



UNIVERSIDAD DE MÁLAGA



E.T.S. INGENIERÍA
INFORMÁTICA

UNIVERSIDAD DE MÁLAGA

Grado en Ingeniería Informática

Sistemas De La Información

**Segmentación de lesiones a partir de imágenes de
resonancia magnética en pacientes con lesiones
cerebrales**

**Lesion segmentation from magnetic resonance in
patients with brain lesions**

Realizado por

Iván Lago Guerreira

Tutorizado por

Dr. Rafael Marcos Luque Baena

Co-tutorizado por

Dr. Esteban José Palomo Ferrer

Departamento

Departamento de Lenguajes y Ciencias de la Computación

Málaga, February 20, 2026



UNIVERSIDAD
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

Sistemas De La Información

**Segmentación de lesiones a partir de imágenes de
resonancia magnética en pacientes con lesiones
cerebrales**

**Lesion segmentation from magnetic resonance in
patients with brain lesions**

Realizado por

Iván Lago Guerreira

Tutorizado por

Dr. Rafael Marcos Luque Baena

Co-tutorizado por

Dr. Esteban José Palomo Ferrer

Departamento

Departamento de Lenguajes y Ciencias de la Computación

UNIVERSIDAD DE MÁLAGA

MÁLAGA, FEBRUARY 20, 2026

Fecha defensa: marzo de 2026

Resumen

La segmentación automática de lesiones en imágenes médicas constituye un área de investigación fundamental para apoyar el diagnóstico y seguimiento de enfermedades neurológicas. En este Trabajo de Fin de Grado se aborda el problema de la segmentación de lesiones en imágenes de resonancia magnética (MRI) de pacientes con metástasis y esclerosis múltiple, con el objetivo de evaluar el rendimiento de diferentes arquitecturas de aprendizaje profundo aplicadas a este tipo de datos.

Para ello, se han empleado dos conjuntos de datos públicos procedentes de fuentes abiertas, que fueron sometidos a un proceso de organización, conversión y normalización previo al entrenamiento. A partir de dichos datos, se entrenaron y compararon dos modelos basados en la infraestructura del algoritmo *nnU-Net v2*: el modelo original de referencia y una versión modificada denominada *U-Mamba*, que introduce módulos de tipo *State Space Model* para mejorar la capacidad de representación espacial y contextual de la red.

El trabajo describe de forma detallada el flujo de preprocesamiento, entrenamiento y evaluación seguido, y presenta un análisis comparativo de los resultados obtenidos mediante métricas de segmentación como el coeficiente *Dice*. Finalmente, se exponen las principales conclusiones, limitaciones del estudio y posibles líneas de investigación futura orientadas a mejorar la segmentación automática de lesiones cerebrales.

Palabras clave: Resonancia magnética (MRI), *nnU-Net v2*, Lesiones cerebrales, segmentación

Abstract

Automatic lesion segmentation in medical imaging is a key research field that supports both the diagnosis and monitoring of neurological disorders. This Bachelor's Thesis addresses the problem of lesion segmentation in magnetic resonance imaging (MRI) of patients with metastasis and multiple sclerosis, aiming to evaluate the performance of different *deep learning* architectures applied to this domain.

Two publicly available datasets were used, both subjected to preprocessing and normalization before training. Based on these data, two models built upon the ***nnU-Net v2*** framework were trained and compared: the standard *nnU-Net v2* configuration and an alternative architecture called ***U-Mamba***, which integrates *State Space Model* modules to enhance spatial and contextual representation.

The work provides a detailed description of the preprocessing, training, and evaluation *pipeline*, and includes a comparative analysis of the results using segmentation metrics such as the *Dice Similarity Coefficient*. Finally, the main conclusions, study limitations, and potential directions for future research are presented, focusing on improving the automatic segmentation of brain lesions.

Keywords: Magnetic resonance imaging (MRI), *nnU-Net v2*, Brain lesions, segmentation

Agradecimientos

En primer lugar, me gustaría expresar mis agradecimientos a mi tutor, Rafael Marcos Luque Baena y a mi cotutor Esteban José Palomo Ferrer y a Ariadna Jiménez Partinen. Gracias a ellos he podido llevar a cabo un trabajo de fin de grado interesante a la vez que desafiante. Agradezco la paciencia y el apoyo que me han brindado. Me han sabido guiar perfectamente en los desafíos que conlleva este TFG.

A los alumnos, profesores, departamentos y, en general, a toda la Universidad de Málaga, por proporcionarme un entorno adecuado en el que poder desarrollarme, aprender y crecer como persona y como profesional.

En lo cercano, me gustaría agradecer especialmente a mi familia. A mis padres y mi hermano, por su apoyo incondicional no solo durante esta etapa, sino durante todo mi camino. Siempre me han empujado hacia delante, disminuyendo mis posibles pérdidas y celebrando mis logros con alegría. No es exagerado decir que todos mis éxitos y la persona que soy a día de hoy se debe en su totalidad, al cariño y apoyo que me han dado durante toda mi vida.

A mis amigos cercanos y, especialmente, a los compañeros de grado que he podido conocer durante esta etapa, con quienes he pasado horas y días en clase, en la cafetería, zonas de descanso, gimnasio, salidas... y con quienes he podido compartir una etapa muy bonita que siempre recordaré con una sonrisa y que, por supuesto, no termina aquí. Agradezco el haber forjado amistades tan honestas y llenas de momentos en los que no se puede evitar llorar de la risa.

Por último me gustaría agradecer a mi pareja, Sandra, por apoyarme no solo en los momentos decisivos, sino en los momentos que no se ven del día a día. Tu compañía, los pequeños detalles, cariños y gestos han conseguido que pueda avanzar en los momentos que más necesitaba. Gracias por el apoyo constante y por hacerme una mejor persona.

Índice

Resumen	1
Abstract	2
Agradecimientos	3
1 Introducción	10
1.1 Motivación	10
1.2 Objetivos	11
1.2.1 Objetivo general	11
1.2.2 Objetivos específicos	11
1.3 Estructura del documento	12
1.4 Tecnologías usadas	13
2 Estado del arte	15
3 Metodología	19
3.1 Conjunto de datos	19
3.1.1 Formato de Almacenamiento y Modalidades de Imagen Estructural	19
3.1.2 Dataset de esclerosis múltiple	22
3.1.3 Dataset de tumores cerebrales metastásicos	23
3.1.4 Estructura de carpetas	24
3.2 Preprocesamiento	27
3.2.1 Motivación del preprocesamiento	27
3.2.2 Normalización espacial mediante plantillas anatómicas	28
3.2.3 Corrección de inhomogeneidades de intensidad (N4 Bias Field Correction)	30
3.2.4 Extracción de cráneo (Skull Stripping)	30
3.3 Algoritmos de entrenamiento	32
3.3.1 <i>deep learning</i>	32
3.3.2 <i>nnU-Net v2</i>	36
3.3.3 <i>U-Mamba</i>	48
3.4 Diseño experimental	53
3.4.1 Preparación y organización de los conjuntos de datos	53
3.4.2 Preprocesamiento previo al entrenamiento	54
3.4.3 <i>Pipeline</i> del algoritmo	55
4 Resultados	63
4.1 Resumen de experimentos realizados	63
4.2 Estructura de la sección de resultados	64

4.3	Resultados en esclerosis múltiple	65
4.3.1	Resultados cuantitativos globales	65
4.3.2	Impacto de la dimensionalidad	65
4.3.3	Impacto de la plantilla anatómica	66
4.3.4	Comparación entre modelos	66
4.3.5	Análisis cualitativo	67
4.3.6	Resumen	70
4.4	Resultados en metástasis	70
4.4.1	Resultados cuantitativos globales	71
4.4.2	Impacto de la dimensionalidad	71
4.4.3	Comparación entre modelos	72
4.4.4	Análisis cualitativo	73
4.4.5	Resumen	76
4.5	Discusión comparativa	76
4.5.1	Influencia de la dimensionalidad	77
4.5.2	Comparación entre arquitecturas	77
4.5.3	Sobresegmentación e infrasegmentación	78
4.5.4	Coste computacional	78
4.5.5	Interpretación global	79
5	Conclusiones y líneas futuras	81
5.1	Conclusiones	81
5.2	Líneas futuras	82
A	Apéndice I: Manual de usuario	84
A.1	Requisitos del sistema	84
A.2	Creación de entornos virtuales	84
A.3	Descarga y Organización de los Datasets	87
A.4	Pipeline de Preprocesamiento	88
A.5	Diseño Experimental y Entrenamiento	89
A.6	Diseño Experimental y Ejecución	89
A.7	Visualización y Análisis de Resultados	91

Índice de figuras

1.1	Esquema representativo de tecnologías utilizadas y el hardware utilizado para realizar este TFG.	14
2.1	Arquitectura <i>U-Net</i> original propuesta por Ronneberger et al.[21], base de numerosos modelos posteriores de segmentación biomédica.	16
3.1	Ejemplo de cortes cerebrales en modalidades FLAIR y T1-weighted correspondientes a un mismo sujeto. La secuencia FLAIR resalta lesiones hiperintensas al suprimir la señal del líquido cefalorraquídeo, mientras que T1-weighted proporciona mayor detalle anatómico.	21
3.2	Plantilla anatómica cerebral MNI152 representada en las tres vistas ortogonales habituales (coronal, sagital y axial). Las líneas de referencia indican la correspondencia espacial entre los distintos planos.	29
3.3	Plantilla anatómica cerebral MRI225 mostrada en las tres vistas ortogonales.	29
3.4	Plantilla anatómica cerebral SRI24 representada en vistas coronal, sagital y axial. Este atlas proporciona una referencia anatómica de alta resolución para el alineamiento y análisis de imágenes de resonancia magnética cerebral.	30
3.5	Ejemplo de volumen de resonancia magnética antes del preprocesamiento. Se aprecian inhomogeneidades de intensidad y la presencia de estructuras extracraneales.	32
3.6	Ejemplo del mismo volumen tras el preprocesamiento completo, incluyendo corrección de inhomogeneidades, extracción de cráneo y normalización espacial.	32
3.7	Relación jerárquica entre inteligencia artificial, <i>machine learning</i> y <i>deep learning</i>	33
3.8	Esquema general de una red neuronal profunda con múltiples capas ocultas [12].	34
3.9	Representación del procesamiento volumétrico en redes neuronales convolucionales. El esquema ilustra la operación sobre tensores multidimensionales, habitual en CNNs aplicadas a imágenes médicas [12].	34
3.10	Ejemplo de segmentación semántica en resonancia magnética cerebral. La máscara de lesión se superpone a la imagen original y muestra la clasificación a nivel de <i>voxel</i> en las tres vistas ortogonales habituales.	35
3.11	Visión general del <i>pipeline</i> de segmentación automática propuesto por nnU-Net, que abarca desde el preprocesamiento de los datos hasta la inferencia y evaluación de resultados. Adaptado de Isensee et al. [10].	36
3.12	Esquema general del procesamiento mediante redes neuronales convolucionales en análisis de imágenes médicas. Adaptado de Litjens et al. [13].	38
3.13	Ejemplo de segmentación volumétrica mediante una red neuronal tridimensional (3D U-Net), donde se aprovecha el contexto espacial completo del volumen para producir una segmentación densa. Adaptado de Çiçek et al. [4].	40

3.14	Esquema del enfoque de entrenamiento en cascada empleado por nnU-Net. En una primera etapa, un modelo opera sobre datos a baja resolución para capturar el contexto global. En una segunda etapa, la segmentación obtenida se concatena con la imagen a resolución completa y es refinada por un segundo modelo. Adaptado de Isensee et al. [10].	41
3.15	Esquema conceptual del sistema de <i>trainers</i> en <i>nnU-Net v2</i> . El <i>trainer</i> actúa como un módulo que integra la arquitectura, la función de pérdida, la optimización y la estrategia de validación dentro del <i>pipeline</i> automatizado de entrenamiento.	45
3.16	Esquema conceptual de la arquitectura <i>U-Mamba</i> . La estructura <i>encoder-decoder</i> tipo <i>U-Net</i> se mantiene, incorporando bloques <i>Mamba</i> basados en <i>State Space Models</i> en determinadas etapas para enriquecer el modelado contextual global.	50
3.17	Esquema del proceso de preparación y organización de los datasets para su uso en <i>pipeline</i> de <i>nnU-Net v2</i>	54
3.18	Resumen del <i>pipeline</i> empleado con <i>nnU-Net v2</i> y <i>U-Mamba</i>	55
3.19	Comparación entre uno de los folds de entrenamiento del conjunto de Dataset550_Esclerosis_MNI152	57
4.1	Comparación del rendimiento medio (Dice) entre modelos 2D y 3D.	66
4.2	Comparación del rendimiento medio entre <i>nnU-Net v2</i> y <i>U-Mamba</i>	67
4.3	Caso fácil: lesiones extensas y bien delimitadas.	68
4.4	Caso intermedio: mayor coherencia espacial en <i>U-Mamba</i>	69
4.5	Caso difícil: fallos por bajo contraste y lesiones pequeñas.	70
4.6	Comparación del rendimiento medio (Dice) entre configuraciones 2D y 3D en metástasis.	72
4.7	Comparación del rendimiento medio entre <i>nnU-Net v2</i> y <i>U-Mamba</i> en metástasis cerebrales.	73
4.8	Caso sencillo: ambos modelos segmentan correctamente la lesión principal.	74
4.9	Fallo estructural de detección: <i>nnU-Net v2</i> no detecta la lesión mientras <i>U-Mamba</i> sí la segmenta correctamente.	75
4.10	Caso complejo: lesiones cercanas al límite de resolución con aparición de falsos positivos estructurados.	76
4.11	Tiempo medio de entrenamiento por época para cada arquitectura y dimensionalidad.	79
A.1	Configuración de variables de entorno necesarias para <i>nnU-Net v2</i> en Windows.	86

Índice de tablas

2.1	Resumen de arquitecturas relevantes en segmentación médica de imágenes de resonancia magnética.	17
3.1	Comparación de modalidades de resonancia magnética estructural utilizadas en la segmentación de lesiones cerebrales.	22
3.2	Resumen de las etapas de preprocesamiento aplicadas a los volúmenes de resonancia magnética.	31
3.3	Ejemplos de tipos de <i>trainers</i> disponibles en <i>nnU-Net v2</i> y sus características generales.	46
3.4	Comparación conceptual entre diferentes enfoques de segmentación médica.	52
3.5	Parámetros principales del entrenamiento en <i>nnU-Net v2</i> y su impacto. Los valores concretos dependen de los <i>plans</i> generados para cada dataset.	58
4.1	Resumen de entrenamientos realizados por dataset, plantilla, configuración y arquitectura.	64
4.2	Resultados para el dataset de esclerosis múltiple.	65
4.3	Resultados para el dataset de metástasis cerebrales.	71

Nomenclatura

nnU-Net v2 Framework automático de segmentación biomédica basado en U-Net

U-Mamba Arquitectura de segmentación basada en modelos de estado (SSM)

CNN *Convolutional Neural Networks* (redes neuronales convolucionales)

Dice Coeficiente de Dice (Dice Similarity Coefficient), mide el solapamiento entre segmentación y referencia

IoU Intersection over Union, medida alternativa de solapamiento

MRI *Magnetic Resonance Imaging* (imagen por resonancia magnética)

Plantilla anatómica Espacio de referencia para normalización (MNI152, MRI225, SRI24)

TE Tiempo de eco (*Echo Time*), intervalo entre la excitación del tejido y la medición de la señal

TR Tiempo de repetición (*Repetition Time*), intervalo entre pulsos consecutivos de excitación por radiofrecuencia

1

Introducción

1.1 Motivación

La detección de lesiones cerebrales continúa siendo uno de los desafíos más complejos en el ámbito de la neurorradiología. Incluso para profesionales altamente especializados, la identificación de estas anomalías puede resultar extremadamente difícil, ya que muchas veces se manifiestan con patrones mínimos, difusos o ambiguos en las imágenes de resonancia magnética. Además, existe una notable escasez de expertos con formación específica en este campo, lo que incrementa la carga asistencial y dificulta la realización de diagnósticos rápidos y precisos. Este tipo de decisiones clínicas suelen tomarse en contextos contrarreloj, ya que el diagnóstico temprano tiene un impacto directo en la calidad de vida del paciente y en la planificación de una posible intervención quirúrgica. Por tanto, disponer de herramientas de apoyo que permitan reducir la variabilidad diagnóstica y mejorar la sensibilidad en la detección es una necesidad creciente dentro del sistema sanitario.

