterizar el comportamiento real del sistema. En particular, se distingue entre el rendimiento obtenido en condiciones controladas y la capacidad de generalización ante cambios de dominio, ya que las señales mmWave son altamente sensibles a pequeñas variaciones físicas del entorno.

6.2. Métricas de evaluación

Para evaluar el rendimiento de los modelos se utilizaron tres métricas principales: exactitud o *accuracy*, macro-F1 y *balanced accuracy*. Estas métricas permiten analizar el comportamiento de los modelos desde perspectivas complementarias.

La exactitud mide la proporción de muestras correctamente clasificadas:

$$Accuracy = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(\hat{y}_i = y_i), \tag{6.1}$$

donde N es el número total de muestras, y_i es la etiqueta real de la muestra i, $\hat{y}_i$ es la clase predicha por el modelo y $\mathbb{1}(\cdot)$ es la función indicadora.

Aunque la exactitud proporciona una medida global intuitiva, no resulta suficiente cuando el dataset presenta desbalance entre clases. En este trabajo, las clases estáticas tienen más muestras que las clases dinámicas, por lo que un modelo podría obtener una exactitud alta clasificando correctamente las clases mayoritarias y fallando en las minoritarias. Por este motivo, se utiliza también el macro-F1:

$$MacroF1 = \frac{1}{C} \sum_{c=1}^{C} F1_c, \tag{6.2}$$

donde C es el número de clases y $F1_c$ es la métrica F1 calculada de forma independiente para la clase c:

$$F1_c = 2 \cdot \frac{Precision_c \cdot Recall_c}{Precision_c + Recall_c}. \tag{6.3}$$

La tercera métrica empleada es la *balanced accuracy*, definida como la media del *recall* por clase:

$$BalancedAccuracy = \frac{1}{C} \sum_{c=1}^{C} Recall_c. \tag{6.4}$$

Esta métrica es especialmente útil en el análisis intersesión, ya que permite identificar si el modelo mantiene un comportamiento equilibrado entre clases o si tiende a concentrar sus aciertos en un subconjunto reducido de comportamientos.

Además de estas métricas agregadas, se emplean matrices de confusión para estudiar el patrón de errores. Las matrices permiten observar si las confusiones se producen principalmente entre clases estáticas, entre clases dinámicas o entre ambos grupos.

6.3. Protocolos experimentales

La evaluación se realizó sobre tres protocolos experimentales principales. El primero corresponde a una partición aleatoria estratificada, donde las muestras se dividen en entrenamiento, validación y prueba manteniendo la proporción relativa de clases. Este protocolo

permite evaluar si las representaciones CSI contienen información discriminativa cuando las condiciones de adquisición están mezcladas entre los distintos subconjuntos.

Los otros dos protocolos evalúan la generalización entre sesiones. En ellos, el modelo se entrena con muestras de una sesión temporal y se evalúa sobre otra sesión distinta. Estos protocolos son más exigentes, ya que el conjunto de prueba procede de una distribución de canal diferente a la observada durante el entrenamiento.

La Tabla 6.1 resume los tres protocolos utilizados.

Tabla 6.1: Protocolos experimentales utilizados en la evaluación.

Protocolo	**Entrenamiento**	**Validación**	**Prueba**
Aleatorio estratificado	1264 muestras	271 muestras	271 muestras
21/04/2026 → 24/04/2026	Sesión 21/04/2026	Subconjunto de 21/04/2026	Sesión 24/04/2026
24/04/2026 → 21/04/2026	Sesión 24/04/2026	Subconjunto de 24/04/2026	Sesión 21/04/2026

La sesión del 05/05/2026 contiene únicamente clases estáticas y, por tanto, no se utiliza como sesión completa de prueba cross-day para el problema de cuatro clases. No obstante, su existencia refuerza la observación de que las capturas se realizaron en condiciones temporales diferenciadas y que la variabilidad entre sesiones es un aspecto central del trabajo.

6.4. Comparativa entre arquitecturas

Como primera etapa experimental se compararon distintas familias de modelos con el objetivo de analizar qué tipo de arquitectura resulta más adecuada para explotar la información contenida en las secuencias CSI. Los modelos evaluados fueron:

- SVM con núcleo RBF.
- Perceptrón multicapa (MLP).
- Red convolucional simple.
- Transformer reducido.

- Arquitectura CsiNet propuesta.

Todos los modelos se evaluaron inicialmente sobre la partición aleatoria estratificada. La Tabla 6.2 resume los resultados obtenidos.

Tabla 6.2: Comparativa de modelos sobre partición aleatoria estratificada.

Modelo	**Accuracy (%)**	**Macro-F1 (%)**	**Balanced Accuracy (%)**
SVM-RBF	**98,52**	**98,47**	**97,93**
Transformer tiny	**98,52**	97,41	97,37
CsiNet baseline-less-aug	97,79	96,34	96,32
MLP	87,82	85,62	85,53
CNN simple	79,34	78,66	81,05

Los resultados muestran que las secuencias CSI contienen información altamente discriminativa para la clasificación de comportamientos del robot cuando entrenamiento y prueba comparten condiciones de captura similares. Tanto el modelo SVM-RBF como el Transformer reducido alcanzan resultados superiores al 97 % de macro-F1, lo que indica que los patrones inducidos por los distintos comportamientos son claramente separables en la partición aleatoria.

El rendimiento del SVM-RBF es especialmente relevante. Aunque se trata de un modelo clásico, su resultado sugiere que las características estadísticas extraídas de las ventanas CSI ya contienen una separación significativa entre clases. Esta observación es importante porque confirma que el problema no depende exclusivamente de arquitecturas profundas complejas, sino que la propia señal presenta una estructura discriminativa fuerte.

El Transformer reducido obtiene también un rendimiento muy alto. Este resultado es coherente con su capacidad para modelar dependencias temporales dentro de la ventana de 120 medidas. Sin embargo, los modelos basados en atención suelen requerir más datos y una regularización cuidadosa para evitar sobreajuste cuando el número de sesiones es limitado.

El modelo CsiNet alcanza un rendimiento ligeramente inferior al SVM-RBF y al Transformer reducido en partición aleatoria, pero mantiene valores muy elevados: 97,79 % de accuracy, 96,34 % de macro-F1 y 96,32 % de balanced accuracy. A pesar de no ser el mejor modelo absoluto en esta prueba offline, se selecciona como modelo principal del sistema por

tres razones. En primer lugar, su arquitectura está diseñada específicamente para procesar tensores CSI con estructura temporal y de canal. En segundo lugar, incorpora una cabeza auxiliar estático/dinámico y permite extenderse de forma natural hacia mecanismos de adaptación de dominio. En tercer lugar, su estructura convolucional residual resulta adecuada para su integración en un pipeline de inferencia online.

Para complementar las métricas agregadas, la Figura 6.1 muestra las matrices de confusión de los tres modelos con mejor rendimiento en la partición aleatoria: SVM-RBF, Transformer reducido y CsiNet baseline-less-aug. Las matrices permiten comprobar si los modelos aciertan de forma equilibrada en todas las clases o si concentran los errores en comportamientos concretos.

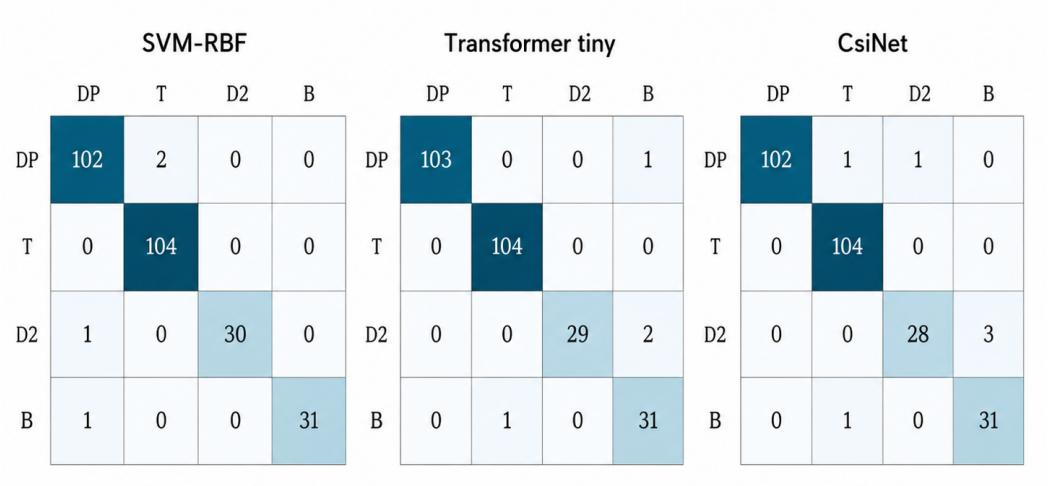


Figura 6.1: Matrices de confusión de los modelos con mejor rendimiento en partición aleatoria estratificada. DP: `de_pie`; T: `tumbado`; D2: `dos_patas`; B: `baile`.