En los últimos años, los avances en **inteligencia artificial** —y especialmente en algoritmos de *deep learning*— han transformado profundamente la forma en que se abordan problemas complejos en distintos dominios científicos y tecnológicos. Desde un punto de vista técnico, el *deep learning* se basa en arquitecturas de redes neuronales profundas capaces de modelar relaciones altamente no lineales y aprender representaciones jerárquicas de los datos. Cuando se someten a procesos de entrenamiento con grandes cantidades de información, estas redes logran captar patrones que resultan imperceptibles para los humanos o incluso para los métodos estadísticos tradicionales.

En el ámbito de la visión por computador y, en particular, del análisis de imágenes médicas, el *deep learning* ha demostrado un rendimiento especialmente destacado en tareas de segmentación automática. Arquitecturas basadas en redes neuronales convolucionales permiten explotar de forma eficiente la información espacial presente en los volúmenes de resonancia magnética, logrando delinear estructuras anatómicas y lesiones con un alto grado de precisión. Esta capacidad para **inteligencia artificial**, generalizar a nuevos casos y operar con grandes volúmenes de datos biomédicos ha convertido estos modelos en aliados fundamentales dentro del apoyo al diagnóstico clínico.

En este contexto, los sistemas de **Computer-Aided Diagnosis (CAD)** se han consolidado como herramientas clave para asistir al profesional médico en la interpretación de estudios de imagen. Su objetivo no es reemplazar al especialista, sino complementar su criterio mediante análisis cuantitativos, segmentaciones automáticas, clasificación de anomalías y priorización

de regiones sospechosas. Estos sistemas reducen la carga cognitiva durante la evaluación clínica y ayudan a minimizar errores derivados del cansancio, la presión asistencial o las limitaciones humanas en la percepción visual. En patologías derivadas de malformaciones o daños cerebrales, donde la lesión puede ser extremadamente sutil o incluso pasar desapercibida visualmente, los CAD basados en redes neuronales pueden proporcionar una ventaja diagnóstica decisiva y mejorar significativamente la planificación quirúrgica y el pronóstico del paciente.

Finalmente, este Trabajo de Fin de Grado también se motiva por la posibilidad de explorar nuevos horizontes dentro del campo del *deep learning* aplicado a imágenes médicas. Las arquitecturas modernas, como U-Net, nnU-Net y los modelos basados en *state space models*, abren oportunidades para desarrollar técnicas más robustas, generalizables y eficientes.

La evaluación de estas aproximaciones sobre conjuntos de datos reales de pacientes con lesiones cerebrales permite no solo analizar su rendimiento en escenarios clínicos complejos, sino también comprender en profundidad sus fortalezas y limitaciones. De este modo, el presente trabajo no solo aborda un problema de alta relevancia clínica como apoyo al profesional sanitario, sino que también contribuye a una exploración técnica rigurosa de modelos avanzados de segmentación aplicados a imágenes de resonancia magnética.

1.2 Objetivos

1.2.1 Objetivo general

El objetivo principal de este Trabajo de Fin de Grado es **analizar, entrenar y evaluar modelos de segmentación automática** de lesiones cerebrales en resonancia magnética (*Magnetic Resonance Imaging, MRI*) utilizando técnicas de *deep learning*, comparando el rendimiento en diferentes conjuntos de datos que han sido tratados previamente. Posteriormente, se determinarán las capacidades y limitaciones de estos algoritmos en escenarios clínicos reales aplicados a varias patologías, para comprobar si pueden llegar a ser utilizados como herramientas de ayuda por los profesionales.

1.2.2 Objetivos específicos

1. Recolección, tratamiento y organización de datos médicos.

Preparar adecuadamente dos conjuntos de datos públicos de neuroimagen, realizando su estandarización, análisis preliminar y conversión al formato requerido por los modelos de segmentación. Esto incluye trabajar con imágenes reales de pacientes y sus máscaras de lesión correspondientes.

2. Configuración de un entorno operativo para el entrenamiento de modelos de segmentación.

Construir un entorno funcional y reproducible que permita ejecutar procesos de entrenamiento intensivos, incluyendo la instalación de librerías, CUDA, PyTorch, *nnU-Net v2* y *U-Mamba*, así como la gestión de hardware y dependencias.

3. Entrenamiento de modelos basados en *deep learning*.

Entrenar los modelos seleccionados con los diferentes conjuntos de datos, aplicando configuraciones y plantillas específicas (*templates*) y garantizando un *pipeline* adecuado de preprocesamiento, aumentación, validación y monitorización.

4. Evaluación cuantitativa y cualitativa del rendimiento.

Analizar los resultados obtenidos por cada modelo mediante métricas estándar de segmentación (como el *Dice Similarity*

Coefficient o IoU) y mediante inspección visual de los volúmenes segmentados, valorando la calidad y coherencia clínica de las predicciones.

5. Comparación entre modelos y estudio de generalización.

Comparar el comportamiento de los algoritmos de segmentación en distintos dominios para estudiar su robustez, capacidad de generalización y sensibilidad a la variabilidad intra-paciente e inter-paciente.

6. Obtención de conclusiones y propuestas de mejora.

Extraer conclusiones sobre el rendimiento, ventajas y limitaciones de ambos enfoques, y proponer posibles líneas futuras de investigación o mejoras metodológicas aplicables a problemas de segmentación de lesiones cerebrales.

1.3 Estructura del documento

El contenido de este Trabajo de Fin de Grado se organiza en tres bloques principales, distribuidos de la siguiente manera:

• Introducción

- **Motivación:** se expone el contexto del proyecto y las motivaciones para llevar a cabo este estudio y que beneficios podría traer.
- **Objetivos:** se presenta el objetivo general y los objetivos específicos que guían el desarrollo del trabajo.
- **Estructura del documento:** se describe la organización del TFG (sección actual).
- **Tecnologías usadas:** se detallan los recursos hardware empleados para el entrenamiento y los componentes principales del software utilizado.

- **Estado del arte:** se aportan estudios recientes y avances relevantes en segmentación de neuroimagen mediante *deep learning*, que han ayudado al desarrollo actual de estas tecnologías.

• Metodología

- **Conjunto de datos:** descripción de los datasets utilizados, las lesiones incluidas, el formato de imágenes utilizado y la organización interna de los datasets.
- **Preprocesamiento:** transformaciones, scripts y plantillas utilizadas para preparar los datos antes del entrenamiento.
- **Algoritmos de entrenamiento:** explicación detallada de los modelos empleados (*nnU-Net v2* y *U-Mamba*), su funcionamiento y parámetros principales.
- **Diseño experimental:** descripción de las fases del entrenamiento, parámetros configurados, pruebas realizadas y criterios de evaluación.

- **Resultados:** presentación y análisis de los resultados obtenidos y comparación del rendimiento entre modelos.

• Conclusiones y líneas futuras

- **Conclusiones:** síntesis de los aprendizajes, aportaciones del trabajo y sus limitaciones.
- **Líneas futuras:** posibles ampliaciones y mejoras para continuar el proyecto.

1.4 Tecnologías usadas

El desarrollo de este Trabajo de Fin de Grado se ha llevado a cabo empleando un conjunto de **herramientas y entornos de software** que han permitido la preparación de los datos, el entrenamiento de los modelos de segmentación y la posterior evaluación de resultados.

En primer lugar, se utilizó el lenguaje de programación **Python**, ampliamente extendido en el ámbito del aprendizaje profundo y la visión por computador. La gestión de dependencias y entornos virtuales se realizó mediante las herramientas **pip** y **Conda**, lo que permitió mantener configuraciones independientes para cada uno de los modelos empleados.

Para la implementación de los algoritmos de segmentación se recurrió al marco de trabajo **PyTorch**, utilizando diferentes versiones según las necesidades de compatibilidad con CUDA. En el caso de **nnU-Net v2**, la instalación se llevó a cabo en el sistema operativo **Windows 11**, con **PyTorch 2.6.0** y soporte para **CUDA 12.6**. Por otro lado, el modelo **U-Mamba** se ejecutó en un entorno **Ubuntu 20.04** instalado a través de **Windows Subsystem for Linux (WSL)**, donde se configuró **PyTorch 2.0.1** con **CUDA 11.8**, en cumplimiento con los requisitos específicos del modelo.

Cabe señalar que la utilización de gestores de entornos distintos no responde a una decisión metodológica, sino a requisitos de compatibilidad del software. El framework **nnU-Net v2** dispone de soporte nativo para ejecución en Windows, permitiendo su instalación directa mediante *pip*. Por el contrario, **U-Mamba** depende de bibliotecas del sistema y versiones específicas de CUDA disponibles únicamente en entornos Linux, por lo que fue necesario emplear *Windows Subsystem for Linux (WSL)* y gestionar sus dependencias mediante *Conda*. Ambos modelos se ejecutaron sobre el mismo hardware físico, garantizando así la comparabilidad de los resultados experimentales.

Para la edición y ejecución de scripts auxiliares en Python se utilizó el entorno de desarrollo **Visual Studio Code**, que facilitó tanto la automatización de tareas de preprocesamiento y renombrado de archivos como la integración con los entornos virtuales previamente configurados. Asimismo, se empleó **Jupyter Lab** para la exploración de datos y el preprocesamiento de los datasets y las diferentes plantillas.

Además, se utilizó la herramienta en línea **Biolmage Suite Web** [20] para la visualización interactiva de imágenes médicas en formato NIfTI (**.nii.gz**), lo que resultó especialmente útil para la inspección de los volúmenes y máscaras de segmentación generados.

Finalmente, la ejecución de los modelos de aprendizaje profundo se realizó sobre un equipo con una **GPU NVIDIA RTX 4070 Ti SUPER de 16 GB**, un **procesador Intel Core i9-14900KF** y **64 GB de memoria RAM**, configuración que permitió reducir significativamente los tiempos de entrenamiento y facilitar el procesamiento de grandes volúmenes de datos.



Figura 1.1. Esquema representativo de tecnologías utilizadas y el hardware utilizado para realizar este TFG.

2

Estado del arte

La segmentación automática de estructuras y lesiones en imágenes de resonancia magnética (MRI) ha sido durante décadas un reto central en el ámbito de la neuroimagen, debido a la complejidad anatómica del cerebro y a la alta variabilidad interindividual de las patologías neurológicas. Tradicionalmente, esta tarea se realizaba mediante segmentación manual o semiautomática por expertos, apoyándose en técnicas basadas en umbrales de intensidad, crecimiento de regiones o modelos basados en atlas anatómicos. Sin embargo, estos métodos clásicos presentaban limitaciones importantes en términos de escalabilidad, reproducibilidad y dependencia del operador.

El auge del *machine learning* supuso un primer paso hacia la automatización del análisis de imágenes médicas, mediante el uso de clasificadores supervisados y descriptores diseñados manualmente. No obstante, estos enfoques requerían una selección cuidadosa de características y mostraban dificultades para generalizar a nuevos conjuntos de datos o protocolos de adquisición diferentes [13].

Con la irrupción del *deep learning*, y en particular de las redes neuronales convolucionales (*Convolutional Neural Networks*, CNN), se produjo un cambio de paradigma en la segmentación médica. Estas arquitecturas permiten aprender representaciones jerárquicas directamente a partir de los datos, eliminando la necesidad de seleccionar manualmente las características relevantes. En este contexto, uno de los trabajos más influyentes es la arquitectura *U-Net*, propuesta por Ronneberger et al. [21], diseñada específicamente para tareas de segmentación biomédica. Su estructura en forma de codificador-decodificador, junto con las conexiones de salto entre niveles, permitió combinar información semántica de alto nivel con detalles espaciales finos, logrando resultados sobresalientes incluso con conjuntos de datos relativamente pequeños.

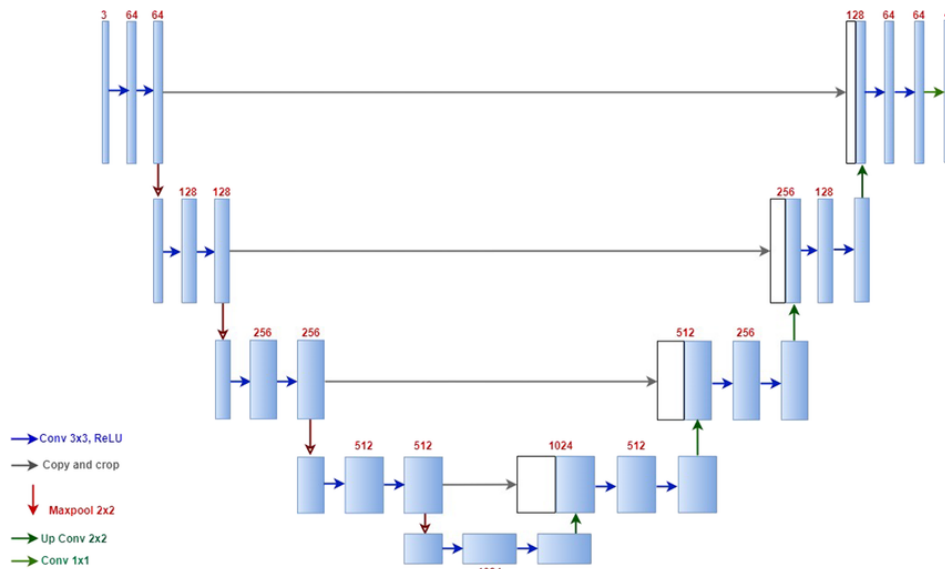


Figura 2.1. Arquitectura *U-Net* original propuesta por Ronneberger et al.[21], base de numerosos modelos posteriores de segmentación biomédica.

La Figura 2.1 muestra la arquitectura *U-Net* original propuesta por Ronneberger et al. [21], considerada uno de los hitos fundamentales en el desarrollo de modelos de segmentación biomédica basados en redes neuronales convolucionales. Esta arquitectura presenta una estructura simétrica de tipo codificador–decodificador, en la que el camino de contracción se encarga de extraer representaciones jerárquicas de alto nivel mediante operaciones sucesivas de convolución y reducción de resolución espacial.

El camino de expansión permite recuperar progresivamente la resolución original de la imagen, combinando la información semántica profunda con detalles espaciales finos procedentes de las capas iniciales a través de conexiones de salto entre niveles homólogos. Este mecanismo resulta especialmente relevante en la segmentación de estructuras anatómicas complejas, donde la precisión espacial es un factor crítico.

El diseño de *U-Net* ha demostrado una notable eficacia incluso en escenarios con conjuntos de datos relativamente reducidos, lo que ha favorecido su amplia adopción en el ámbito de la imagen médica. A partir de esta arquitectura se han desarrollado numerosas variantes posteriores, tanto en dos como en tres dimensiones, que han consolidado a *U-Net* como un estándar de referencia en tareas de segmentación automática de imágenes de resonancia magnética.

En el caso particular de la resonancia magnética cerebral, revisiones específicas han analizado el impacto del *deep learning* en segmentación, así como los principales desafíos asociados —como la variabilidad entre escáneres y protocolos, la anisotropía (diferencia de resolución espacial entre los ejes del volumen, típica en MRI donde el grosor de corte es mayor que el tamaño del píxel en el plano) y el desbalance de clases— y las líneas futuras de investigación [2].

A partir de *U-Net* se han desarrollado numerosas variantes orientadas a mejorar su rendimiento en escenarios clínicos específicos. Entre ellas destacan *3D U-Net*, que extiende el procesamiento a volúmenes tridimensionales completos; *V-Net*, optimizada para segmentación volumétrica mediante convoluciones 3D; y modelos con mecanismos de atención que refuerzan regiones relevantes de la imagen. Estas arquitecturas han demostrado su eficacia en tareas como la segmentación de tumores

cerebrales, lesiones de esclerosis múltiple, infartos isquémicos y anomalías vasculares [4, 17].

En el ámbito de la resonancia magnética cerebral, múltiples estudios han evidenciado la utilidad de las CNN para la detección y segmentación de lesiones sutiles, especialmente en secuencias como FLAIR y T1. En patologías como la esclerosis múltiple, donde las lesiones presentan una gran variabilidad morfológica y de intensidad, la segmentación automática se ha convertido en una herramienta clave tanto para el diagnóstico como para el seguimiento longitudinal de la enfermedad [5].

Tabla 2.1. Resumen de arquitecturas relevantes en segmentación médica de imágenes de resonancia magnética.

Autor	Año	Arquitectura	Modalidad	Aplicación
Ronneberger et al. [21]	2015	<i>U-Net</i>	MRI, Microscopia	Segmentación biomédica
Çiçek et al. [4]	2016	3D <i>U-Net</i>	MRI	Segmentación volumétrica
Milletari et al. [17]	2016	V-Net	MRI	Segmentación 3D
Litjens et al. [13]	2017	Revisión CNN	MRI, CT	Visión médica
Isensee et al. [10]	2021	nnU-Net	Multimodal	Segmentación automática
Gu et al. [7]	2023	<i>U-Mamba</i>	MRI	Segmentación con SSM

En este contexto surge *nnU-Net*, un framework que propone un enfoque radicalmente diferente: la *configuración automática* de *pipelines* de segmentación médica [10]. En lugar de diseñar manualmente la arquitectura, los hiperparámetros y las estrategias de entrenamiento, *nnU-Net* adapta automáticamente estos elementos en función de las propiedades del conjunto de datos, como el tamaño del volumen, el espaciado de *voxel* o el número de muestras disponibles. Este enfoque ha demostrado una capacidad de generalización excepcional, logrando resultados de referencia en numerosos desafíos internacionales de segmentación médica.

La importancia de *nnU-Net* se ha consolidado a través de su participación en competiciones organizadas en plataformas como *Grand Challenge*, donde se evalúan algoritmos bajo condiciones estandarizadas y reproducibles [6]. Estos desafíos han permitido comparar de forma objetiva diferentes métodos y han contribuido significativamente al avance del estado del arte en segmentación biomédica.

Más recientemente, la comunidad científica ha comenzado a explorar arquitecturas alternativas a las CNN tradicionales, como los *Transformers*, capaces de modelar dependencias globales mediante mecanismos de autoatención. Aunque estos modelos han mostrado un gran potencial, su elevado coste computacional y la necesidad de grandes volúmenes de datos han limitado su adopción directa en entornos clínicos.

Como respuesta a estas limitaciones, han emergido enfoques híbridos que combinan convoluciones con modelos más eficientes de modelado de dependencias a largo alcance, como los *State Space Models* (SSM). En este contexto, destaca *U-Mamba*, una arquitectura que integra modelos de espacio de estados para capturar información contextual global de forma eficiente, manteniendo un coste computacional reducido. Esta aproximación resulta especialmente atractiva para imágenes médicas tridimensionales de alta resolución, donde el contexto global juega un papel fundamental en la correcta delimitación de las lesiones [7].

Desde una perspectiva clínica, el desarrollo de sistemas de apoyo a la decisión basados en inteligencia artificial, conocidos

como *Computer-Aided Diagnosis* (CAD), ha adquirido un papel creciente. Estas herramientas no pretenden sustituir al profesional médico, sino actuar como apoyo en la detección de patrones sutiles, reducir la variabilidad interobservador y mejorar la eficiencia del flujo de trabajo clínico. La segmentación automática de lesiones cerebrales constituye uno de los pilares fundamentales para el desarrollo de este tipo de sistemas.

En conjunto, el estado actual de la investigación muestra una clara tendencia hacia modelos más robustos, adaptativos y generalizables, capaces de trabajar con datasets heterogéneos procedentes de distintos centros y protocolos de adquisición. En este Trabajo de Fin de Grado se exploran y comparan enfoques consolidados como *nnU-Net v2* con arquitecturas más recientes como *U-Mamba*, con el objetivo de evaluar su rendimiento en la segmentación de lesiones cerebrales en distintos contextos patológicos.

3

Metodología

3.1 Conjunto de datos

Para el desarrollo de este trabajo se emplearon dos conjuntos de datos públicos con imágenes de resonancia magnética (MRI), ambos orientados al estudio de patologías neurológicas con presencia de lesiones cerebrales.

3.1.1 Formato de Almacenamiento y Modalidades de Imagen Estructural

Para el procesamiento y análisis de las imágenes médicas en este trabajo, la organización de los datos se fundamenta en dos pilares: el estándar de archivo **NIfTI** para la información volumétrica y el estándar **BIDS** para la organización y gestión de metadatos asociados a cada adquisición.

Formato de Datos: NIfTI

El formato **NIfTI** (*Neuroimaging Informatics Technology Initiative*) es el estándar *de facto* en neuroimagen para almacenar datos volumétricos obtenidos principalmente mediante resonancia magnética (MRI) y tomografía computarizada [18, 1]. Su principal ventaja reside en encapsular en un único archivo tanto la información de intensidades como los metadatos geométricos necesarios para interpretar correctamente el volumen en el espacio.

En este trabajo se emplean archivos con extensión `.nii.gz` (una versión comprimida sin pérdida del formato `.nii`) tanto para las imágenes originales (canales de entrada al modelo) como para las máscaras de segmentación (etiquetas o *ground truth*).

Cada archivo NIfTI contiene dos componentes esenciales:

- **Volumen 3D**: compuesto por **voxels** (*volumetric pixels*), que representan la intensidad de señal medida en cada posición del espacio.
- **Encabezado (*header*)**: almacena metadatos críticos que definen la geometría y el sistema de coordenadas del volumen, permitiendo su correcta interpretación espacial (p. ej., matrices *qform/sform* o matrices afines) [1]. La información relevante incluye:
 - Dimensiones del volumen y resolución espacial (*voxel spacing*).

- Orientación anatómica y transformación de coordenadas imagen→espacio (alineación espacial).
- Tipo de dato y rango de intensidades.

Estas propiedades son especialmente importantes en resonancia magnética, donde pueden existir variaciones notables de resolución y orientación entre centros y protocolos. Por ello, en *pipelines* como *nnU-Net v2* es habitual aplicar estrategias de remuestreo y normalización de intensidades para homogeneizar la entrada del modelo (detallado en secciones posteriores).

Esta estructura facilita el procesamiento automático, la carga eficiente de datos en memoria y la interoperabilidad con librerías y herramientas de neuroimagen (p. ej., `nibabel`) y con frameworks de segmentación como *nnU-Net v2*.

Modalidades de Resonancia Magnética (MRI)

Las imágenes estructurales de resonancia magnética (MRI) se diferencian en función de los parámetros de adquisición (p. ej., tiempo de repetición - TR y tiempo de eco - TE) y de las secuencias de pulsos empleadas, lo que determina el contraste entre tejidos. Las modalidades más relevantes para la segmentación de lesiones focales en neuroimagen son:

- **T1-weighted (T1w)**: modalidad base para caracterización morfológica. La sustancia blanca suele aparecer más brillante que la sustancia gris, mientras que el líquido cefalorraquídeo (LCR) aparece oscuro. Es fundamental para la **localización precisa** y el análisis anatómico de estructuras.
- **T2-weighted (T2w)**: contraste opuesto al T1w; el LCR y tejidos con alto contenido de agua (p. ej., edema) aparecen brillantes, mientras que la sustancia blanca es relativamente más oscura.
- **FLAIR (Fluid-Attenuated Inversion Recovery)**: secuencia T2 modificada que **suprime la señal del LCR**, haciendo que el LCR aparezca oscuro y resaltando lesiones hiperintensas (brillantes). Es especialmente útil para visualizar lesiones focales y procesos inflamatorios/desmielinizantes.

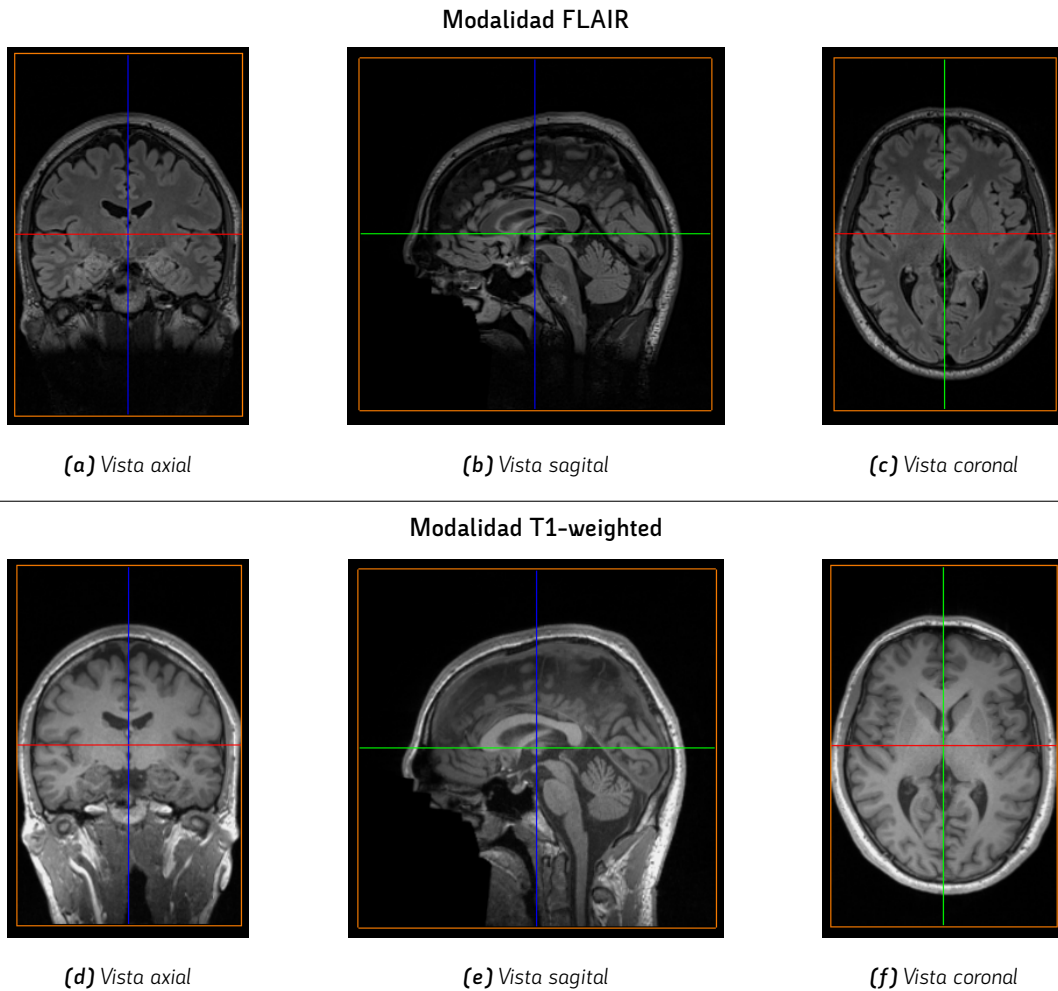


Figura 3.1. Ejemplo de cortes cerebrales en modalidades FLAIR y T1-weighted correspondientes a un mismo sujeto. La secuencia FLAIR resalta lesiones hiperintensas al suprimir la señal del líquido cefalorraquídeo, mientras que T1-weighted proporciona mayor detalle anatómico.

En la Figura 3.1 se muestran ejemplos de cortes cerebrales en las modalidades FLAIR y T1-weighted correspondientes a un mismo sujeto, ilustrando las diferencias de contraste entre ambas secuencias y su complementariedad para la segmentación de lesiones. Las líneas de referencia superpuestas indican la intersección de los planos axial, sagital y coronal, permitiendo visualizar la correspondencia espacial entre vistas del mismo volumen.

En el presente trabajo, las modalidades **T1w** y **FLAIR** fueron seleccionadas y empleadas conjuntamente (*multicanaal*) para el entrenamiento y la evaluación de los modelos. Esta combinación resulta informativa para tareas de segmentación de lesiones al aportar simultáneamente alto detalle anatómico (T1w) y elevada sensibilidad a alteraciones de señal asociadas a patología (FLAIR).

La Tabla 3.1 resume las principales características de las modalidades estructurales más relevantes para la segmentación de lesiones cerebrales.

Tabla 3.1. Comparación de modalidades de resonancia magnética estructural utilizadas en la segmentación de lesiones cerebrales.

Modalidad	Contraste principal	Características destacadas	Aplicación en segmentación
T1-weighted (T1w)	Alta resolución anatómica	Sustancia blanca brillante, LCR oscuro, excelente definición estructural	Localización precisa de estructuras y delimitación anatómica
T2-weighted (T2w)	Sensibilidad al contenido de agua	LCR y tejidos patológicos brillantes	Detección general de edemas y lesiones
FLAIR	T2 con supresión de LCR	Lesiones hiperintensas destacadas sobre fondo oscuro	Detección de lesiones focales y procesos inflamatorios/desmielinizantes

3.1.2 Dataset de esclerosis múltiple

El primer conjunto de datos empleado corresponde al **MSLesSeg** (*Multiple Sclerosis Lesion Segmentation dataset*) [8], una base de datos diseñada específicamente para el estudio y la segmentación de lesiones asociadas a la **esclerosis múltiple (EM)**. Publicado en 2024, este conjunto de datos constituye una referencia reciente para la evaluación y comparación de modelos de segmentación automática aplicados a patologías desmielinizantes del sistema nervioso central.

La esclerosis múltiple es una enfermedad neurológica crónica de origen autoinmune caracterizada por la aparición de lesiones desmielinizantes en el cerebro y la médula espinal. Estas lesiones se producen como consecuencia del daño a la mielina, la sustancia que recubre y protege las fibras nerviosas, lo que altera la transmisión de los impulsos neuronales. Desde el punto de vista clínico, la localización, el número y la evolución temporal de las lesiones constituyen indicadores clave para el diagnóstico, la clasificación de la enfermedad y el seguimiento de la progresión del paciente.