En la partición aleatoria los tres modelos presentan un comportamiento muy estable. La clase `tumbado` es prácticamente perfecta en los tres casos, mientras que las clases dinámicas `dos_patas` y `baile` presentan algunas confusiones entre sí. Esta confusión es razonable, ya que ambas clases generan patrones temporales dinámicos sobre el canal, en contraste con las posturas estáticas.

En CsiNet, los errores se concentran principalmente en la clase `dos_patas`, donde 3 muestras se clasifican como `baile`. Esto indica que la red aprende de forma clara la separación estático/dinámico, pero dentro del subconjunto dinámico algunas secuencias presentan firmas de canal similares.

Por otro lado, las arquitecturas más simples, como el MLP y la CNN básica, presentan un rendimiento significativamente inferior. El MLP alcanza un macro-F1 del 85.62 %, mientras que la CNN simple queda por debajo del 80 %. Este resultado indica que no basta con introducir el tensor completo en una red neuronal genérica: la arquitectura debe ser capaz de explotar adecuadamente la estructura temporal y la correlación entre los índices de canal.

6.5. Análisis de variantes de CsiNet

Tras la comparación inicial entre arquitecturas, se realizó un análisis específico de distintas variantes de CsiNet. El objetivo de este análisis fue identificar una configuración que ofreciera un buen equilibrio entre rendimiento offline, estabilidad del entrenamiento y capacidad de generalización intersesión.

Se evaluaron las siguientes configuraciones:

- `baseline_original_like`: configuración base similar a la empleada inicialmente.
- `baseline_long`: variante con mayor número de épocas y paciencia de parada temprana.
- `baseline_less_aug`: variante con aumento de datos más suave.
- `domain_adv_soft`: variante con adaptación adversarial de dominio conservadora.
- `domain_adv_mixstyle_soft`: variante adversarial suave con MixStyle.
- `domain_adv_delayed`: variante adversarial con activación retardada y peso reducido.

La Tabla 6.3 muestra los resultados de estas configuraciones sobre la partición aleatoria estratificada.

Tabla 6.3: Resultados de variantes CsiNet sobre partición aleatoria estratificada.

Variante	**Acc. (%)**	**Macro-F1 (%)**	**Balanced Acc. (%)**	**Época**
`baseline_less_aug`	**97,79**	**96,34**	**96,32**	38
`baseline_original`	97,05	95,88	95,86	37
`baseline_long`	96,31	94,22	93,73	63
`domain_adv_mixstyle`	94,46	92,61	92,53	36
`domain_adv_delayed`	94,10	92,57	93,37	35
`domain_adv_soft`	94,10	91,87	92,88	33

La variante `baseline_less_aug` obtiene el mejor rendimiento global en partición aleatoria. Este resultado indica que reducir la intensidad del aumento de datos mejora la capacidad de ajuste del modelo sin comprometer su regularización. En este problema, un uso de técnicas de aumento de datos demasiado agresivo puede introducir perturbaciones artificiales alejadas de la distribución física real de la señal CSI. Por tanto, no todas las transformaciones que aumentan la diversidad aparente del dataset son necesariamente beneficiosas.

La variante `baseline_long`, con mayor número de épocas, no mejora el rendimiento. Aunque alcanza una época óptima más tardía, su macro-F1 es inferior al de las configuraciones más cortas. Esto sugiere que prolongar el entrenamiento no resuelve el problema y puede favorecer cierto sobreajuste sobre patrones específicos del conjunto de entrenamiento.

Las variantes con adaptación adversarial presentan un rendimiento inferior en partición aleatoria. Este comportamiento era esperable en parte, ya que la adaptación de dominio no está orientada a maximizar el rendimiento cuando entrenamiento y prueba proceden de distribuciones mezcladas, sino a reducir la dependencia del embedding respecto al dominio de adquisición. Sin embargo, los resultados muestran que el peso adversarial debe ajustarse con mucho cuidado: configuraciones agresivas pueden eliminar información útil para la tarea principal.

Desde el punto de vista de selección del modelo, la variante `baseline_less_aug` se adopta como configuración principal para el sistema. Esta decisión no se basa únicamente en su mejor rendimiento sobre la partición aleatoria, sino también en su comportamiento más estable en los protocolos intersesión presentados en la siguiente sección.

6.6. Evaluación de generalización intersesión

La evaluación intersesión constituye una de las partes más relevantes del trabajo. A diferencia de la partición aleatoria, donde muestras similares pueden aparecer repartidas entre entrenamiento y prueba, en este protocolo se fuerza al modelo a clasificar datos procedentes de una sesión distinta a la utilizada para entrenar.

Este escenario es más exigente y más representativo de un despliegue real, ya que un sistema de *wireless sensing* no debería depender de que las condiciones del canal sean exactamente iguales a las observadas durante el entrenamiento.

La Tabla 6.4 muestra los resultados obtenidos por las principales variantes de CsiNet en los dos sentidos de evaluación intersesión.

Tabla 6.4: Evaluación intersesión de variantes CsiNet.

Protocolo	**Variante**	**Acc. (%)**	**Macro-F1 (%)**	**Bal. Acc. (%)**
21/04 → 24/04	`baseline_less_aug`	**52,92**	**51,36**	62,05
21/04 → 24/04	`domain_adv_soft`	50,73	51,36	**64,19**
21/04 → 24/04	`domain_adv_mixstyle`	48,91	49,02	60,89
21/04 → 24/04	`baseline_original`	46,35	46,12	58,36
21/04 → 24/04	`baseline_long`	50,00	44,77	56,94
21/04 → 24/04	`domain_adv_delayed`	40,88	44,10	56,89
24/04 → 21/04	`baseline_less_aug`	62,06	**62,66**	62,49
24/04 → 21/04	`baseline_original`	**63,28**	60,15	**64,04**
24/04 → 21/04	`domain_adv_soft`	57,86	58,75	58,58
24/04 → 21/04	`domain_adv_delayed`	59,08	58,71	59,87
24/04 → 21/04	`domain_adv_mixstyle`	56,91	56,33	57,81
24/04 → 21/04	`baseline_long`	57,99	50,88	58,50

La caída respecto al rendimiento en partición aleatoria es notable. Mientras que CsiNet alcanza un macro-F1 del 96,34 % en partición aleatoria, en los protocolos intersesión el valor se reduce hasta aproximadamente el 51–63 %. Este comportamiento confirma que el principal reto del sistema no es la capacidad de clasificación en condiciones controladas, sino la robustez frente a cambios de dominio.

El sentido 21/04 → 24/04 resulta más difícil que el sentido inverso. Esto puede explicarse por la menor cantidad de datos disponibles en la sesión del 24/04/2026 y por diferencias en la distribución de clases. Además, las condiciones del canal en la sesión de prueba pueden diferir de forma significativa respecto a la sesión de entrenamiento, provocando que patrones aprendidos como discriminativos en un día dejen de serlo en otro.

La variante `baseline_less_aug` obtiene el mejor macro-F1 en ambos sentidos de evaluación, lo que refuerza su selección como configuración principal. La variante `domain_adv_soft` obtiene la mejor *balanced accuracy* en el protocolo 21/04 → 24/04, lo que indica que la adaptación adversarial suave puede ayudar parcialmente a equilibrar el comportamiento entre clases. Sin embargo, no mejora de forma clara el macro-F1 global y empeora en el sentido inverso, por lo que no se selecciona como modelo principal.

La Figura 6.2 muestra las matrices de confusión de los principales experimentos intersesión. En este caso, las matrices son especialmente útiles porque permiten observar cómo cambia el patrón de errores cuando el modelo se enfrenta a una sesión no vista.

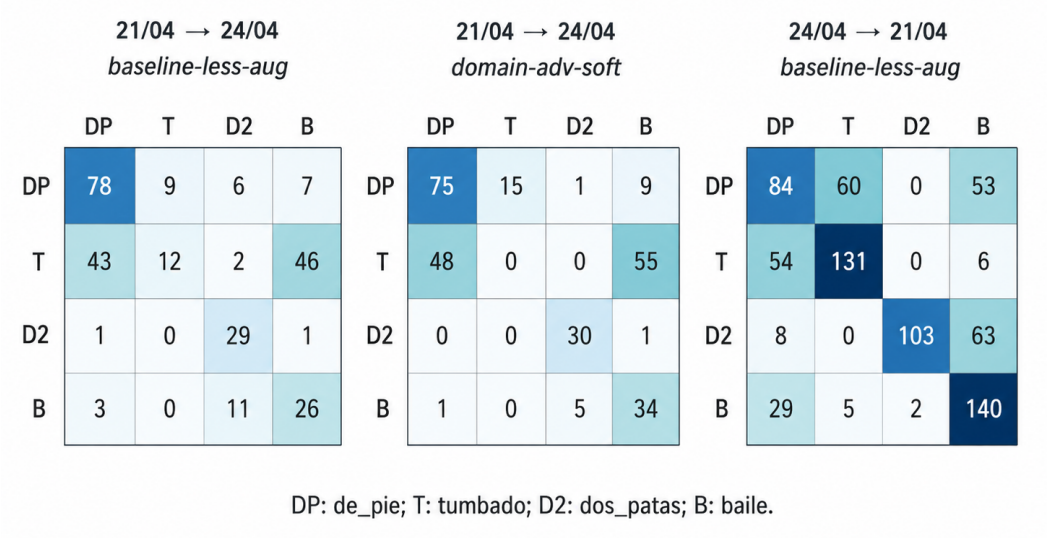


Figura 6.2: Matrices de confusión de los principales experimentos de generalización intersesión. DP: `de_pie`; T: `tumbado`; D2: `dos_patas`; B: `baile`.

Las matrices intersesión muestran un patrón muy distinto al observado en la partición aleatoria. El número de errores aumenta significativamente y aparecen confusiones entre clases que en la evaluación aleatoria estaban prácticamente separadas.

En el protocolo 21/04 → 24/04 con `baseline_less_aug`, la clase `de_pie` mantiene un recall razonable, pero `tumbado` se degrada de forma severa, con un número elevado

de muestras predichas como `de_pie` o `baile`. Esto indica que las posturas estáticas son especialmente sensibles al cambio de distribución del canal: una misma postura física puede generar una firma CSI distinta si varían ligeramente la orientación de antenas, el entorno o la geometría de reflexión.

La variante `domain_adv_soft` mejora parcialmente el comportamiento sobre las clases dinámicas en el protocolo 21/04 → 24/04: `dos_patas` y `baile` presentan mayor estabilidad que en el baseline. Sin embargo, la clase `tumbado` queda prácticamente absorbida por otras clases, lo que explica por qué la mejora en *balanced accuracy* no se traduce en una mejora clara del macro-F1.

En el sentido inverso, 24/04 → 21/04, el rendimiento global mejora, pero la matriz muestra que el modelo sigue cometiendo errores importantes entre clases estáticas y dinámicas. En particular, parte de las muestras `de_pie` se reparten entre `tumbado` y `baile`, mientras que `dos_patas` se confunde en ocasiones con `baile`. Este resultado confirma que el modelo conserva cierta capacidad de discriminación temporal, pero no logra aprender una representación completamente invariante al dominio de captura.

Estos resultados muestran que la adaptación de dominio es una línea prometedora, pero no trivial. En datasets pequeños y con pocos dominios disponibles, los mecanismos adversariales pueden eliminar información útil para la clasificación si la presión de invariancia es demasiado intensa. Por ello, las técnicas de dominio deben calibrarse cuidadosamente y, previsiblemente, requieren más sesiones de captura para aportar mejoras consistentes.

6.7. Análisis por tipo de comportamiento

Los resultados anteriores permiten diferenciar dos niveles de dificultad en el problema. El primero corresponde a la separación entre comportamientos estáticos y dinámicos. En partición aleatoria, esta separación se aprende de forma muy robusta: las clases `de_pie` y `tumbado` apenas se confunden con `dos_patas` y `baile`. Esto sugiere que la evolución temporal de la CSI contiene información suficiente para distinguir posturas relativamente estables de movimientos con mayor variabilidad temporal.

El segundo nivel corresponde a la discriminación fina dentro de cada grupo. En las clases dinámicas, la confusión entre `dos_patas` y `baile` aparece incluso en la partición aleatoria, aunque con frecuencia baja. Esta confusión puede deberse a que ambos comportamientos producen perturbaciones temporales complejas sobre el canal y comparten

componentes dinámicas similares.

En la evaluación intersesión, la dificultad aumenta de forma considerable. Las clases estáticas dejan de ser triviales, especialmente `tumbado`. Este resultado es coherente con la física del canal: en una postura estática, la información discriminativa depende mucho de la geometría exacta de las trayectorias de propagación. Si la configuración del entorno cambia, la diferencia entre dos posturas puede verse desplazada o incluso invertida en el espacio de características.

Por tanto, la clasificación de comportamientos dinámicos parece depender en parte de patrones temporales más robustos, mientras que la clasificación de posturas estáticas es más sensible a la distribución absoluta del canal en cada sesión. Esta observación resulta relevante para el diseño de futuras versiones del sistema, donde podría ser conveniente combinar información temporal relativa con mecanismos de calibración por sesión.

6.8. Discusión crítica de los resultados

Los resultados obtenidos permiten extraer varias conclusiones experimentales importantes.

En primer lugar, el rendimiento en partición aleatoria confirma que la señal CSI en banda mmWave contiene información suficiente para clasificar comportamientos predefinidos del robot con alta precisión. Este resultado valida la hipótesis de partida del trabajo: las perturbaciones inducidas por el robot en el canal inalámbrico generan patrones temporales discriminativos que pueden ser explotados por modelos de aprendizaje automático.

En segundo lugar, la comparación entre arquitecturas muestra que el problema puede resolverse de forma efectiva mediante modelos de distinta naturaleza. El alto rendimiento del SVM-RBF indica que las características estadísticas agregadas ya separan bien las clases bajo condiciones controladas. El buen comportamiento del Transformer reducido sugiere que los mecanismos de atención pueden capturar dependencias temporales relevantes. CsiNet, por su parte, ofrece una solución especializada para tensores CSI, con buen rendimiento y mejor integración con el pipeline online.

En tercer lugar, la evaluación intersesión revela la principal limitación del sistema: el desplazamiento de dominio. La diferencia entre los resultados de partición aleatoria y cross-day es demasiado grande como para ignorarse. Una evaluación basada únicamente en partición aleatoria habría llevado a una conclusión excesivamente optimista. En cambio,

la evaluación intersesión muestra que el sistema todavía necesita mecanismos adicionales de robustez antes de poder desplegarse en escenarios no controlados.

En cuarto lugar, las técnicas de regularización no producen mejoras uniformes. La reducción de técnicas de aumento mejora el rendimiento global, mientras que las variantes adversariales solo aportan beneficios parciales y dependen fuertemente de los hiperparámetros. Este resultado es importante porque muestra que no basta con introducir técnicas avanzadas de dominio: su efecto debe validarse experimentalmente en cada escenario.

Finalmente, los resultados justifican la elección de CsiNet baseline-less-aug como modelo principal del sistema. Aunque SVM-RBF y Transformer alcanzan valores ligeramente superiores en partición aleatoria, CsiNet ofrece un equilibrio más adecuado entre rendimiento, diseño específico para CSI, extensibilidad y despliegue online. Esta elección es coherente con el objetivo general del TFM, que no se limita a maximizar una métrica offline, sino que busca construir un sistema completo de clasificación sobre flujo continuo de medidas.

6.9. Validación funcional del pipeline online y gemelo digital

Los resultados presentados en las secciones anteriores evalúan el comportamiento del clasificador en condiciones offline, utilizando conjuntos de prueba previamente segmentados y etiquetados. Sin embargo, uno de los objetivos del trabajo no es únicamente entrenar un modelo con buenas métricas sobre ficheros estáticos, sino comprobar que dicho modelo puede integrarse en un flujo operativo de inferencia.

Por este motivo, además de la evaluación cuantitativa offline, se desarrolló una validación funcional del pipeline online. Esta validación tiene un objetivo distinto al de las pruebas anteriores: no pretende estimar de nuevo la precisión del modelo sobre un conjunto etiquetado, sino comprobar que la cadena completa de procesamiento puede ejecutarse de forma continua y alimentar una representación externa del sistema.

El pipeline online cierra el ciclo completo del sistema propuesto:

$$\begin{aligned} &\text{CSI} \rightarrow \text{ventana temporal} \rightarrow \text{preprocesado} \rightarrow \text{modelo CsiNet} \\ &\rightarrow \text{predicción estabilizada} \rightarrow \text{JSON} \rightarrow \text{gemelo digital.} \end{aligned} \tag{6.5}$$

Esta validación resulta relevante porque permite comprobar aspectos que no aparecen en la evaluación offline: tiempo de inferencia, estabilidad temporal de la salida, formato

de intercambio entre módulos y capacidad de integración con un entorno de simulación.

6.9.1. Objetivo y alcance de la validación online

La validación online se plantea como una prueba funcional del sistema desplegado. Su objetivo es verificar que el clasificador seleccionado, CsiNet `baseline_less_aug`, puede recibir ventanas generadas de forma continua, ejecutar inferencia periódica y producir una salida semántica consumible por otros módulos.

Es importante diferenciar esta validación de la evaluación offline realizada previamente. En la evaluación offline, cada muestra del conjunto de prueba tiene una etiqueta conocida y se calcula directamente si la predicción coincide con dicha etiqueta. En la validación online, en cambio, el sistema opera como un flujo continuo. Las predicciones se generan sobre ventanas sucesivas y pueden estar afectadas por solapamiento temporal, fluctuaciones instantáneas y mecanismos de estabilización.

Por tanto, en esta sección no se reporta una nueva exactitud online equivalente a la accuracy offline. Para obtener una métrica de precisión online completamente comparable sería necesario disponer de una fuente externa de etiquetado sincronizada, como odometría del robot, registro temporal de comandos, vídeo anotado o sensores adicionales. Esa sincronización queda fuera del alcance de esta versión del sistema. En su lugar, se evalúan dos aspectos funcionales:

- el coste temporal del pipeline de inferencia;
- la estabilidad de la salida antes y después del mecanismo de suavizado temporal.