El dataset **MSLesSeg** contiene imágenes de resonancia magnética estructural de pacientes diagnosticados de EM, junto con sus correspondientes máscaras de segmentación manual generadas por expertos. Las modalidades incluidas son principalmente **T1-weighted (T1w)** y **FLAIR**, ambas ampliamente utilizadas en la práctica clínica. En particular, las secuencias FLAIR desempeñan un papel fundamental en la detección de lesiones de esclerosis múltiple, ya que permiten suprimir la señal del líquido cefalorraquídeo y realzar las áreas de inflamación o daño en la sustancia blanca.

Desde el punto de vista del procesamiento de imagen, las lesiones de esclerosis múltiple presentan una elevada variabilidad en términos de tamaño, forma, localización e intensidad. Pueden aparecer como múltiples focos pequeños dispersos o como lesiones más extensas, y su contraste con el tejido sano puede variar notablemente entre pacientes y sesiones de adquisición. Esta heterogeneidad convierte la segmentación automática en una tarea especialmente compleja y exigente para los modelos de aprendizaje profundo.

El repositorio **MSLesSeg** proporciona no solo un conjunto de datos preprocesado para tareas de segmentación de lesiones de esclerosis múltiple, sino también los recursos necesarios para reproducir y analizar en detalle el flujo completo de preparación de los datos. Esta organización permite a los investigadores trabajar tanto con imágenes listas para el entrenamiento como con

los datos originales, facilitando la comparación de diferentes estrategias de preprocesamiento.

El repositorio se estructura en varios archivos comprimidos principales. En primer lugar, el archivo `MSLesSeg Dataset.zip` contiene el conjunto de datos ya preprocesado, incluyendo los volúmenes de resonancia magnética y sus correspondientes máscaras de segmentación de lesiones. Este conjunto está diseñado para su uso directo en tareas de entrenamiento y evaluación de modelos de segmentación supervisada, garantizando la coherencia espacial entre imágenes y etiquetas.

Por otro lado, el archivo `MSLesSeg_RAW.zip` incluye las imágenes de resonancia magnética originales, sin ningún tipo de procesamiento previo. Estos datos brutos permiten a los investigadores aplicar *pipelines* alternativos de normalización, registro o corrección de artefactos, lo que resulta especialmente relevante en el contexto de estudios comparativos o experimentales.

El repositorio incorpora además el archivo `MSLesSeg-2024-main.zip`, que contiene el código empleado por los autores para llevar a cabo el proceso de preprocesamiento aplicado a los datos originales. Este código proporciona una referencia explícita del flujo seguido para generar el dataset final, reforzando la reproducibilidad de los experimentos y permitiendo replicar o modificar cada una de las etapas del preprocesado.

Finalmente, se incluyen los archivos `.mat` agrupados en `mat_files.zip`, los cuales almacenan los parámetros de transformación utilizados durante el registro y la alineación de las imágenes. Estos archivos permiten mapear de forma precisa las máscaras de segmentación generadas sobre los datos preprocesados de vuelta a las imágenes originales, garantizando la correspondencia espacial entre ambos dominios.

Selección de un único estudio por paciente. Dado que algunos pacientes disponen de varias adquisiciones longitudinales (p. ej., diferentes tiempos o sesiones), en este trabajo se seleccionó de forma sistemática **una única adquisición por paciente** con el objetivo de evitar dependencias entre muestras del mismo sujeto y reducir el riesgo de introducir sesgo en la evaluación. En el conjunto de entrenamiento se utilizó la adquisición T1 (primer tiempo) de cada paciente, tomando las modalidades FLAIR y T1 junto con su máscara asociada. Esta decisión permite que cada caso represente un sujeto independiente, haciendo más fiable la comparación entre configuraciones y modelos.

En conjunto, esta estructura dota al dataset `MSLesSeg` de una elevada flexibilidad y transparencia, ya que permite trabajar tanto con datos listos para el entrenamiento de modelos de *deep learning* como con imágenes crudas y transformaciones intermedias. Esta característica resulta especialmente valiosa para evaluar la robustez de algoritmos de segmentación automática, como *nnU-Net v2* y *U-Mamba*, frente a diferentes configuraciones de preprocesamiento.

Resumen de uso en este trabajo. A partir de la estructura del repositorio, los casos se reorganizaron al formato *nnU-Net v2* manteniendo la correspondencia paciente–identificador (p. ej., `P3` → `CEREBRO_00003`). Las imágenes de entrenamiento se almacenaron en `imagesTr` y sus máscaras en `labelsTr`, mientras que los casos de test se ubicaron en `imagesTs` y `labelsTs`. Los datos se dividieron para tener 60 imágenes de entrenamiento (*train*) y 15 de prueba (*test*).

3.1.3 Dataset de tumores cerebrales metastásicos

El dataset de metástasis cerebrales empleado proviene de MetaS-RD, descrito en [19]. Los archivos utilizados en este trabajo se descargaron desde el repositorio web indicado en [15]. A diferencia de los tumores primarios del sistema nervioso central, las metástasis cerebrales son lesiones secundarias originadas por la diseminación hematológica de un tumor primario (frecuentemente de pulmón, mama o melanoma) hacia el parénquima cerebral. Clínicamente, su detección y delimitación precisa

es relevante tanto para la planificación terapéutica (radiocirugía, resección) como para el seguimiento longitudinal.

El conjunto contiene estudios de resonancia magnética en formato `.nii/.nii.gz` asociados a **segmentaciones manuales** proporcionadas en forma de máscaras (`msk`). En el repositorio, cada paciente se organiza en una carpeta propia, dentro de la cual pueden existir **múltiples adquisiciones** correspondientes a diferentes tiempos o seguimientos. Esta variabilidad longitudinal es útil para estudios de progresión, pero en el contexto de este TFG (segmentación supervisada y comparación de arquitecturas bajo condiciones controladas) puede introducir dependencia entre muestras si se incluyen varios estudios del mismo sujeto.

Selección de un único estudio por paciente. Para homogeneizar el conjunto y asegurar independencia entre casos, se seleccionó también **una única adquisición por paciente**. Dado que todas las resonancias disponibles eran con contraste, el criterio de selección se definió como:

- Priorizar adquisiciones marcadas como **alta resolución (HR)** cuando estuvieran disponibles.
- En segundo lugar, priorizar estudios cuya nomenclatura contuviera **GD/Gd** (cuando existieran variantes).
- En ausencia de los criterios anteriores, seleccionar el **estudio más reciente** disponible (según la fecha codificada en el nombre del fichero).

A cada estudio seleccionado se le asoció su máscara correspondiente, garantizando que imagen y label pertenecen al mismo tiempo de adquisición. Posteriormente, se reorganizaron los datos al formato requerido por *nnU-Net v2* y se homogeneizó la nomenclatura a la forma `CEREBRO_XXXXX_0001.nii.gz` para la imagen (T1) y `CEREBRO_XXXXX.nii.gz` para la máscara.

Binarización de máscaras. Las segmentaciones originales incluyen etiquetas que pueden contener más de una clase (p. ej., diferentes subregiones tumorales). Sin embargo, en este trabajo se definió un problema **binario** (*tumor vs. fondo*) para facilitar la comparación entre datasets y reducir la complejidad de la tarea. En consecuencia, todas las etiquetas distintas de 0 se unificaron como lesión:

$$0 \rightarrow \text{fondo}, \quad \{1, 2, \dots\} \rightarrow 1$$

Esta transformación permite además robustecer el entrenamiento frente a posibles inconsistencias puntuales de etiquetado en algunas máscaras.

Resumen de uso en este trabajo. Este dataset aporta un escenario complementario al de esclerosis múltiple: lesiones más focales, de realce con contraste y con una distribución espacial distinta. Su inclusión permite evaluar la robustez de *nnU-Net v2* y *U-Mamba* en una patología tumoral, manteniendo un protocolo experimental comparable y controlado. De igual forma los datos se dividieron para tener 60 imágenes de entrenamiento (*train*) y 15 de prueba (*test*).

3.1.4 Estructura de carpetas

En este trabajo fue necesario reorganizar ambos conjuntos de datos para adaptarlos al formato estándar requerido por el framework *nnU-Net v2*. Este formato define una estructura de directorios estricta, que permite al framework localizar de forma automática las imágenes, las etiquetas y la configuración asociada a cada dataset. Adicionalmente, se añade una carpeta para las futuras inferencias con los conjuntos de test.

La estructura general queda organizada de la siguiente forma para cada uno de los datasets que se utilizan:

```

nnUNet_inference/XXX/
nnUNet_preprocessed/DatasetXXX/
nnUNet_results/DatasetXXX/
nnUNet_raw/DatasetXXX/
  imagesTr/      → Imágenes de entrenamiento (.nii.gz)
  labelsTr/      → Máscaras de entrenamiento (.nii.gz)
  imagesTs/      → Imágenes de test (.nii.gz)
  labelsTs/      → Máscaras de test (opcional)
  dataset.json   → Archivo de configuración del dataset

```

A continuación, se describe el contenido de cada uno de estos elementos:

- **nnUNet_raw/**: contiene los datasets en su forma inicial, organizados conforme al formato requerido por nnU-Net. En este directorio se almacenan las carpetas `DatasetXXX`, que incluyen las imágenes originales, las máscaras de segmentación y el archivo `dataset.json`. Este directorio constituye el punto de partida del *pipeline*.
- **nnUNet_preprocessed/**: almacena los datos tras la fase de planificación y preprocesamiento automático realizada por nnU-Net. En esta etapa se generan versiones reescaladas, normalizadas y adaptadas de las imágenes, así como los planes de entrenamiento (*plans*) que definen la arquitectura y los parámetros a emplear.
- **nnUNet_results/**: contiene los resultados derivados del entrenamiento de los modelos. Para cada dataset y configuración se generan subdirectorios que almacenan los pesos de los modelos entrenados, los registros de entrenamiento, las métricas obtenidas y los ficheros necesarios para la inferencia y el postprocesamiento.
- **nnUNet_inference/**: directorio utilizado para almacenar las predicciones generadas por los modelos entrenados sobre nuevos datos o conjuntos de test. En este trabajo se emplea para guardar las segmentaciones inferidas, así como las versiones postprocesadas empleadas posteriormente en la evaluación cuantitativa.
- **imagesTr/**: contiene todas las imágenes destinadas al entrenamiento del modelo. Cada caso debe tener un nombre único y, en caso de disponer de varias modalidades (por ejemplo, FLAIR y T1), el archivo debe incluir un sufijo numérico `_0000.nii.gz`, `_0001.nii.gz`, etc.
- **labelsTr/**: incluye las máscaras de segmentación correspondientes a los casos presentes en `imagesTr`. El nombre del archivo debe coincidir exactamente, sin sufijos de modalidad. Por ejemplo:

```

imagesTr/CEREBRO_00001_0000.nii.gz
labelsTr/CEREBRO_00001.nii.gz

```

- **imagesTs/**: contiene las imágenes sobre las que se hará inferencia. Estas imágenes no necesitan etiquetas durante el entrenamiento.

- **labelsTs/** (opcional): solo se incluye cuando se dispone de etiquetas para el conjunto de test, normalmente con fines de evaluación externa. En este trabajo, se ha preocupado por que todas las imágenes de test tuvieran su máscara correspondiente.
- **dataset.json**: archivo fundamental que describe el dataset, incluyendo:
 - Modalidades utilizadas (p. ej., FLAIR como canal 0).
 - Número de clases y su significado.
 - Relación entre cada caso y sus archivos de imagen.
 - Información adicional como el nombre del dataset, la versión o posibles metadatos.

Ejemplo de estructura (basado en nnU-Net)

A modo de referencia, el formato esperado por **nnU-Net** para un dataset con modalidad única es el siguiente:

```
Dataset501_Esclerosis/
  imagesTr/
    CEREBRO_00001_0000.nii.gz
    CEREBRO_00001_0001.nii.gz
    CEREBRO_00002_0000.nii.gz
    CEREBRO_00002_0001.nii.gz
    ...
  labelsTr/
    CEREBRO_00001.nii.gz
    CEREBRO_00002.nii.gz
    ...
  imagesTs/
    CEREBRO_00098_0000.nii.gz
    CEREBRO_00099_0000.nii.gz
    ...
  dataset.json
```

Ejemplo de archivo dataset.json

El archivo `dataset.json` sigue estrictamente el formato definido por **nnU-Net v2**. A continuación se muestra un ejemplo adaptado al caso de un dataset de resonancias FLAIR con una única clase de lesión:

```
{
  "channel_names": {
    "0": "FLAIR"
  },

```

```

"labels": {
  "background": 0,
  "fcd": 1
},

"numTraining": 50,
"file_ending": ".nii.gz",
"name": "Dataset501_Epilepsia"
}

```

Este archivo permite al framework interpretar correctamente:

- el número de modalidades,
- la ubicación de las imágenes y etiquetas,
- el número de casos de entrenamientos y su extensión.

Por último, en este proyecto se definieron 3 datasets diferentes para cada conjunto de datos, para realizar los entrenamientos con las diferentes **plantillas** (utilizadas y explicadas a continuación en el preprocesamiento previo al algoritmo [3.2.2](#)):

- **Primer dataset (Esclerosis múltiple):**
 - Dataset450_Esclerosis_MRI225
 - Dataset550_Esclerosis_MNI152
 - Dataset650_Esclerosis_SRI24
- **Segundo dataset (Metástasis):**
 - Dataset810_metastasis_MRI225
 - Dataset820_metastasis_MNI152
 - Dataset830_metastasis_SRI24

Cada uno se almacenó en el directorio raíz de datos con la nomenclatura **DatasetXXX_Nombre**, manteniendo así separados los experimentos y resultados asociados a cada conjunto de datos.

3.2 Preprocesamiento

3.2.1 Motivación del preprocesamiento

Las imágenes de resonancia magnética presentan una elevada variabilidad debida a factores como las diferencias entre escáneres, protocolos de adquisición, resoluciones espaciales y condiciones del paciente. Esta heterogeneidad puede dificultar el

aprendizaje de modelos de segmentación basados en *deep learning*, especialmente cuando se trabaja con conjuntos de datos procedentes de distintas fuentes.

El objetivo del preprocesamiento es reducir esta variabilidad no relacionada con la patología, preservando al mismo tiempo la información anatómica relevante. Para ello, se aplicaron distintas técnicas de normalización espacial e intensidad, siendo especialmente relevante el uso de plantillas anatómicas de referencia, que permiten alinear los volúmenes cerebrales en un espacio común.

Desde el punto de vista computacional, esta variabilidad no controlada puede provocar que el modelo aprenda patrones espurios asociados a diferencias de adquisición en lugar de características anatómicas o patológicas relevantes. En redes neuronales profundas, este fenómeno se traduce en una peor convergencia durante el entrenamiento y una disminución de la capacidad de generalización a nuevos datos.

En el caso del framework *nnU-Net v2*, aunque muchos pasos de preprocesamiento están integrados de forma automática, la calidad y homogeneidad de los datos de entrada sigue siendo un factor determinante. Por este motivo, en este trabajo se aplicó un preprocesamiento previo cuidadosamente diseñado, con el objetivo de proporcionar al modelo datos espacial y radiométricamente coherentes antes de la fase de entrenamiento.

3.2.2 Normalización espacial mediante plantillas anatómicas

La normalización espacial consiste en registrar las imágenes de resonancia magnética de cada sujeto a un espacio anatómico de referencia común, definido por una plantilla o **atlas cerebral**.

Para ello, en este trabajo se empleó un registro no lineal de las imágenes a la plantilla de referencia, empleando transformaciones difeomórficas (transformaciones suaves, continuas y reversibles que deforman la imagen sin generar pliegues ni discontinuidades anatómicas). Este tipo de registro permite modelar deformaciones anatómicas complejas y lograr una alineación más precisa entre estructuras cerebrales, especialmente relevante en patologías con alta variabilidad espacial como la esclerosis múltiple o las lesiones metastásicas.