De esta forma, la validación online complementa a la evaluación offline: las secciones anteriores demuestran la capacidad discriminativa del modelo, mientras que esta sección demuestra la viabilidad de integrarlo en una arquitectura operativa.

6.9.2. Medida de latencia del pipeline

La primera magnitud evaluada en el pipeline online es la latencia de procesamiento. Esta medida puede obtenerse sin necesidad de que el robot esté ejecutando una clase concreta, ya que el objetivo no es evaluar la corrección semántica de la predicción, sino el tiempo que tarda el sistema en transformar una ventana de entrada en una salida.

La latencia total por ventana se descompone en varias etapas:

- lectura y parsing de medidas CSI;
- construcción de la ventana temporal;
- preprocesado y normalización;
- inferencia mediante CsiNet;
- estabilización temporal de la predicción;
- escritura del fichero JSON compartido.

Formalmente, el tiempo total de una iteración online puede expresarse como:

$$t_{online} = t_{parse} + t_{window} + t_{pre} + t_{model} + t_{stab} + t_{json}, \tag{6.6}$$

donde t_{parse} representa el tiempo de lectura y extracción de la información de canal, t_{window} el coste de actualización del buffer temporal, t_{pre} el preprocesado, t_{model} la inferencia del modelo, t_{stab} el mecanismo de estabilización y t_{json} la escritura del estado compartido.

La Tabla 6.5 recoge el tiempo extremo a extremo medido durante dos pruebas funcionales del pipeline online. En ambas pruebas se emplearon ventanas de 120 medidas CSI, un salto de 2 medidas entre inferencias y una duración útil de 60 segundos tras el periodo inicial de calentamiento. El tiempo reportado incluye la ejecución completa del predictor sobre cada ventana y, por tanto, representa la latencia observada por el sistema desde que una ventana está disponible hasta que se obtiene una predicción.

Tabla 6.5: Tiempo extremo a extremo del pipeline de inferencia online.

Prueba	**Ventanas**	**Media (ms)**	**Desv. típ. (ms)**	**Máximo (ms)**
Secuencia estable	50	1151,35	26,31	1246,80
Secuencia con transiciones	50	1135,65	26,29	1205,38

Los resultados muestran que el pipeline genera 50 predicciones en 60 segundos útiles en cada prueba, lo que equivale aproximadamente a 0,83 predicciones por segundo. El tiempo medio extremo a extremo se sitúa entre 1,14 y 1,15 segundos por ventana. Esta latencia está condicionada por la implementación empleada para la prueba, donde el predictor se ejecuta como proceso externo para cada ventana. Por tanto, no debe interpretarse como

el límite computacional del modelo, sino como la latencia funcional del prototipo online. Una implementación optimizada mantendría el modelo cargado en memoria y reduciría el coste asociado al lanzamiento repetido del script de inferencia.

Desde el punto de vista del sistema, esta medida permite verificar si el cuello de botella se encuentra en el acceso al flujo de medidas, en el preprocesado o en la propia inferencia del modelo. Esta distinción es importante para futuras optimizaciones, ya que una latencia dominada por el parsing requeriría mejoras distintas a una latencia dominada por la red neuronal.

6.9.3. Estabilización temporal de la salida

Una característica observada durante el diseño del pipeline online es que las predicciones instantáneas pueden fluctuar entre clases consecutivas, incluso cuando la entrada procede de un flujo relativamente estable. Este comportamiento es habitual en sistemas basados en *wireless sensing*, ya que pequeñas variaciones del canal pueden modificar localmente la distribución de la ventana de entrada.

Por esta razón, el sistema incorpora una etapa de estabilización temporal posterior a la inferencia. La predicción directa del modelo se interpreta como una salida instantánea, mientras que la salida publicada al resto del sistema se obtiene tras aplicar un mecanismo de suavizado.

El objetivo de esta etapa no es aumentar artificialmente la precisión offline, sino producir una señal más estable para módulos consumidores, como el gemelo digital. En un sistema de monitorización, una salida que cambia de clase de forma espuria puede resultar menos útil que una salida ligeramente más conservadora pero temporalmente coherente.

La estabilización puede formularse de forma general como una función sobre una secuencia reciente de predicciones:

$$\hat{y}_t^{stable} = g(\hat{y}_{t-k+1}, \hat{y}_{t-k+2}, \ldots, \hat{y}_t), \tag{6.7}$$

donde $\hat{y}_t$ es la predicción instantánea en el instante t, k es el tamaño del historial considerado y $g(\cdot)$ representa la estrategia de consenso temporal. En la implementación del pipeline se contemplan mecanismos como votación mayoritaria, suavizado de probabilidades e histéresis de cambio de estado.

Para cuantificar el efecto de esta etapa se analiza el número de cambios de clase antes y después de la estabilización. La Tabla 6.6 recoge los resultados obtenidos en dos pruebas

funcionales del pipeline online: una secuencia estable y una secuencia con transiciones entre clases.

Tabla 6.6: Efecto de la estabilización temporal sobre la salida online.

Secuencia	Ventanas	Cambios sin suavizado	Cambios suavizados	Reducción (%)
Secuencia estable	50	0	0	0,00
Secuencia con transiciones	50	11	7	36,36

En la secuencia estable, tanto la predicción instantánea como la predicción suavizada permanecen constantes durante toda la prueba. Este resultado confirma que el pipeline no introduce oscilaciones artificiales cuando la salida del modelo es estable.

En la secuencia con transiciones, la salida cruda del modelo presenta 11 cambios de clase, mientras que la salida estabilizada reduce este número a 7 cambios. Esto supone una reducción del 36.36 % en el número de transiciones publicadas al resto del sistema. El suavizado no pretende modificar la decisión del clasificador de forma arbitraria, sino evitar que cambios aislados o de corta duración se propaguen directamente al gemelo digital.

Este resultado debe interpretarse como una métrica de estabilidad temporal, no como una métrica de precisión semántica. La evaluación de precisión del modelo se realiza mediante los protocolos offline e intersesión descritos en las secciones anteriores. En cambio, esta prueba online evalúa si la salida publicada por el sistema resulta suficientemente estable para alimentar módulos consumidores en tiempo real, como el fichero JSON compartido o el gemelo digital.

6.9.4. Salida JSON e intercambio de estado

Una vez obtenida la predicción estabilizada, el pipeline escribe el estado inferido en un fichero JSON. Este fichero actúa como mecanismo de intercambio desacoplado entre el módulo de inferencia y los módulos consumidores. La ventaja de esta aproximación es que la inferencia no depende directamente del simulador, y el gemelo digital puede leer el estado a su propio ritmo.

El Listado 6.1 muestra un ejemplo simplificado del formato utilizado para publicar el estado del sistema.

Listing 6.1: Ejemplo de salida JSON generada por el pipeline online.

```
{
    "predicted_class": "dos_patas",
```

```
    "stable_class": "dos_patas",
    "confidence": 0.91,
    "timestamp": 1715189200,
    "source": "mmwave_csi"
}
```

El campo `predicted_class` representa la predicción instantánea del modelo, mientras que `stable_class` contiene el estado resultante tras aplicar el mecanismo de estabilización. El campo `confidence` proporciona una medida de confianza asociada a la clase seleccionada, y `timestamp` permite sincronizar la actualización con otros módulos del sistema.

Este diseño permite extender fácilmente la arquitectura en trabajos futuros. Por ejemplo, podrían añadirse nuevos campos procedentes de otras fuentes de sensado, como estimaciones de cámara, detecciones YOLO, odometría, LiDAR, IMU o información procedente de enlaces 5G.

6.9.5. Integración con el gemelo digital

Como demostración funcional del sistema, la salida del pipeline online se integró con un gemelo digital del robot implementado en Isaac Sim. El objetivo de esta integración no es validar de nuevo la precisión del clasificador, sino demostrar que la información semántica inferida a partir del canal inalámbrico puede ser consumida por una representación virtual del sistema físico.

La Figura 6.3 muestra el escenario de simulación utilizado para visualizar el estado inferido del robot.

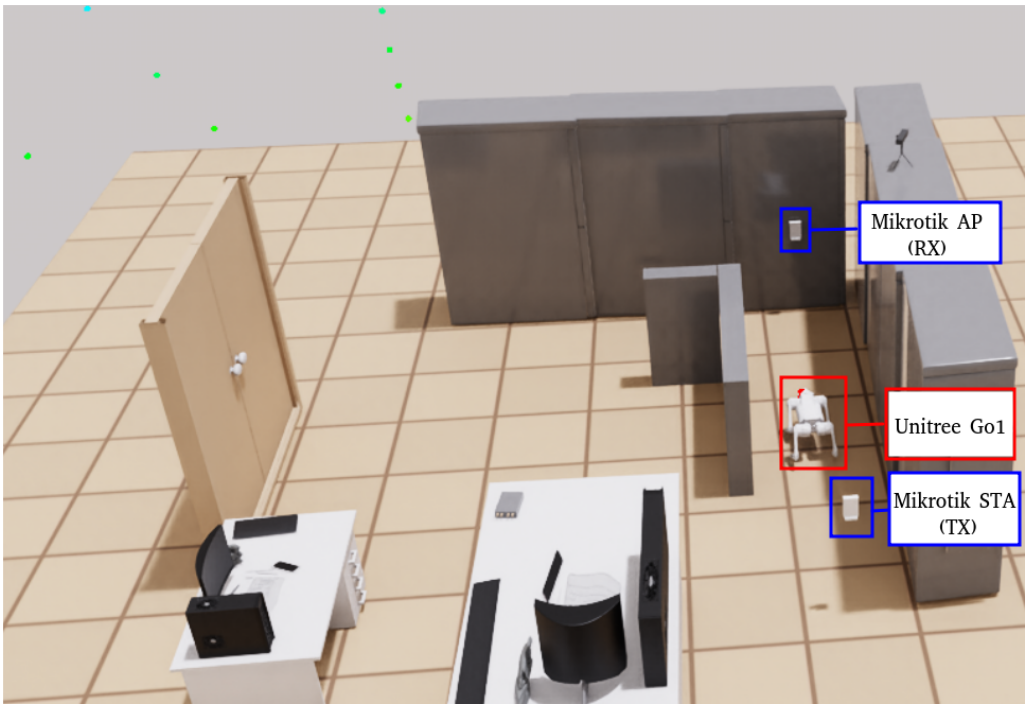


Figura 6.3: Escenario del gemelo digital en Isaac Sim con el robot Unitree Go1 y las antenas MikroTik empleadas en la representación del sistema.

Durante la ejecución, Isaac Sim monitoriza el fichero JSON generado por el pipeline y actualiza la representación del robot en función de la clase estabilizada. La correspondencia entre predicción y representación virtual se resume en la Tabla 6.7.

Tabla 6.7: Correspondencia entre clase inferida y representación en el gemelo digital.

Clase inferida	**Representación en el gemelo digital**
`de_pie`	Estado del robot de pie
`tumbado`	Estado del robot tumbado
`dos_patas`	Comportamiento predefinido de dos patas
`baile`	Comportamiento predefinido de baile

La Figura 6.4 muestra las cuatro representaciones semánticas asociadas a las clases del clasificador: `de_pie`, `tumbado`, `dos_patas` y `baile`. Esta composición permite observar cómo la etiqueta generada por el modelo se transforma en un estado visual interpretable dentro del entorno virtual.

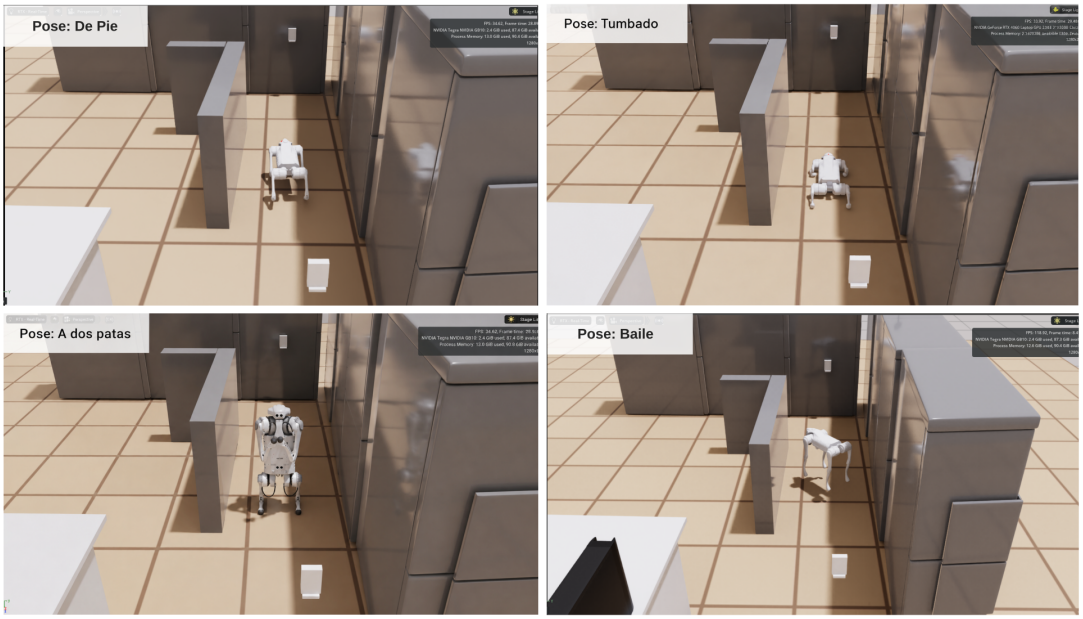


Figura 6.4: Representación de las cuatro clases inferidas en el gemelo digital.

Esta integración convierte el clasificador en un módulo semántico dentro de una arquitectura más amplia. La salida del sistema deja de ser únicamente una etiqueta generada por consola y pasa a actualizar una representación virtual sincronizada. Esto permite utilizar el gemelo digital como una plataforma común para visualizar, validar y combinar diferentes fuentes de percepción.

En este sentido, la arquitectura no se limita necesariamente a medidas CSI en banda mmWave. El mismo mecanismo de estado compartido podría incorporar en el futuro sensado basado en 5G, cámaras con detección de objetos mediante YOLO, sensores LiDAR, unidades inerciales u otras fuentes capaces de aportar información espacial o semántica sobre el entorno. De esta forma, el gemelo digital puede actuar como capa de fusión, visualización y validación funcional de sistemas de percepción heterogéneos.

6.9.6. Limitaciones de la validación online

La validación funcional del pipeline online presenta varias limitaciones que deben ser consideradas. En primer lugar, no sustituye a una evaluación cuantitativa completa con etiquetado temporal sincronizado. Para medir la precisión online de forma rigurosa sería necesario disponer de una referencia externa que indicase en cada instante el comportamiento real del robot. Esta referencia podría obtenerse mediante odometría, sensores internos, vídeo anotado o registro sincronizado de comandos.

En segundo lugar, la integración con el gemelo digital valida la interoperabilidad del sistema, pero no implica por sí misma una mejora de precisión. El gemelo digital actúa como consumidor de la salida semántica y como plataforma de visualización, mientras que la calidad del clasificador debe evaluarse mediante las métricas offline e intersesión presentadas anteriormente.

En tercer lugar, la estabilización temporal introduce un compromiso entre estabilidad y rapidez de reacción. Un suavizado fuerte reduce oscilaciones espurias, pero también puede retrasar la detección de cambios reales de comportamiento. Este compromiso deberá ajustarse en función del escenario de aplicación.

A pesar de estas limitaciones, la validación online demuestra que el modelo entrenado puede integrarse en un flujo operativo completo. Esto diferencia el sistema de una evaluación puramente offline y abre la puerta a futuras versiones con adquisición en vivo, etiquetado sincronizado y fusión multimodal con otras fuentes de sensado.

7. Conclusiones y trabajo futuro

7.1. Conclusiones

El presente Trabajo Fin de Máster ha abordado el problema de la clasificación de comportamientos de un robot cuadrúpedo mediante técnicas de mmWave sensing y aprendizaje automático, utilizando las perturbaciones producidas sobre el canal inalámbrico como fuente principal de información.

A diferencia de enfoques tradicionales basados en visión artificial, sensores inerciales, LiDAR u odometría interna, el sistema desarrollado se fundamenta en un paradigma de *wireless sensing* pasivo, donde el comportamiento del robot se infiere indirectamente a partir de variaciones en las medidas CSI capturadas entre dos antenas mmWave. Esta aproximación no pretende sustituir a los sensores embarcados del robot, sino proporcionar una fuente externa y complementaria de percepción, potencialmente útil en escenarios donde se desea monitorizar el comportamiento del sistema sin depender exclusivamente de la telemetría interna.

El trabajo realizado ha permitido demostrar experimentalmente que este enfoque es viable utilizando hardware comercial COTS. En concreto, se emplearon dispositivos MikroTik wAP 60G configurados mediante firmware y herramientas de investigación que permiten acceder a medidas internas del canal. A partir de esta infraestructura se diseñó un pipeline completo de adquisición, almacenamiento, preprocesado, entrenamiento, evaluación e inferencia, cubriendo no solo la construcción del dataset, sino también la integración del clasificador en un flujo operativo.

Durante el desarrollo experimental se construyó un dataset propio compuesto por múltiples sesiones de captura realizadas en distintos días y bajo condiciones de adquisición diferenciadas. Cada muestra se representó mediante ventanas temporales de 120 medidas CSI, transformadas en tensores tridimensionales capaces de preservar simultáneamente la evolución temporal de la señal y la estructura interna de los valores de canal. Esta representación permitió formular el problema como una tarea de clasificación supervisada de secuencias multivariantes.

El sistema final se centró en la clasificación de cuatro comportamientos predefinidos del robot:

- `de_pie`;
- `tumbado`;
- `dos_patas`;
- `baile`.

Además, durante la fase de adquisición se capturaron muestras correspondientes a la condición `no_robot`. Estas muestras se conservaron como condición de control, pero no se incorporaron al clasificador final, ya que el objetivo principal de esta versión del sistema era discriminar estados y movimientos del robot una vez presente en la zona de sensado, no resolver simultáneamente el problema de detección de presencia.