Este proceso permite reducir las diferencias anatómicas globales entre sujetos y facilita la comparación y el aprendizaje de patrones comunes por parte de los modelos de segmentación.

En este trabajo se emplearon distintas plantillas anatómicas como referencia durante el preprocesamiento, concretamente **MNI152**, **MRI225** y **SRI24**. Cada una de ellas define un espacio estándar diferente, lo que permite analizar el impacto de la elección de la plantilla sobre el rendimiento de los modelos entrenados.

Desde el punto de vista del aprendizaje profundo, la normalización espacial reduce la carga que debe asumir el modelo para aprender invariancias geométricas, permitiendo que la red concentre su capacidad en la identificación de patrones patológicos locales en lugar de compensar diferencias anatómicas globales entre sujetos.

3.2.2.1 Plantilla MNI152

La plantilla **MNI152** es uno de los espacios de referencia más utilizados en neuroimagen. Fue construida a partir del promedio de 152 cerebros humanos alineados al espacio del *Montreal Neurological Institute* (MNI). Su uso está ampliamente extendido en estudios clínicos y de investigación, lo que la convierte en un estándar de facto para la normalización espacial.

El registro de las imágenes a la plantilla MNI152 permite una alineación anatómica global robusta, aunque puede introducir

cierto suavizado de estructuras finas, especialmente en lesiones pequeñas o de morfología irregular.

Por este motivo, aunque MNI152 proporciona una alineación anatómica robusta a gran escala, su uso puede resultar subóptimo en tareas donde la detección de lesiones pequeñas y sutiles es crítica.

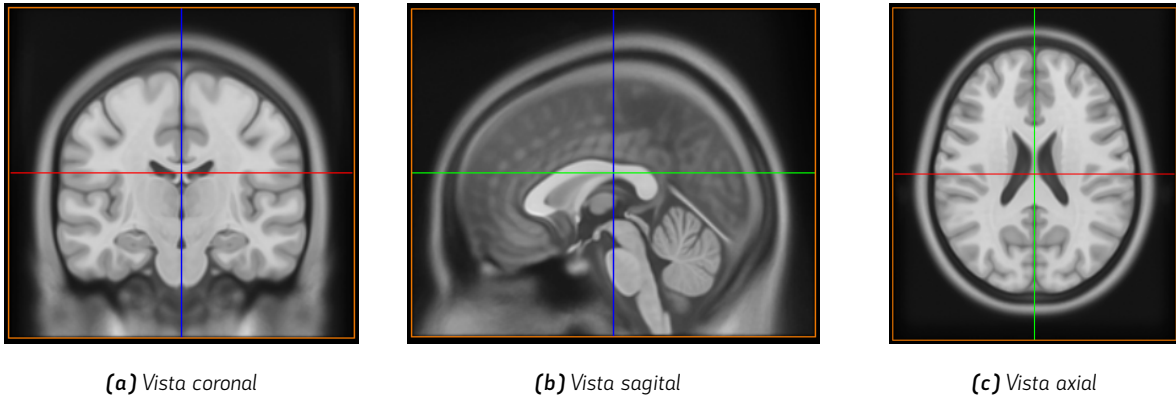


Figura 3.2. Plantilla anatómica cerebral **MNI152** representada en las tres vistas ortogonales habituales (coronal, sagital y axial). Las líneas de referencia indican la correspondencia espacial entre los distintos planos.

3.2.2.2 Plantilla MRI225

La plantilla **MRI225** se construyó a partir de un conjunto más amplio de imágenes anatómicas, lo que proporciona una representación estadística diferente del cerebro humano. En comparación con MNI152, esta plantilla presenta características anatómicas que pueden adaptarse mejor a determinados conjuntos de datos, especialmente aquellos con mayor variabilidad intersujeto.

En este trabajo, la plantilla MRI225 se utilizó como una alternativa a MNI152 con el objetivo de evaluar si una referencia anatómica distinta influye en la capacidad del modelo para aprender patrones de segmentación más precisos, particularmente en el caso de lesiones difusas o múltiples.

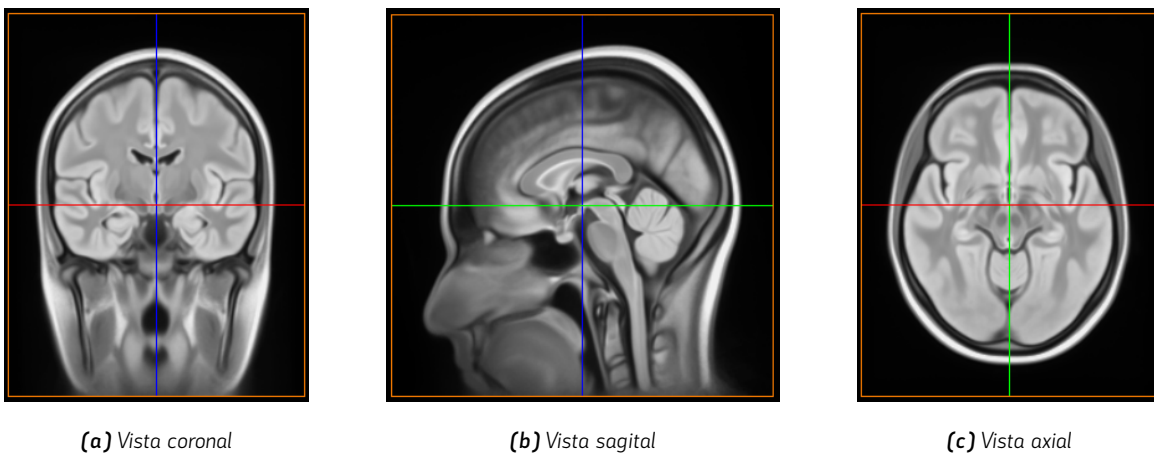


Figura 3.3. Plantilla anatómica cerebral **MRI225** mostrada en las tres vistas ortogonales.

3.2.2.3 Plantilla SR124

La plantilla **SR124** es un atlas anatómico de alta resolución diseñado específicamente para estudios de neuroimagen estructural. Incluye información anatómica detallada y suele emplearse en tareas que requieren una mayor precisión espacial.

Debido a la naturaleza dispersa y heterogénea de las lesiones de esclerosis múltiple, el uso de la plantilla SR124 resulta especialmente interesante, ya que puede facilitar una mejor alineación de las regiones de sustancia blanca donde suelen localizarse estas lesiones.

Esta mayor fidelidad anatómica puede resultar especialmente beneficiosa en modelos de segmentación que dependen de una localización precisa de las lesiones en relación con estructuras cerebrales específicas.

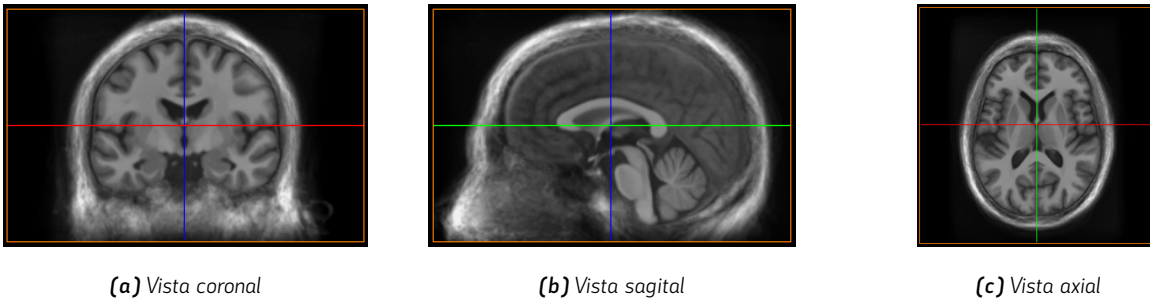


Figura 3.4. Plantilla anatómica cerebral **SR124** representada en vistas coronal, sagital y axial. Este atlas proporciona una referencia anatómica de alta resolución para el alineamiento y análisis de imágenes de resonancia magnética cerebral.

3.2.3 Corrección de inhomogeneidades de intensidad (N4 Bias Field Correction)

Las imágenes de resonancia magnética suelen presentar uniformidades de intensidad debidas a imperfecciones del campo magnético del escáner. Estas variaciones, conocidas como *bias field*, pueden dificultar la correcta interpretación de las intensidades de los voxels.

Para mitigar este efecto, se aplicó el algoritmo **N4 Bias Field Correction**, que estima y corrige suavemente estas irregularidades, mejorando la consistencia de las intensidades dentro de un mismo tejido y favoreciendo el aprendizaje del modelo.

El *bias field* se manifiesta como una variación lenta y no uniforme de la intensidad a lo largo de la imagen, provocada por imperfecciones del campo de radiofrecuencia y la sensibilidad de las bobinas del escáner. Como consecuencia, voxels pertenecientes al mismo tipo de tejido pueden presentar intensidades significativamente diferentes en distintas regiones del volumen.

En modelos de segmentación basados en CNN, estas inconsistencias pueden inducir errores en el aprendizaje, ya que la red puede interpretar variaciones de intensidad no patológicas como características relevantes. La corrección mediante N4 contribuye a homogeneizar las intensidades intra-tejido, facilitando un aprendizaje más estable y robusto.

3.2.4 Extracción de cráneo (Skull Stripping)

La extracción de cráneo o *skull stripping* consiste en eliminar del volumen de resonancia magnética las estructuras no cerebrales, como el cráneo, la piel o los tejidos extracraneales. Este paso permite centrar el análisis exclusivamente en el parénquima

cerebral y reduce la información irrelevante durante el entrenamiento del modelo.

Las etapas descritas anteriormente fueron implementadas mediante scripts en Python, desarrollados en forma de cuadernos Jupyter, que permitieron automatizar el preprocesamiento previo a la fase de entrenamiento. El código específico empleado se adjuntará en el apartado correspondiente separado de la memoria.

La eliminación de tejido no cerebral reduce significativamente el espacio de búsqueda del modelo, disminuye la cantidad de información irrelevante y mejora la relación señal-ruido durante el entrenamiento. No obstante, este paso debe realizarse con precaución, ya que una extracción excesivamente agresiva puede eliminar tejido cerebral válido o alterar regiones cercanas a la corteza, afectando negativamente a la segmentación de lesiones periféricas.

Adicionalmente, tras la corrección de inhomogeneidades y la extracción de cráneo, se aplicó una normalización de intensidades mediante un escalado Min–Max al rango [0,1]. Esta transformación permite homogeneizar el rango dinámico de las intensidades entre sujetos y escáneres, facilitando la estabilidad del entrenamiento y reduciendo la influencia de diferencias radiométricas no relacionadas con la patología.

Tabla 3.2. Resumen de las etapas de preprocesamiento aplicadas a los volúmenes de resonancia magnética.

Etapas	Objetivo principal
Normalización espacial (registro a plantillas)	Alinear todos los volúmenes cerebrales a un espacio anatómico común, reduciendo la variabilidad intersujeto y facilitando el aprendizaje de patrones espaciales consistentes.
Corrección de inhomogeneidades (N4 Bias Field)	Eliminar variaciones lentas de intensidad debidas al campo magnético del escáner, homogeneizando la señal dentro de un mismo tipo de tejido.
Extracción de cráneo (Skull stripping)	Eliminar estructuras no cerebrales (cráneo, piel y tejidos extracraneales) para centrar el análisis exclusivamente en el parénquima cerebral.
Normalización de intensidades	Estandarizar el rango dinámico de intensidades mediante un escalado Min–Max para mejorar la estabilidad del entrenamiento y la convergencia del modelo.

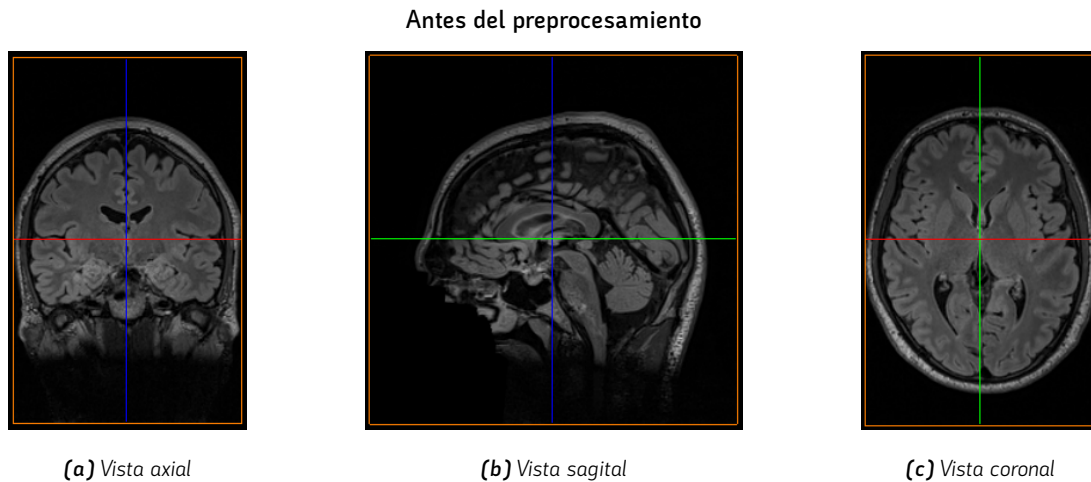


Figura 3.5. Ejemplo de volumen de resonancia magnética antes del preprocesamiento. Se aprecian inhomogeneidades de intensidad y la presencia de estructuras extracraneales.

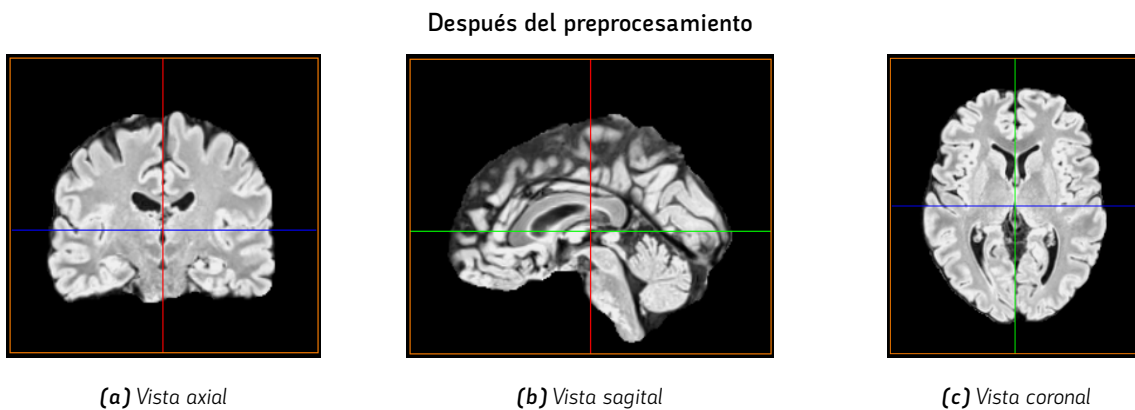


Figura 3.6. Ejemplo del mismo volumen tras el preprocesamiento completo, incluyendo corrección de inhomogeneidades, extracción de cráneo y normalización espacial.

Con el fin de ilustrar el efecto global del preprocesamiento aplicado, en las Figuras 3.5 y 3.6 se muestra un ejemplo representativo de un mismo volumen cerebral antes y después de la aplicación conjunta de todas las transformaciones descritas.