Desde el punto de vista de inteligencia artificial, se evaluaron distintas familias de modelos, incluyendo SVM con núcleo RBF, perceptrones multicapa, redes convolucionales simples, Transformers reducidos y la arquitectura CsiNet propuesta. Esta comparativa permitió comprobar que las señales CSI contienen información altamente discriminativa para el problema tratado. En partición aleatoria estratificada, varios modelos alcanzaron resultados superiores al 96 % de macro-F1, lo que valida la hipótesis de partida: los comportamientos del robot generan perturbaciones suficientemente diferenciables en el canal inalámbrico.

Entre las configuraciones evaluadas, la variante CsiNet `baseline_less_aug` fue seleccionada como modelo principal del sistema. Esta configuración alcanzó aproximadamente:

- 97,79 % de *accuracy*;
- 96,34 % de macro-F1;
- 96,32 % de *balanced accuracy*.

Estos resultados muestran que la arquitectura propuesta es capaz de clasificar correctamente los comportamientos del robot en condiciones donde entrenamiento y prueba comparten distribuciones de adquisición similares. Aunque otros modelos, como SVM-RBF o el Transformer reducido, obtuvieron resultados muy competitivos en partición aleatoria, CsiNet se seleccionó como modelo principal por su equilibrio entre rendimiento, estructura específica para tensores CSI, extensibilidad hacia salidas auxiliares y facilidad de integración en un pipeline de inferencia online.

Uno de los hallazgos más relevantes del trabajo ha sido la identificación del problema de generalización intersesión. Cuando el modelo se entrena en una sesión de captura y se evalúa en otra sesión distinta, el rendimiento disminuye de forma significativa. Este fenómeno confirma que pequeñas variaciones físicas del entorno, orientación de antenas, reflexiones multitrayecto o cambios ambientales producen desplazamientos importantes en la distribución estadística de la señal CSI.

En este sentido, el trabajo no solo demuestra la viabilidad del enfoque propuesto, sino que también evidencia una de las principales limitaciones actuales de los sistemas de *wireless sensing* basados en CSI: la dificultad de aprender representaciones invariantes frente a cambios de dominio. Una evaluación basada únicamente en particiones aleatorias habría ofrecido una visión demasiado optimista del sistema. La evaluación intersesión permitió mostrar de forma más realista el comportamiento del modelo ante condiciones no vistas.

Asimismo, los experimentos realizados muestran que las estrategias de regularización deben aplicarse con cautela. La variante con técnicas de aumento suavizada obtuvo mejores resultados que configuraciones con técnicas de aumento más agresivas, lo que sugiere que, en señales físicas como la CSI, no toda perturbación artificial contribuye necesariamente a mejorar la generalización. Del mismo modo, las variantes con adaptación adversarial de dominio ofrecieron mejoras parciales en algunas métricas de equilibrio entre clases, pero no resolvieron completamente el fenómeno de *session shift*.

Otro resultado importante del trabajo ha sido el diseño e integración de un pipeline de inferencia online. Este pipeline permite consumir medidas CSI en flujo continuo, construir ventanas temporales, aplicar el preprocesado correspondiente, ejecutar el modelo CsiNet, estabilizar temporalmente la predicción y publicar el estado inferido mediante un fichero JSON. Aunque la evaluación principal del clasificador se realizó en condiciones offline controladas, esta integración demuestra que el sistema puede trasladarse desde un entorno de entrenamiento a una arquitectura funcional de inferencia.

Finalmente, se desarrolló una demostración complementaria mediante un gemelo digital en Isaac Sim. Este gemelo digital consume la salida semántica del pipeline y representa visualmente los comportamientos inferidos del robot. Esta integración no debe interpretarse como una nueva métrica de precisión, sino como una validación funcional del cierre del sistema: medidas inalámbricas, modelo de inteligencia artificial, estado semántico y representación virtual sincronizada.

En conjunto, el trabajo desarrollado demuestra que las tecnologías de mmWave sensing combinadas con modelos de aprendizaje automático constituyen una línea prometedora para la monitorización pasiva de robots y entidades dinámicas mediante perturbaciones del canal inalámbrico. Al mismo tiempo, el análisis realizado delimita claramente los retos pendientes, especialmente en términos de robustez, generalización e integración multimodal.

7.2. Limitaciones del trabajo

A pesar de los resultados obtenidos, el sistema desarrollado presenta diversas limitaciones que deben tenerse en cuenta al interpretar el alcance del trabajo.

La primera limitación corresponde al número de sesiones disponibles para entrenamiento y evaluación. Aunque el dataset generado permitió validar experimentalmente el enfoque propuesto, el número de días de captura sigue siendo reducido para entrenar modelos altamente robustos frente a cambios de dominio. Esta limitación afecta especialmente al análisis intersesión, donde sería deseable disponer de muchas más sesiones independientes, con variaciones controladas de entorno, orientación, temperatura, distancia y posición relativa del robot.

La segunda limitación está relacionada con el entorno físico de adquisición. Las capturas se realizaron en un escenario concreto y bajo una geometría relativamente controlada. Por tanto, el modelo puede aprender no solo características asociadas a los comportamientos del robot, sino también propiedades específicas del entorno experimental. Esta dependencia dificulta la extrapolación directa a otros laboratorios, distancias, orientaciones o configuraciones de propagación.

Otra limitación importante reside en la configuración hardware utilizada. El sistema se basa en una única pareja transmisor–receptor y en una única portadora mmWave. Como consecuencia, no se dispone de información Doppler explícita ni de capacidades avanzadas de estimación espacial similares a las disponibles en radares FMCW, sistemas multiportadora o configuraciones Multiple-Input Multiple-Output (MIMO). Toda la información dinámica debe inferirse indirectamente a partir de la evolución temporal de la fase y la magnitud de la CSI.

También debe considerarse el desbalance entre clases. Las clases estáticas cuentan con

más muestras que las clases dinámicas, debido a la mayor facilidad de capturar posturas mantenidas frente a comportamientos en movimiento. Aunque se emplearon métricas como macro-F1 y *balanced accuracy* para reducir el sesgo hacia clases mayoritarias, este desbalance sigue siendo un factor que puede afectar a la interpretación de los resultados.

Otra limitación del sistema es que el problema se formuló como clasificación cerrada de comportamientos predefinidos. El modelo distingue entre cuatro clases concretas, pero no estima de forma continua la postura, trayectoria, orientación o velocidad del robot. Por tanto, el sistema no debe interpretarse como un método general de reconstrucción de movimiento, sino como una prueba de concepto para inferencia semántica discreta basada en canal inalámbrico.

La validación online también presenta limitaciones. Aunque se implementó un pipeline funcional de inferencia y comunicación mediante JSON, no se realizó una evaluación cuantitativa online completamente sincronizada con una fuente externa de verdad terreno. Para medir con precisión el rendimiento online sería necesario disponer de odometría sincronizada, registro temporal de comandos, sensores internos o vídeo anotado. En consecuencia, la parte online debe entenderse como una validación funcional del sistema, no como un benchmark exhaustivo de precisión en tiempo real.

Por último, la integración con el gemelo digital se planteó como demostración tecnológica. El entorno de Isaac Sim permite visualizar el estado inferido por el clasificador, pero no resuelve por sí mismo problemas de precisión, sincronización física completa o control cerrado del robot. Su valor principal en este trabajo reside en demostrar la posibilidad de consumir una salida semántica generada a partir de wireless sensing y representarla dentro de una plataforma virtual.

7.3. Trabajo futuro

Las limitaciones identificadas durante el desarrollo del proyecto abren múltiples líneas de trabajo futuro, tanto desde el punto de vista de inteligencia artificial como desde la perspectiva de comunicaciones inalámbricas, robótica y gemelos digitales.

Una primera línea de trabajo consiste en ampliar significativamente el número de sesiones y escenarios de captura. Disponer de un dataset más grande, diverso y balanceado permitiría entrenar modelos más robustos frente al fenómeno de *session shift*. Esta ampliación debería incluir variaciones sistemáticas de distancia, orientación de antenas, posición

del robot, entorno físico y condiciones ambientales.

Una segunda línea especialmente relevante es la incorporación de técnicas avanzadas de adaptación de dominio. Los resultados obtenidos muestran que la adaptación adversarial puede aportar beneficios parciales, pero no resuelve de forma completa la generalización intersesión. En trabajos futuros podrían explorarse métodos más avanzados, como aprendizaje contrastivo, aprendizaje autosupervisado, adaptación test-time, normalización específica de dominio o entrenamiento con dominios múltiples.

También sería interesante estudiar estrategias de calibración por sesión. Dado que el canal mmWave es muy sensible a la configuración física, podría incorporarse una fase breve de calibración antes del despliegue, utilizando muestras no etiquetadas o secuencias de referencia. Esta calibración permitiría ajustar la distribución de entrada sin necesidad de reentrenar completamente el modelo.

Desde el punto de vista hardware, una línea de mejora clara consiste en incorporar información Doppler explícita. El sistema actual utiliza medidas CSI proporcionadas por la plataforma experimental, pero no dispone de mapas rango-Doppler ni de estimación directa de velocidad radial. El uso de radares FMCW, sistemas multiportadora o hardware específicamente diseñado para sensing permitiría capturar información dinámica más rica, previsiblemente útil para distinguir movimientos con mayor robustez.