3.3 Algoritmos de entrenamiento

En este apartado se explican los algoritmos de segmentación utilizados para tratar las imágenes, sus características y peculiaridades, así como el flujo del entrenamiento.

3.3.1 deep learning

La inteligencia artificial (IA) engloba un conjunto de técnicas y algoritmos cuyo objetivo es dotar a los sistemas computacionales de la capacidad de realizar tareas que, tradicionalmente, requieren inteligencia humana, como el reconocimiento de patrones, la toma de decisiones o el análisis de información compleja. Dentro de este ámbito, el *machine learning* constituye una

rama fundamental, basada en el diseño de modelos que aprenden automáticamente a partir de datos, sin necesidad de ser programados explícitamente para cada caso concreto.

El *machine learning* se apoya en la idea de que un modelo puede extraer relaciones subyacentes entre las variables de entrada y salida mediante un proceso de entrenamiento. Durante dicho proceso, el algoritmo ajusta sus parámetros internos con el objetivo de minimizar una función de error, que mide la diferencia entre las predicciones del modelo y los valores reales. Este enfoque ha demostrado ser especialmente eficaz en problemas donde las reglas explícitas son difíciles de definir, como ocurre en el análisis de imágenes médicas.

Dentro del *machine learning*, el *deep learning* representa una evolución basada en el uso de redes neuronales profundas, es decir, arquitecturas compuestas por múltiples capas intermedias que permiten **aprender representaciones jerárquicas de los datos**. A diferencia de los métodos tradicionales, donde las características relevantes debían ser diseñadas manualmente, los modelos de *deep learning* son capaces de aprender automáticamente estas representaciones a partir de los datos brutos, lo que los hace especialmente adecuados para tareas complejas de visión por computador.

Como se ilustra en la Figura 3.7, el *deep learning* constituye un subconjunto del *machine learning* dentro del marco general de la inteligencia artificial. Este tipo de representaciones jerárquicas es habitual en la literatura introductoria y docente sobre aprendizaje automático, y permite situar conceptualmente las redes neuronales profundas dentro del ecosistema general de técnicas de IA [11, 12].

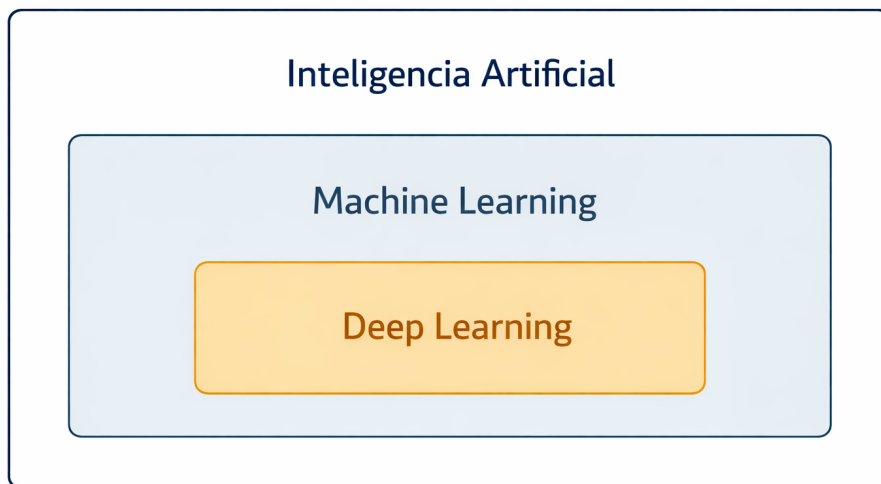


Figura 3.7. Relación jerárquica entre inteligencia artificial, machine learning y deep learning.

En el contexto del procesamiento de imágenes, las redes neuronales convolucionales (*Convolutional Neural Networks*, CNNs) constituyen la arquitectura más extendida. Estas redes emplean operaciones de convolución para extraer características espaciales locales, permitiendo capturar patrones como bordes, texturas o estructuras anatómicas de forma progresiva. A medida que se profundiza en la red, las capas sucesivas aprenden representaciones cada vez más abstractas y semánticamente relevantes.

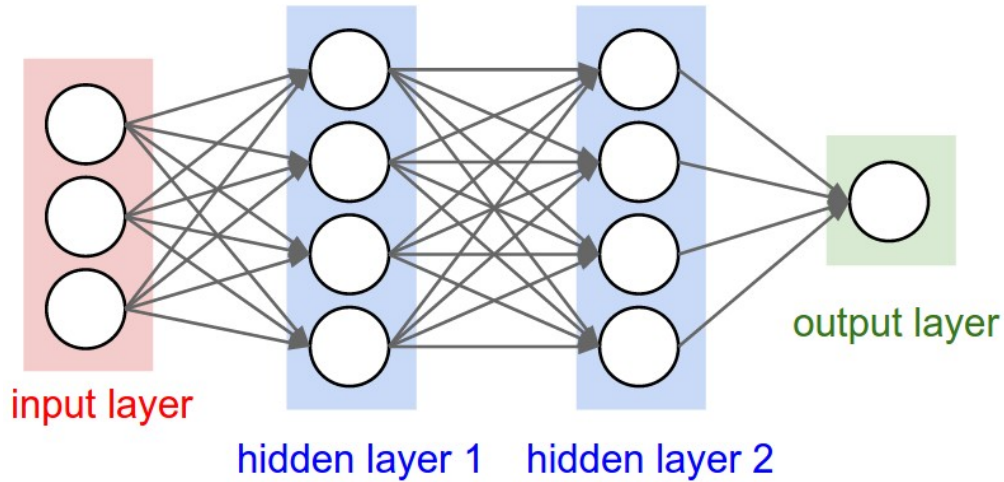


Figura 3.8. Esquema general de una red neuronal profunda con múltiples capas ocultas [12].

El entrenamiento de una **red neuronal profunda** se basa habitualmente en un esquema de aprendizaje supervisado, donde se dispone de pares de datos de entrada (por ejemplo, volúmenes de resonancia magnética) y etiquetas de salida (máscaras de segmentación). Mediante algoritmos de optimización basados en descenso de gradiente y retropropagación del error (*backpropagation*), los parámetros del modelo se ajustan iterativamente para maximizar su capacidad de generalización sobre datos no vistos.

La Figura 3.8 muestra un esquema simplificado de una red neuronal profunda, compuesta por una capa de entrada, varias capas ocultas y una capa de salida, donde cada neurona está conectada con la siguiente capa mediante pesos entrenables. Este tipo de representación se emplea habitualmente con fines didácticos para ilustrar el funcionamiento básico del aprendizaje profundo.

La Figura 3.9 ilustra el concepto de procesamiento convolucional, donde las imágenes se representan como tensores multidimensionales que permiten capturar información espacial local y contextual. Este enfoque constituye la base de las redes convolucionales empleadas en segmentación médica, tal y como se describe en revisiones exhaustivas del campo [13].

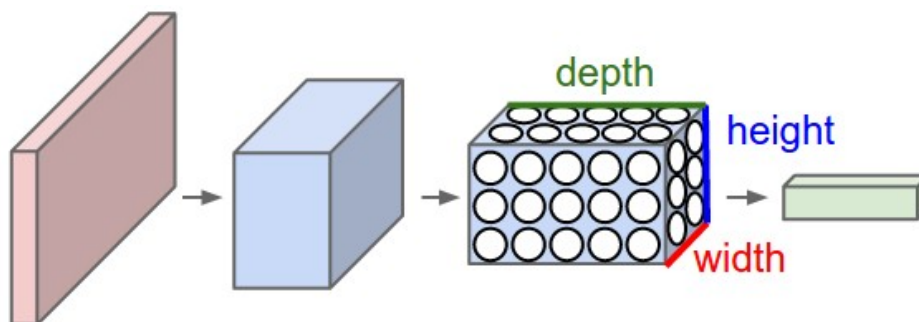


Figura 3.9. Representación del procesamiento volumétrico en redes neuronales convolucionales. El esquema ilustra la operación sobre tensores multidimensionales, habitual en CNNs aplicadas a imágenes médicas [12].

Gracias a su capacidad para modelar relaciones altamente no lineales y su escalabilidad a grandes volúmenes de datos, el

deep learning se ha consolidado como la técnica dominante en tareas de segmentación automática de imágenes médicas. Su aplicación permite abordar problemas de alta complejidad, como la identificación precisa de lesiones cerebrales en resonancia magnética, ofreciendo resultados que en muchos casos igualan o superan el rendimiento humano en escenarios controlados en términos de consistencia y reproducibilidad.

En tareas de segmentación semántica, como las abordadas en este trabajo, el objetivo no es únicamente clasificar una imagen completa, sino asignar una etiqueta a cada uno de los elementos espaciales que la componen. En el caso de imágenes médicas volumétricas, esto implica realizar una predicción a nivel de *voxel*, diferenciando entre tejido sano y regiones patológicas con una alta precisión espacial. Este tipo de segmentación a nivel de *voxel* se ilustra en la Figura 3.10.

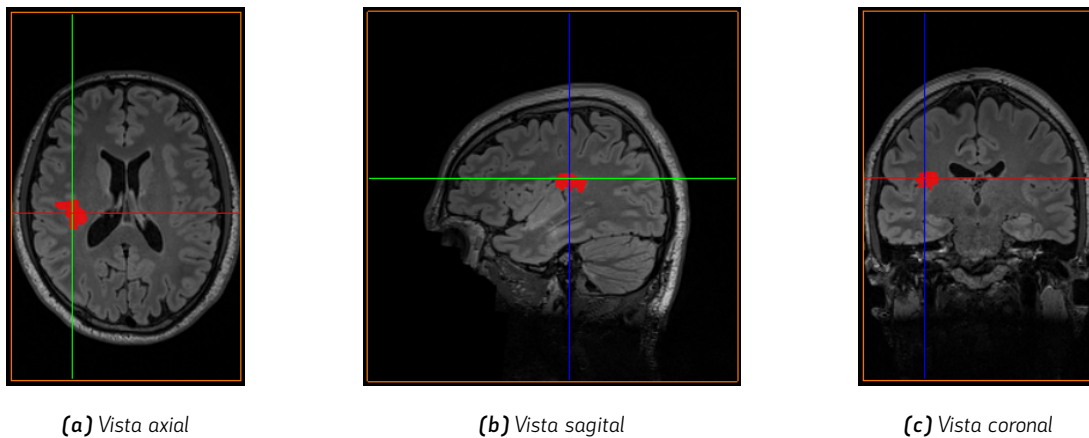


Figura 3.10. Ejemplo de segmentación semántica en resonancia magnética cerebral. La máscara de lesión se superpone a la imagen original y muestra la clasificación a nivel de *voxel* en las tres vistas ortogonales habituales.

Para abordar este problema, las arquitecturas de *deep learning* empleadas suelen adoptar un esquema de tipo *encoder-decoder* [21]. En este enfoque, la parte codificadora (**encoder**) se encarga de reducir progresivamente la resolución espacial de la imagen mientras incrementa la abstracción de las características extraídas, capturando información contextual de alto nivel. Por otro lado, la parte decodificadora (**decoder**) reconstruye la resolución original mediante operaciones de interpolación o convoluciones transpuestas, permitiendo recuperar la localización precisa de las estructuras de interés.

Las conexiones de salto (**skip connections**) entre el *encoder* y el *decoder* juegan un papel fundamental, ya que permiten combinar información semántica profunda con detalles espaciales finos procedentes de las capas iniciales. Este mecanismo resulta especialmente relevante en imágenes médicas, donde pequeñas variaciones espaciales pueden tener un alto impacto clínico, como ocurre en la segmentación de lesiones cerebrales de reducido tamaño.

En el contexto de la resonancia magnética, el *deep learning* ha demostrado una gran capacidad para adaptarse a diferentes modalidades de imagen (T1, T2, FLAIR), así como a datos multicanal y tridimensionales. Las CNN pueden operar tanto sobre cortes bidimensionales como sobre volúmenes completos en 3D, permitiendo capturar dependencias espaciales en múltiples direcciones y mejorar la coherencia de las segmentaciones obtenidas.

3.3.2 nnU-Net v2

nnU-Net es un framework de segmentación automática basado en *deep learning*, diseñado específicamente para el procesamiento de imágenes biomédicas [10, 16]. A diferencia de otros enfoques que proponen arquitecturas novedosas o altamente especializadas, nnU-Net se centra en la **automatización completa del flujo de trabajo de segmentación**, abarcando desde el preprocesamiento de los datos hasta la definición de la arquitectura, el entrenamiento del modelo y la evaluación de resultados.

Introducido inicialmente en el contexto del desafío MICCAI Medical Segmentation Decathlon [3], nnU-Net demostró que una implementación cuidadosamente optimizada de la arquitectura U-Net, acompañada de decisiones automáticas basadas en las características intrínsecas del conjunto de datos, puede superar a métodos más complejos y ajustados manualmente. Este resultado puso de manifiesto que, en muchos casos, el rendimiento de los sistemas de segmentación depende más de una correcta configuración del *pipeline* que de la complejidad de la arquitectura empleada.

Como consecuencia, nnU-Net ha sido adoptado de forma generalizada como método de referencia en numerosos trabajos científicos y competiciones internacionales de segmentación médica, convirtiéndose en un estándar de facto para la evaluación comparativa de nuevos enfoques. Su diseño orientado a la reproducibilidad y a la adaptabilidad a distintos datasets lo hace especialmente adecuado para estudios experimentales y entornos clínicos heterogéneos.

La versión **nnU-Net v2** introduce mejoras sustanciales en modularidad, extensibilidad y eficiencia computacional, facilitando la incorporación de nuevas arquitecturas y estrategias de entrenamiento sin perder la automatización que caracteriza al framework. Como se ilustra en la Figura 3.11, nnU-Net estructura el proceso de segmentación como un *pipeline* automatizado de extremo a extremo.

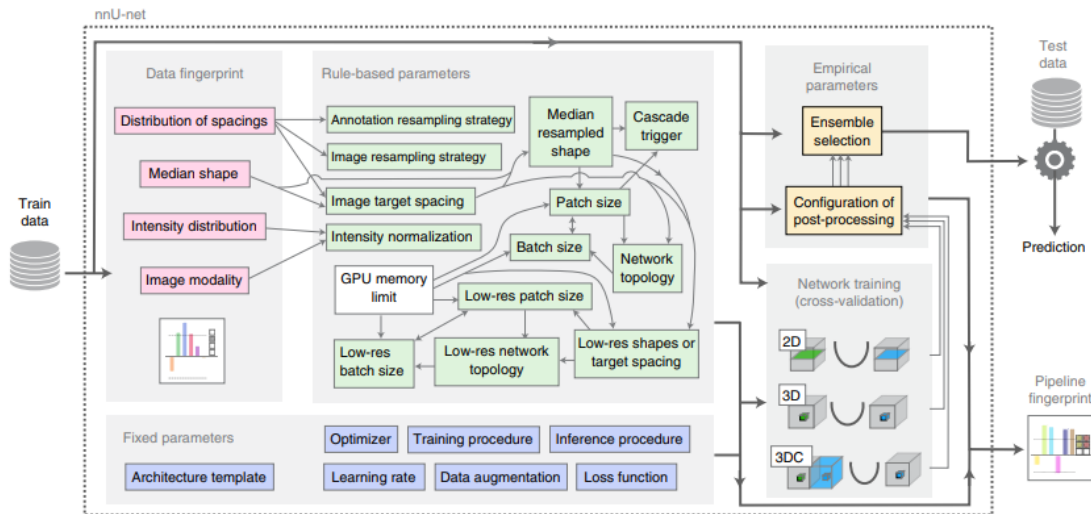


Figura 3.11. Visión general del pipeline de segmentación automática propuesto por nnU-Net, que abarca desde el preprocesamiento de los datos hasta la inferencia y evaluación de resultados. Adaptado de Isensee et al. [10].