Del mismo modo, el empleo de configuraciones MIMO o múltiples enlaces mmWave permitiría explotar información espacial adicional. Varios pares transmisor–receptor situados en distintas posiciones podrían proporcionar observaciones complementarias del robot, reduciendo la dependencia de una única trayectoria dominante y mejorando la capacidad de generalización.

Desde el punto de vista de modelos de inteligencia artificial, podrían explorarse arquitecturas híbridas que combinen convoluciones, atención temporal y mecanismos recurrentes. El Transformer reducido evaluado en este trabajo obtuvo un rendimiento muy alto en partición aleatoria, por lo que una línea futura natural sería estudiar modelos de atención más robustos, especialmente diseñados para generalizar entre sesiones y no solo para ajustar correctamente el conjunto de entrenamiento.

Otra línea importante es la integración multimodal. Aunque el objetivo de este trabajo fue demostrar la viabilidad de clasificar comportamientos a partir de CSI mmWave, en un sistema real sería razonable combinar esta fuente con otras modalidades: cámaras RGB, detección de objetos mediante YOLO, LiDAR, IMU, odometría interna del robot o incluso

señales procedentes de redes 5G. Esta fusión permitiría compensar las limitaciones de cada sensor individual y construir una representación más robusta del entorno.

En particular, incorporar odometría interna del robot como señal auxiliar durante el entrenamiento permitiría construir modelos más ricos. La odometría podría utilizarse como etiqueta temporal, como variable auxiliar de supervisión o como referencia para evaluar de forma más rigurosa la inferencia online. Esto permitiría pasar de una clasificación discreta de comportamientos a una estimación más continua del estado del robot.

Otra línea futura consiste en profundizar en la integración con gemelos digitales. El sistema desarrollado ya demuestra que una predicción generada a partir de CSI puede consumirse desde Isaac Sim mediante un fichero de estado compartido. Sin embargo, esta idea puede ampliarse hacia una plataforma de fusión sensorial, donde distintas fuentes de percepción alimenten una representación virtual sincronizada del sistema físico. En este escenario, el gemelo digital no actuaría solo como visualizador, sino como capa de integración, supervisión y validación de múltiples mecanismos de sensing.

Además, el enfoque desarrollado no se limita exclusivamente a robots cuadrúpedos. En trabajos futuros podría extenderse a otros sistemas móviles, interacción humano–robot, monitorización industrial, vigilancia pasiva, detección de presencia o clasificación de actividades en entornos donde no sea deseable utilizar cámaras por privacidad o condiciones ambientales.

Finalmente, sería especialmente interesante validar el sistema en escenarios reales más complejos, incluyendo múltiples entidades simultáneas, obstáculos móviles, cambios de iluminación, presencia humana, variaciones de mobiliario y condiciones de propagación altamente dinámicas. Este tipo de validación permitiría evaluar hasta qué punto el enfoque puede evolucionar desde una prueba de concepto experimental hacia un sistema desplegable en entornos operativos reales.

7.4. Reflexión final

El desarrollo de este Trabajo Fin de Máster ha permitido abordar un problema interdisciplinar situado en la intersección entre comunicaciones inalámbricas, inteligencia artificial, robótica y gemelos digitales.

Más allá de los resultados concretos obtenidos, el trabajo pone de manifiesto el potencial de las señales inalámbricas como fuente de información contextual sobre el entorno

físico. Las ondas de radio no solo transportan datos entre dispositivos, sino que también se ven afectadas por los objetos, movimientos y cambios presentes en el espacio de propagación. Interpretar esas perturbaciones mediante modelos de inteligencia artificial permite transformar una infraestructura de comunicaciones en una infraestructura de percepción.

En este sentido, tecnologías inicialmente diseñadas para transmisión de datos pueden convertirse también en herramientas de sensado, monitorización y supervisión. Esta convergencia entre comunicación y percepción se alinea con la evolución hacia sistemas inteligentes y redes de nueva generación, donde la frontera entre conectar dispositivos y entender el entorno será cada vez más difusa.

El trabajo desarrollado demuestra que, incluso utilizando hardware comercial y un dataset de tamaño moderado, es posible construir un sistema funcional capaz de inferir comportamientos físicos a partir de medidas de canal. Al mismo tiempo, los resultados muestran con claridad que la robustez frente a cambios de dominio continúa siendo el gran reto técnico de este tipo de sistemas.

Por ello, este TFM no debe interpretarse únicamente como una solución cerrada al problema de clasificación de movimientos de un perro robótico, sino como una validación experimental de una línea de investigación más amplia: el uso de wireless sensing e inteligencia artificial para dotar a los sistemas robóticos y a sus gemelos digitales de nuevas formas de percepción externa, pasiva y no visual.

Referencias

Bilik, I., Longman, O., Villeval, S., y Tabrikian, J. (2019, septiembre). The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE Signal Processing Magazine*, *36*(5), 20–31. doi: 10.1109/MSP.2019.2926573

Bilik, I., Longman, O., Villeval, S., y Tabrikian, J. (2020). YOLO-based simultaneous target detection and classification in automotive FMCW radar systems. *Sensors*, *20*(10), 2897. Descargado de `https://www.mdpi.com/1424-8220/20/10/2897` doi: 10.3390/s20102897

Blender Foundation. (s.f.). *Blender: Free and open source 3d creation software.* `https://www.blender.org/`. (Accessed: 2026-06-24)

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., ... Leonard, J. J. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332. doi: 10.1109/TRO.2016.2624754

Cortes, C., y Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*, 273–297. doi: 10.1007/BF00994018

Ding, J., y Wang, Y. (2021). A domain-independent generative adversarial network for activity recognition using WiFi CSI data. *Sensors*, *21*(23), 7852. Descargado de `https://www.mdpi.com/1424-8220/21/23/7852` doi: 10.3390/s21237852

ETSI. (2021). *ETSI EN 302 567 V2.2.1: Multiple-Gigabit/s radio equipment operating in the 60 GHz band; Harmonised Standard for access to radio spectrum* (Inf. Téc. n.º EN 302 567 V2.2.1). European Telecommunications Standards Institute. Descargado de `https://www.etsi.org/deliver/etsi_en/302500_302599/302567/02.02.01_60/en_302567v020201p.pdf`

Fard Moshiri, P., Nabati, M., Shahbazian, R., y Ghorashi, S. A. (2021). CSI-Based Human Activity Recognition using Convolutional Neural Networks. En *2021 11th international conference on computer engineering and knowledge (iccke)* (pp. 7–12). doi: 10.1109/ICCKE54056.2021.9721516

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., ... Lempitsky, V. (2016). Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, *17*(59), 1–35. Descargado de `https://jmlr.org/papers/v17/15-239.html`

GitHub, Inc. (s.f.). *GitHub: Software development platform.* `https://github.com/`. (Accessed: 2026-06-24)

Grieves, M. (2014). *Digital twin: Manufacturing excellence through virtual factory replication.* Descargado de `https://www.researchgate.net/publication/275211047` (White paper)

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... Oliphant, T. E. (2020). Array programming with numpy. *Nature*, *585*(7825), 357–362. doi: 10.1038/s41586-020-2649-2

He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778). doi: 10.1109/CVPR.2016.90

Hunter, J. D. (2007). Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. doi: 10.1109/MCSE.2007.55

IEEE. (2012). *IEEE Standard for Information Technology–Telecommunications and Information Exchange Between Systems–Local and Metropolitan Area Networks–Specific Requirements–Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications–Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band* (n.º IEEE Std 802.11ad-2012). doi: 10.1109/IEEESTD.2012.6392842

Iwana, B. K., y Uchida, S. (2021). An empirical survey of data augmentation for time series classification with neural networks. *PLOS ONE*, *16*(7), e0254841. doi: 10.1371/journal.pone.0254841

Lara Hurtado, F. J., Fernández-Lozano, J. J., y Vázquez-Martín, R. (2019). Una red de sensores inalámbrica basada en hardware y software abierto para colaboración con aplicaciones de robótica a misiones de búsqueda y rescate. En *Xl jornadas de automática: Libro de actas* (pp. 735–740). Ferrol, España: Universidade da Coruña, Servizo de Publicacións. doi: 10.17979/spudc.9788497497169.735

Li, Y., Du, L., Liu, H., Müller, T. S. F., y Waldschmidt, C. (2020). An elaborated signal model for simultaneous range and vector velocity estimation in FMCW radar. *Sensors*, *20*(20), 5860. Descargado de `https://www.mdpi.com/1424-8220/20/20/5860` doi: 10.3390/s20205860

Liu, F., Cui, Y., Masouros, C., Xu, J., Hanzo, T. X., Eldar, Y. C., y Buzzi, S. (2022). Integrated sensing and communications: Toward dual-functional wireless networks

for 6G and beyond. *IEEE Journal on Selected Areas in Communications*, *40*(6), 1728–1767. doi: 10.1109/JSAC.2022.3156632

Ma, Y., Zhou, G., y Wang, S. (2019). Wifi sensing with channel state information: A survey. *ACM Computing Surveys (CSUR)*, *52*(3), 1–36.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. En *Proceedings of the fifth berkeley symposium on mathematical statistics and probability* (Vol. 1, pp. 281–297). University of California Press.