3.3.2.1 Filosofía "no new U-Net" y automatización del pipeline

El nombre *nnU-Net* proviene de la expresión "no new U-Net", reflejando la idea central del framework: en lugar de proponer una nueva arquitectura de red neuronal, se optimiza de forma sistemática una arquitectura U-Net ampliamente validada mediante

un *pipeline* completamente automatizado [10, 9]. Esta filosofía surge como respuesta a la proliferación de modelos cada vez más complejos que, en muchos casos, no ofrecen mejoras sustanciales frente a configuraciones más sencillas correctamente ajustadas.

nnU-Net elimina la necesidad de realizar ajustes manuales mediante un conjunto de heurísticas basadas en el análisis automático del propio conjunto de datos. Estas heurísticas determinan aspectos críticos del *pipeline*, como la resolución de entrada, el tamaño del parche, la profundidad de la red, la normalización de intensidades o la estrategia general de entrenamiento más adecuada (2D o 3D), teniendo en cuenta tanto las propiedades de los datos como las limitaciones computacionales disponibles.

Este enfoque permite desacoplar el rendimiento del modelo del grado de experiencia del usuario, reduciendo la dependencia de decisiones subjetivas y minimizando el riesgo de configuraciones subóptimas. Como consecuencia, nnU-Net proporciona un rendimiento robusto y consistente incluso cuando se aplica a datasets con propiedades muy heterogéneas, lo que facilita su uso como método de referencia en estudios comparativos.

En el ámbito clínico, donde los conjuntos de datos presentan una elevada variabilidad en resolución espacial, contraste, número de modalidades y tamaño de las lesiones, esta automatización resulta especialmente ventajosa. Al adaptar el *pipeline* a cada dataset de forma específica y reproducible, nnU-Net facilita la transferencia de modelos entre distintos centros y protocolos de adquisición, reduciendo la variabilidad interexperimental.

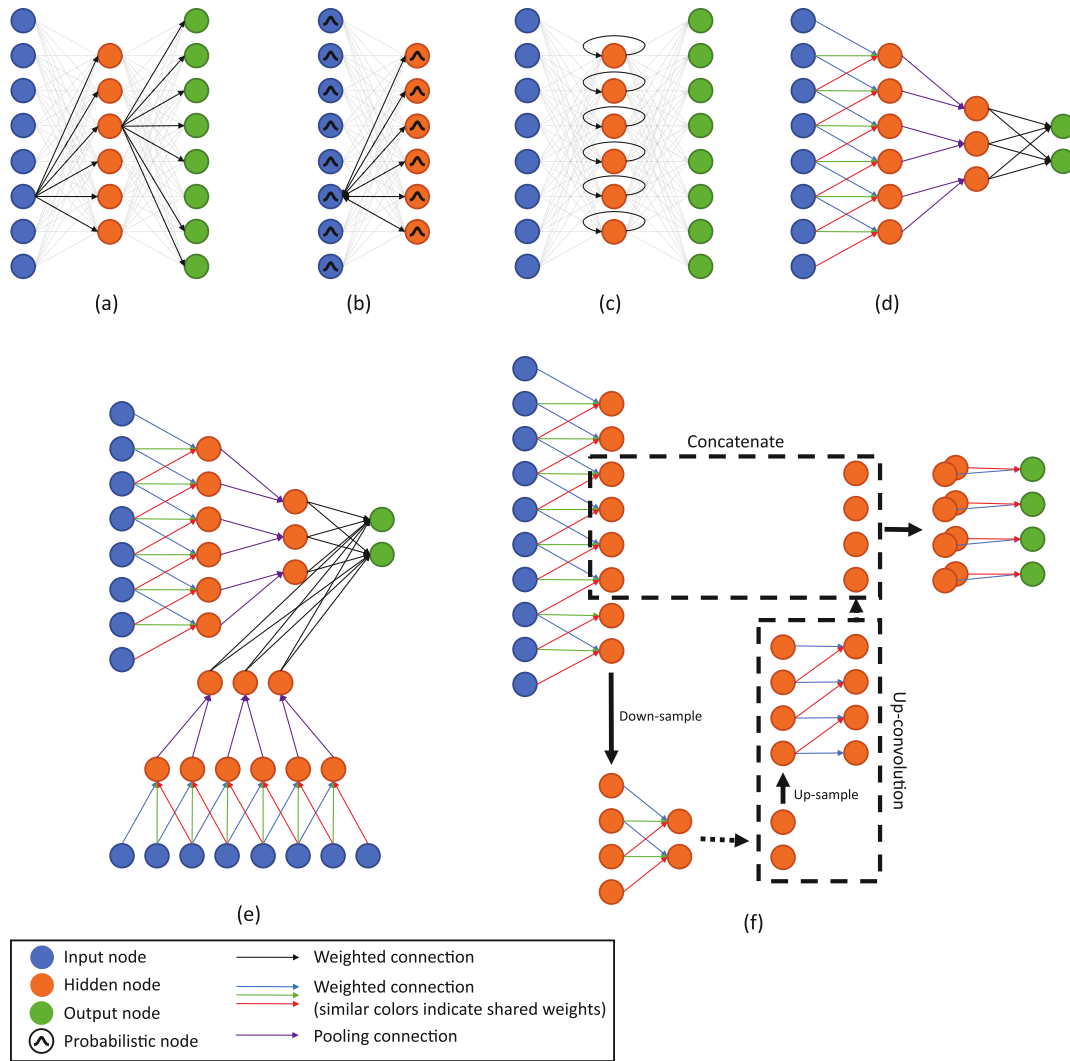


Figura 3.12. Esquema general del procesamiento mediante redes neuronales convolucionales en análisis de imágenes médicas. Adaptado de Litjens et al. [13].

Los enfoques tradicionales de segmentación médica basados en *deep learning* siguen un *pipeline* general como el ilustrado en la Figura 3.12, donde la arquitectura y los parámetros deben ser ajustados manualmente. Frente a este enfoque, nnU-Net propone la automatización sistemática de todas estas etapas, dando lugar a la filosofía conocida como “no new U-Net” [10].

3.3.2.2 Arquitectura base: U-Net y variantes empleadas

El framework *nnU-Net v2* se fundamenta en la arquitectura **U-Net**, un modelo de red neuronal convolucional ampliamente utilizado en tareas de segmentación de imágenes médicas. La arquitectura *U-Net* se caracteriza por una estructura en forma de codificador–decodificador (*encoder–decoder*) con conexiones de salto (*skip connections*) entre niveles simétricos, lo que permite combinar información semántica de alto nivel con detalles espaciales de baja escala [21].

Tal y como se describió en el Estado del Arte (Figura 2.1), la arquitectura *U-Net* se basa en un esquema codificador–decodificador con conexiones de salto. Sobre esta arquitectura base, nnU-Net construye su *pipeline* automatizado.

El **codificador** está compuesto por una sucesión de bloques convolucionales seguidos de operaciones de reducción de resolución (*downsampling*), generalmente mediante convoluciones con *stride* o capas de *pooling*. En esta fase, la red aprende representaciones progresivamente más abstractas del contenido de la imagen, capturando patrones globales relevantes para la segmentación.

El **decodificador**, por su parte, realiza el proceso inverso mediante operaciones de aumento de resolución (*upsampling*), combinando las características aprendidas en el codificador con las activaciones correspondientes a través de las conexiones de salto. Estas conexiones resultan fundamentales para preservar la localización espacial precisa de las estructuras a segmentar, especialmente relevante en imágenes médicas donde las lesiones pueden ser pequeñas o difusas.

Además, la profundidad de la arquitectura y el número de filtros por nivel no se fijan de forma estática, sino que se adaptan automáticamente en función del tamaño de los parches de entrada y de las restricciones de memoria disponibles. Esta flexibilidad permite a *nnU-Net v2* escalar desde arquitecturas relativamente compactas hasta modelos más profundos, manteniendo un equilibrio adecuado entre capacidad representacional y eficiencia computacional.

En *nnU-Net v2*, la arquitectura base se adapta automáticamente a las características del conjunto de datos. Dependiendo de la dimensionalidad de los volúmenes y de la resolución espacial, el framework selecciona entre diferentes variantes:

- **U-Net 2D**: procesa cada corte axial de forma independiente, siendo especialmente adecuada para datasets con gran anisotropía entre ejes.
- **U-Net 3D**: opera directamente sobre volúmenes tridimensionales completos o parches 3D, permitiendo capturar relaciones espaciales entre cortes consecutivos.
- **U-Net 3D de baja resolución**: empleada como complemento en configuraciones en cascada para capturar contexto global en volúmenes de gran tamaño.

Adicionalmente, *nnU-Net v2* introduce variantes más avanzadas como la **Residual Encoder U-Net (ResEnc U-Net)**, que incorpora conexiones residuales en el codificador. Estas conexiones facilitan el entrenamiento de redes profundas, mejoran la propagación del gradiente y contribuyen a una mayor estabilidad durante el aprendizaje, especialmente en datasets complejos y heterogéneos [10].

En conjunto, el uso de una arquitectura *U-Net* flexible y bien establecida permite a *nnU-Net v2* centrar sus esfuerzos en la optimización automática del *pipeline*, en lugar de en el diseño de nuevas arquitecturas. Esta decisión resulta especialmente adecuada en el contexto de la segmentación médica, donde la correcta adaptación de la arquitectura a las propiedades del dataset —como la resolución espacial, la anisotropía o el tamaño de las estructuras de interés— tiene un impacto directo en el rendimiento final del modelo.

3.3.2.3 Configuraciones de entrenamiento

Uno de los principales puntos fuertes de *nnU-Net v2* es su capacidad para soportar múltiples configuraciones de entrenamiento sin requerir ajustes manuales por parte del usuario. El framework evalúa automáticamente las propiedades del conjunto de datos —como el tamaño de los volúmenes, la resolución espacial, el número de modalidades y la disponibilidad de memoria GPU— y selecciona la configuración más adecuada.

La elección entre entrenamiento bidimensional o tridimensional supone un compromiso entre la cantidad de contexto espacial capturado y el coste computacional asociado. Mientras que los enfoques 2D priorizan la estabilidad del entrenamiento y la eficiencia en memoria, los modelos 3D permiten explotar de forma más completa la información volumétrica inherente a las imágenes médicas, si bien su aplicación práctica está condicionada por un mayor coste computacional y mayores requerimientos de memoria.

En particular, las arquitecturas tridimensionales permiten explotar de forma explícita la continuidad espacial entre cortes adyacentes, capturando información contextual en las tres dimensiones del volumen. La Figura 3.13 muestra un ejemplo de segmentación volumétrica obtenida mediante una red 3D como 3D U-Net, donde el contexto completo del volumen es utilizado para producir una segmentación densa [4].

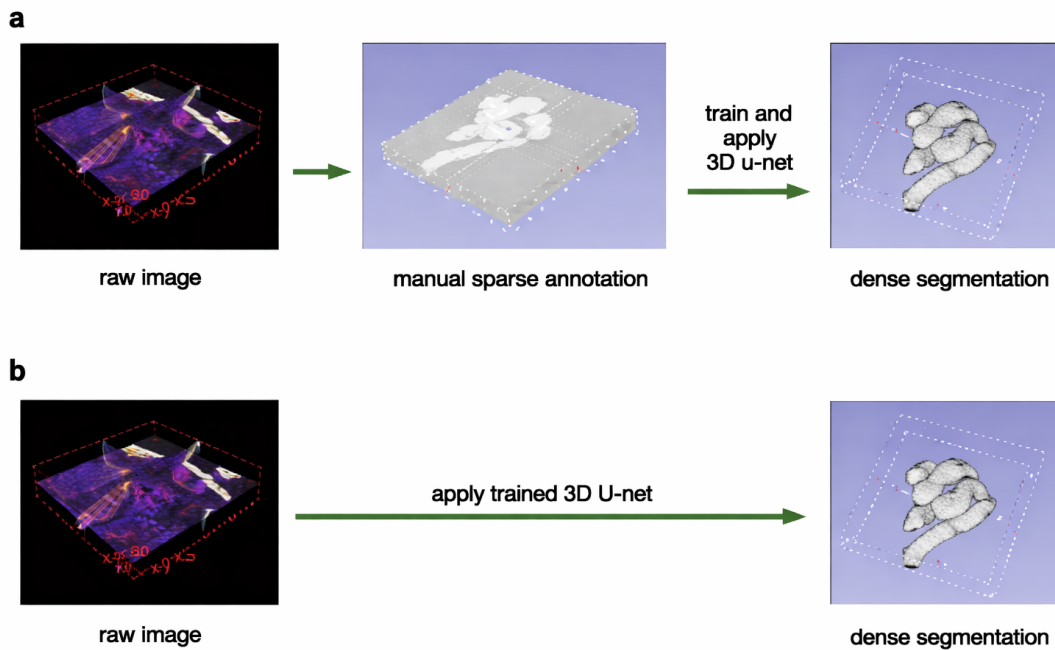


Figura 3.13. Ejemplo de segmentación volumétrica mediante una red neuronal tridimensional (3D U-Net), donde se aprovecha el contexto espacial completo del volumen para producir una segmentación densa. Adaptado de Çiçek et al. [4].

Las configuraciones de entrenamiento más relevantes son las siguientes:

- **Entrenamiento 2D:** el modelo se entrena utilizando cortes bidimensionales extraídos de los volúmenes 3D. Esta configuración reduce significativamente el consumo de memoria y resulta especialmente eficaz en datasets con resolución anisotrópica, donde la distancia entre cortes es considerable.
- **Entrenamiento 3D completo:** el modelo procesa parches tridimensionales, permitiendo capturar información contextual entre cortes adyacentes. Esta configuración suele ofrecer mejores resultados cuando la resolución es aproximadamente isotrópica.

- **Entrenamiento en cascada:** combina un modelo de baja resolución que segmenta el volumen completo con un segundo modelo que refina la segmentación inicial a resolución completa. Esta estrategia es especialmente útil en volúmenes de gran tamaño o con estructuras pequeñas difíciles de detectar [10].

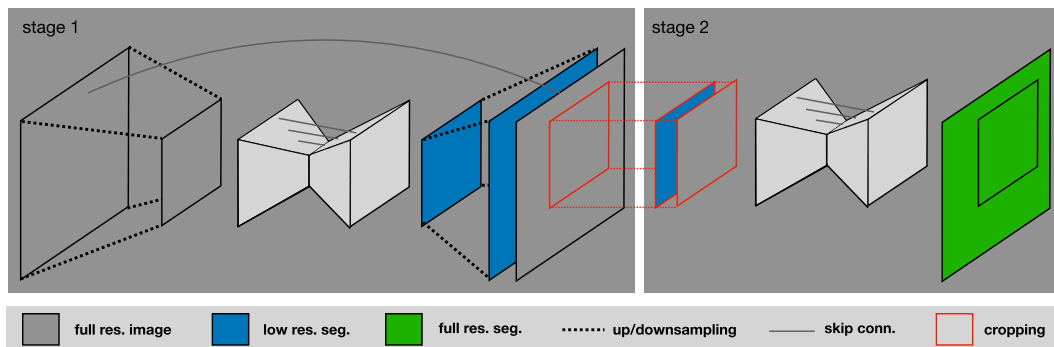


Figura 3.14. Esquema del enfoque de entrenamiento en cascada empleado por nnU-Net. En una primera etapa, un modelo opera sobre datos a baja resolución para capturar el contexto global. En una segunda etapa, la segmentación obtenida se concatena con la imagen a resolución completa y es refinada por un segundo modelo. Adaptado de Isensee et al. [10].