Mateo, P. J., y Widmer, J. (2021). *Mikrotik-researcher-tools: Researcher version of OpenWRT for wAP 60g/LHGG-60ad/wAP 60gx3 AP.* Descargado de `https://github.com/IMDEANetworksWNG/Mikrotik-researcher-tools` (Accedido: 2026-05-19)

Nie, Y., Nguyen, N. H., Sinthong, P., y Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. En *International conference on learning representations.* Descargado de `https://arxiv.org/abs/2211.14730`

NVIDIA Corporation. (s.f.). *NVIDIA Isaac Sim: Robotics simulation and synthetic data generation.* `https://developer.nvidia.com/isaac/sim`. (Accessed: 2026-06-24)

Odonkor, D., Coffelt, J. P., Syrbe, J., y Beetz, M. (2023). Towards a semantic digital twin for marine robotics. En *Isr europe 2023; 56th international symposium on robotics* (pp. 310–317). Descargado de `https://ieeexplore.ieee.org/document/10363088`

Orive, D., López, A., Estévez, E., Orive, A., y Marcos, M. (2021). Desarrollo de gemelos digitales para la simulación e integración de activos de fabricación en la industria 4.0.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019a). Pytorch: An imperative style, high-performance deep learning library. En *Advances in neural information processing systems* (Vol. 32).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... Chintala, S. (2019b). PyTorch: An imperative style, high-performance deep learning library. En *Advances in neural information processing systems 32.* Descargado de `https://arxiv.org/abs/1912.01703`

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Picazo, P., Groshev, M., Blanco, A., Fiandrino, C., de La Oliva, A., y Widmer, J. (2023). waveslam: Empowering accurate indoor mapping using off-the-shelf millimeter-wave self-sensing. En *2023 ieee 98th vehicular technology conference (vtc2023-fall)* (p. 1-7). doi: 10.1109/VTC2023-Fall60731.2023.10333840

Picazo-Martinez, P., Barroso-Fernández, C., Calvillo-Fernandez, A., Groshev, M., Bernardos, C. J., de la Oliva, A., y Mourad, A. (2025). 802.11bf multiband passive sensing: Reusing Wi-Fi signaling for sensing. *arXiv preprint arXiv:2507.22591*. Descargado de `https://arxiv.org/abs/2507.22591` doi: 10.48550/arXiv.2507.22591

Portolés Colón, L., y Rupp, M. (s.f.). Robust csi feedback for high user velocity.

Quevedo Gutierrez, F. A. (2024). *Diseño e implementación de un detector de frecuencia cardiaca y respiratoria utilizando un sensor de banda milimétrica (mmwave)* (Tesis de Master, Universidad Nacional de Colombia). Descargado de `https://repositorio.unal.edu.co/handle/unal/87375`

Redmon, J., Divvala, S., Girshick, R., y Farhadi, A. (2016). You only look once: Unified, real-time object detection. En *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 779–788). doi: 10.1109/CVPR.2016.91

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*. Descargado de `https://arxiv.org/abs/1706.05098`

Segovia, M., y Garcia-Alfaro, J. (2022). Design, modeling and implementation of digital twins. *Sensors*, *22*(14), 5396. doi: 10.3390/s22145396

Seifeldin, M., Saeed, A., Kosba, A. E., El-Keyi, A., y Youssef, M. (2013). Nuzzer: A Large-Scale Device-Free Passive Localization System for Wireless Environments. *IEEE Transactions on Mobile Computing*, *12*(7), 1321–1334. doi: 10.1109/TMC.2012.106

Software Freedom Conservancy. (s.f.). *Git: Distributed version control system.* `https://git-scm.com/`. (Accessed: 2026-06-24)

Strohmayer, J., Sterzinger, R., Wödlinger, M., y Kampel, M. (2024). DATTA: Domain-adversarial test-time adaptation for cross-domain WiFi-based human activity recognition. *arXiv preprint arXiv:2411.13284*. Descargado de `https://arxiv.org/abs/2411.13284`

Thrun, S., Burgard, W., y Fox, D. (2005). *Probabilistic robotics.* MIT Press.

Unitree Robotics. (s.f.-a). *Go1 SDK HighLevel Interfaces.* `https://unitree-docs.readthedocs.io/en/latest/get_started/Go1_Edu.html`. (Accessed: 2026-06-24)

Unitree Robotics. (s.f.-b). *Unitree Go1 Quadruped Robot.* https://www.unitree.com/go1/. (Accessed: 2026-06-24)

van Berlo, B., Elkelany, A., Özçelebi, T., y Meratnia, N. (2021). Millimeter wave sensing: A review of application pipelines and building blocks. *IEEE Sensors Journal*, *21*(9), 10332–10368. doi: 10.1109/JSEN.2021.3057450

Van Rossum, G., y Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. En *Advances in neural information processing systems 30*. Descargado de https://arxiv.org/abs/1706.03762

Wei, Z., Chen, W., Ning, S., Lin, W., Li, N., Lian, B., ... Zhao, J. (2025). A survey on WiFi-based human identification: Scenarios, challenges, and current solutions. *ACM Transactions on Sensor Networks*, *21*(1), 1–32. doi: 10.1145/3708323

Wei, Z., Jia, J., Niu, Y., Wang, L., Wu, H., Yang, H., y Feng, Z. (2025). Integrated sensing and communication channel modeling: A survey. *IEEE Internet of Things Journal*, *12*(12), 18850–18864. doi: 10.1109/JIOT.2024.3449377

Yang, J., Chen, X., Zou, H., Lu, C. X., Wang, D., Sun, S., y Xie, L. (2023). SenseFi: A library and benchmark on deep-learning-empowered wifi human sensing. *Patterns*, *4*(3), 100703. doi: 10.1016/j.patter.2023.100703

Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., y Eickhoff, C. (2021). A transformer-based framework for multivariate time series representation learning. En *Proceedings of the 27th acm sigkdd conference on knowledge discovery & data mining* (pp. 2114–2124). doi: 10.1145/3447548.3467401

Zhao, R., Woodford, T., Wei, T., Qian, K., y Zhang, X. (2020). M-Cube: A millimeter-wave massive MIMO software radio. En *Proceedings of the 26th annual international conference on mobile computing and networking* (pp. 1–14). doi: 10.1145/3372224.3380892

Zhou, K., Yang, Y., Qiao, Y., y Xiang, T. (2021). Domain generalization with mixstyle. En *International conference on learning representations*. Descargado de https://arxiv.org/abs/2104.02008

A. Declaración de uso de herramientas de inteligencia artificial

Durante la elaboración del presente Trabajo Fin de Máster se han utilizado herramientas de inteligencia artificial generativa como apoyo auxiliar en determinadas tareas del proceso de redacción, revisión y preparación del documento. En concreto, se han empleado ChatGPT, basado en GPT-5.5, y Gemini 3.1 Pro. Su uso se ha realizado de forma supervisada y crítica, manteniendo en todo momento la autoría, responsabilidad y validación final por parte del autor del trabajo.

En concreto, estas herramientas se han empleado para las siguientes tareas:

- Mejora de la redacción, reformulación lingüística y revisión de estilo de fragmentos del texto, con el objetivo de aumentar la claridad expositiva, la coherencia interna y la precisión terminológica del documento.
- Apoyo en la estructuración de algunas secciones de la memoria, especialmente en la organización del hilo argumental, la conexión entre apartados y la formulación académica de explicaciones técnicas.
- Búsqueda preliminar y recopilación inicial de referencias bibliográficas relacionadas con *wireless sensing*, CSI, mmWave, aprendizaje automático y técnicas de adaptación de dominio. Todas las referencias incorporadas finalmente al trabajo han sido revisadas, contrastadas y seleccionadas manualmente por el autor.
- Generación o edición asistida de algunas figuras de apoyo visual. En los casos en los que una figura ha sido generada o modificada mediante herramientas de inteligencia artificial, esta circunstancia se indica expresamente en el pie de figura correspondiente.

El uso de estas herramientas no ha sustituido el trabajo experimental, el diseño metodológico, la implementación del sistema, el análisis de resultados ni la interpretación crítica de los mismos. La configuración del sistema experimental, la adquisición del dataset, el desarrollo de los scripts, el entrenamiento de los modelos, la evaluación de resultados y la integración funcional con el *pipeline online* y el gemelo digital han sido realizados por el autor.

Asimismo, las salidas generadas por herramientas de inteligencia artificial no se han incorporado de forma automática al documento. Todo el contenido utilizado ha sido revisado, corregido y adaptado por el autor para asegurar su adecuación al contexto del trabajo, su coherencia con los resultados obtenidos y su rigor técnico.

Por tanto, la inteligencia artificial se ha utilizado como herramienta de apoyo en el proceso de elaboración de la memoria, pero no como sustituto de la contribución intelectual, técnica y experimental desarrollada en este Trabajo Fin de Máster.