Cada configuración define automáticamente parámetros clave como el tamaño del parche de entrada, el número de capas de la red, el tamaño del *batch*, el número de filtros iniciales y la profundidad máxima de la arquitectura. De este modo, nnU-Net v2 evita decisiones heurísticas manuales y garantiza una utilización eficiente de los recursos computacionales disponibles.

La posibilidad de evaluar automáticamente estas configuraciones permite analizar de forma sistemática el impacto del contexto espacial y de la dimensionalidad del modelo en el rendimiento de segmentación, aspecto de especial relevancia en estudios comparativos como el realizado en este trabajo.

3.3.2.4 Preprocesamiento automático en nnU-Net v2

El preprocesamiento constituye una etapa fundamental dentro del *pipeline* de nnU-Net v2, ya que permite estandarizar los datos de entrada y reducir la variabilidad introducida por diferentes protocolos de adquisición. A diferencia de enfoques tradicionales que emplean cadenas de preprocesamiento fijas, nnU-Net adopta una estrategia adaptativa, en la que las operaciones se ajustan automáticamente a las características específicas de cada conjunto de datos.

Entre las principales operaciones de preprocesamiento se incluyen:

- **Normalización de intensidades:** se aplican estrategias de normalización adaptadas a cada modalidad de imagen, como la normalización basada en percentiles o la estandarización por z-score. En el caso de la resonancia magnética, esta etapa resulta especialmente relevante debido a la ausencia de una escala de intensidades absoluta, permitiendo homogeneizar la distribución de intensidades entre sujetos y reducir la variabilidad inter-escáner.
- **Reescalado de la resolución:** las imágenes se re-muestran a una resolución objetivo calculada a partir de la mediana de las resoluciones del dataset. Este compromiso permite preservar información anatómica relevante al tiempo que se mantiene un coste computacional asumible.

- **Ajuste de tamaños y recorte:** se eliminan regiones irrelevantes del fondo y se ajustan los volúmenes para optimizar el tamaño de los parches de entrenamiento, mejorando la eficiencia del aprendizaje y reduciendo el consumo de memoria.
- **Conversión de etiquetas:** las máscaras de segmentación se transforman a un formato interno consistente, asegurando la correspondencia correcta entre clases y valores numéricos y evitando ambigüedades durante el proceso de entrenamiento.

Estas decisiones de preprocesamiento no se definen manualmente, sino que se derivan automáticamente durante la fase de planificación del experimento, lo que garantiza la reproducibilidad y la coherencia entre diferentes entrenamientos. Además, el mismo *pipeline* se aplica de forma consistente tanto a los datos de entrenamiento como a los de inferencia, evitando discrepancias entre ambas fases.

En el contexto de este Trabajo de Fin de Grado, el preprocesamiento automático de *nnU-Net v2* permitió integrar de manera homogénea datasets con distintas resoluciones, modalidades y características clínicas, contribuyendo a un entrenamiento más estable y facilitando una comparación objetiva entre los distintos modelos entrenados.

Como se comenta en el apartado 3.4.2 este paso no pretende sustituir ni modificar el preprocesamiento interno del framework, sino establecer distintas condiciones experimentales controladas para analizar el impacto del espacio de referencia en la segmentación.

3.3.2.5 Planificación automática del experimento

Uno de los elementos distintivos de *nnU-Net v2* es su capacidad para realizar una **planificación automática del experimento** (*experiment planning*) a partir de las propiedades intrínsecas del conjunto de datos. Esta fase se ejecuta antes del entrenamiento y tiene como objetivo determinar de forma óptima la configuración del modelo, el preprocesamiento y los parámetros de entrenamiento, sin intervención manual del usuario.

A diferencia de enfoques basados en búsqueda exhaustiva de hiperparámetros, la planificación automática en *nnU-Net v2* se basa en un conjunto de heurísticas deterministas cuidadosamente diseñadas. Estas reglas se derivan de un análisis empírico extensivo sobre múltiples datasets de segmentación médica y permiten tomar decisiones coherentes sin recurrir a procesos de optimización costosos.

Durante esta etapa, el framework analiza características clave del dataset, como:

- Dimensiones espaciales de los volúmenes.
- Resolución y anisotropía entre ejes.
- Número de modalidades de imagen.
- Distribución de tamaños de las regiones de interés.
- Disponibilidad de memoria GPU.

En particular, existe una relación directa entre la resolución espacial de los volúmenes, el tamaño del parche de entrada y el consumo de memoria GPU. *nnU-Net v2* ajusta automáticamente estos parámetros con el objetivo de maximizar el campo de visión efectivo del modelo sin exceder las limitaciones de hardware, priorizando configuraciones que permitan capturar el mayor contexto anatómico posible.

A partir de este análisis, *nnU-Net v2* selecciona automáticamente aspectos fundamentales como el tipo de arquitectura (2D o 3D), el tamaño del parche de entrada, la profundidad de la red, el número de filtros por nivel y el tamaño del batch. Asimismo, decide si es conveniente emplear configuraciones en cascada o modelos de baja resolución como paso previo.

Este mecanismo de planificación automática asegura que dos usuarios distintos, trabajando con el mismo conjunto de datos, obtengan configuraciones equivalentes sin necesidad de intervención manual. De este modo, *nnU-Net v2* facilita no solo la reproducibilidad experimental, sino también la comparabilidad objetiva entre estudios y arquitecturas.

Este enfoque permite adaptar el *pipeline* de segmentación a cada conjunto de datos de forma específica, evitando configuraciones genéricas que podrían resultar subóptimas. Además, garantiza la reproducibilidad de los experimentos, ya que todas las decisiones quedan registradas en los archivos de planificación generados automáticamente.

3.3.2.6 Función de pérdida y optimización

El proceso de entrenamiento en *nnU-Net v2* se basa en la minimización de una función de pérdida diseñada específicamente para tareas de segmentación de imágenes médicas. En este framework se emplea habitualmente una combinación de dos funciones complementarias: la **Dice Loss** y la **Cross-Entropy Loss**.

La Dice Loss se deriva del coeficiente de similitud Dice (*Dice Similarity Coefficient, DSC*), una métrica ampliamente utilizada para evaluar el solapamiento entre la segmentación predicha y la segmentación de referencia. El coeficiente Dice se define como:

$$DSC = \frac{2|P \cap G|}{|P| + |G|} \quad (3.1)$$

donde P representa el conjunto de voxels predichos como lesión y G el conjunto de voxels correspondientes a la referencia manual. Esta función resulta especialmente adecuada en escenarios con clases desbalanceadas, ya que prioriza el solapamiento relativo entre regiones de interés y fondo.

Por su parte, la Cross-Entropy Loss evalúa el error de clasificación voxel a voxel, penalizando las discrepancias entre la probabilidad predicha por el modelo y la etiqueta real. Este término proporciona gradientes estables durante el entrenamiento y contribuye a una optimización más robusta, especialmente en las fases iniciales del aprendizaje.

La combinación de ambas funciones permite equilibrar la sensibilidad a regiones pequeñas —frecuentes en la segmentación de lesiones cerebrales— con una optimización global más estable. Mientras que la Dice Loss favorece un alto grado de solapamiento entre las regiones segmentadas, la Cross-Entropy Loss garantiza una señal de aprendizaje local consistente, lo que resulta especialmente eficaz en conjuntos de datos con un fuerte desbalance entre clases.

Para la optimización de los parámetros de la red se emplea el algoritmo **Stochastic Gradient Descent (SGD)** con término de momento. El uso de momento permite suavizar las actualizaciones de los pesos y acelerar la convergencia del modelo. Asimismo, *nnU-Net v2* incorpora estrategias de ajuste dinámico de la tasa de aprendizaje, contribuyendo a una convergencia más estable y reduciendo el riesgo de estancamiento en mínimos locales.

3.3.2.7 Estrategia de validación cruzada (folds)

Con el fin de evaluar de manera robusta el rendimiento del modelo y reducir el sesgo asociado a una única partición de los datos, *nnU-Net v2* emplea una estrategia de **validación cruzada** basada en *k-fold cross-validation*. Este procedimiento se aplica

de forma interna sobre el conjunto de entrenamiento y no afecta al conjunto de test final, que se mantiene completamente independiente.

En esta estrategia, el conjunto de entrenamiento se divide en k subconjuntos o *folds*, siendo habitual el uso de cinco particiones. En cada iteración, uno de los folds se utiliza como conjunto de validación, mientras que los restantes se emplean para el entrenamiento del modelo. Este proceso se repite hasta que cada fold ha sido utilizado como validación una vez, permitiendo que todos los datos contribuyan tanto al entrenamiento como a la evaluación.

La elección de un número reducido de folds responde a un compromiso entre estabilidad estadística y coste computacional, especialmente relevante en el entrenamiento de modelos profundos sobre imágenes médicas. Esta estrategia permite obtener una estimación más fiable del rendimiento del modelo y facilita la comparación objetiva entre distintas configuraciones de entrenamiento.

Además, *nnU-Net v2* permite combinar las predicciones generadas en cada fold mediante técnicas de ensamblado (*ensembling*). Este enfoque contribuye a reducir la varianza del modelo y mejorar la robustez de las segmentaciones finales, aspecto especialmente importante en conjuntos de datos clínicos con un número limitado de pacientes y alta variabilidad interindividual.

3.3.2.8 Sistema de trainers en *nnU-Net v2*

El sistema de *trainers* constituye uno de los mecanismos más flexibles y diferenciadores de *nnU-Net v2*. Un *trainer* define de forma explícita cómo se lleva a cabo el proceso de entrenamiento de un modelo de segmentación, integrando de manera coherente la arquitectura utilizada, la función de pérdida, el optimizador, la política de aprendizaje y las estrategias de validación.

Desde un punto de vista conceptual, el *trainer* actúa como una capa de abstracción que encapsula todos los elementos necesarios para entrenar un modelo bajo una configuración concreta. Entre las principales responsabilidades de un *trainer* se incluyen:

- La definición de la arquitectura específica a emplear.
- La configuración de la función de pérdida y sus posibles ponderaciones.
- La selección del optimizador y de sus hiperparámetros.
- La política de ajuste de la tasa de aprendizaje (*learning rate schedule*).
- La gestión de la estrategia de validación cruzada y de los *folds*.
- La definición del procedimiento de inferencia y postprocesamiento asociado.

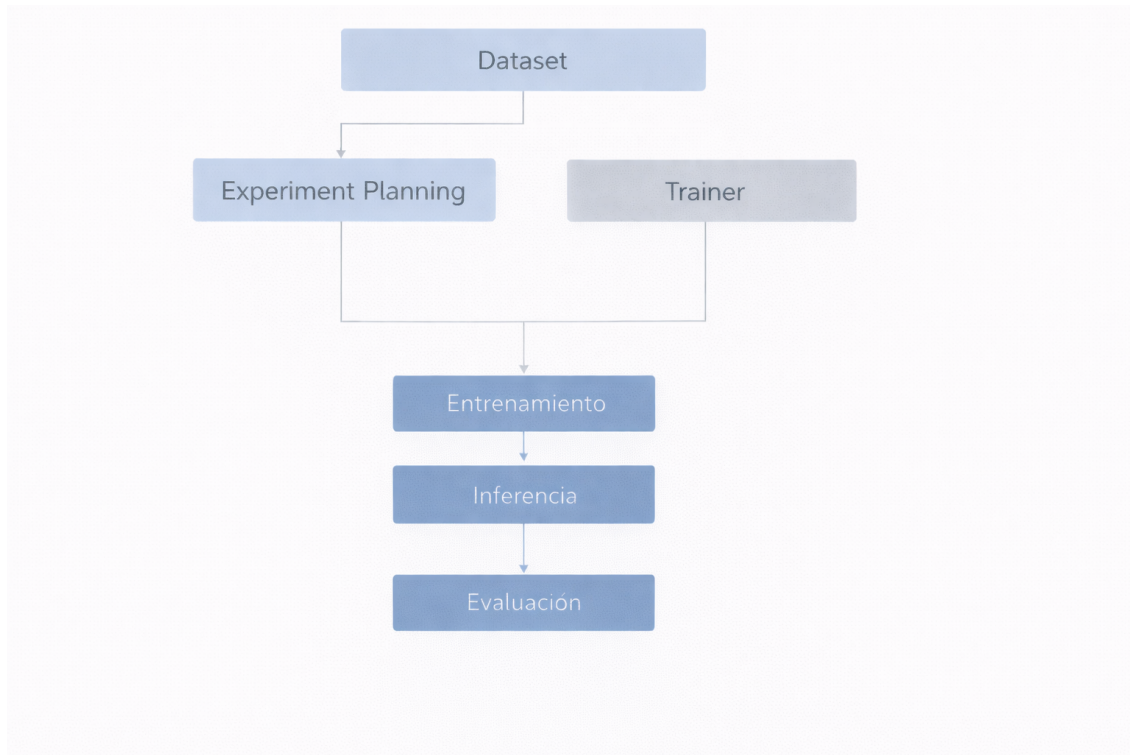


Figura 3.15. Esquema conceptual del sistema de trainers en nnU-Net v2. El trainer actúa como un módulo que integra la arquitectura, la función de pérdida, la optimización y la estrategia de validación dentro del pipeline automatizado de entrenamiento.

nnU-Net v2 proporciona una amplia colección de trainers predefinidos, diseñados para cubrir distintos escenarios experimentales habituales en segmentación médica, como el entrenamiento bidimensional o tridimensional, el uso de arquitecturas con codificadores residuales o configuraciones específicas orientadas a la eficiencia computacional. Asimismo, el framework permite extender o personalizar estos trainers, facilitando la experimentación con nuevas arquitecturas o estrategias de optimización sin necesidad de modificar el núcleo del sistema.

Este diseño modular contrasta con implementaciones monolíticas habituales en *deep learning*, donde la arquitectura y la estrategia de entrenamiento se encuentran fuertemente acopladas. En nnU-Net v2, la separación explícita entre la planificación automática del experimento y la ejecución del entrenamiento mediante trainers favorece la reproducibilidad, la extensibilidad y la comparación sistemática entre distintos enfoques. De este modo, el sistema de trainers constituye una pieza clave para garantizar que los experimentos puedan ser replicados y analizados de forma consistente.

Tabla 3.3. Ejemplos de tipos de trainers disponibles en *nnU-Net v2* y sus características generales.

Trainer	Dim.	Arquitectura base	Uso típico
nnUNetTrainer 2D	2D	<i>U-Net</i>	Datos anisotrópicos
nnUNetTrainer 3D	3D	<i>U-Net</i>	Volúmenes isotrópicos
ResEnc Trainer	2D / 3D	ResEnc <i>U-Net</i>	Datasets complejos
Cascade Trainer	3D	<i>U-Net</i>	Volúmenes grandes

3.3.2.9 Proceso de inferencia y postprocesamiento

Una vez finalizado el entrenamiento, el modelo se utiliza para realizar la inferencia sobre imágenes no vistas, con el objetivo de generar segmentaciones automáticas de nuevos volúmenes de resonancia magnética. En *nnU-Net v2*, este proceso se lleva a cabo aplicando el modelo entrenado sobre parches superpuestos del volumen de entrada, en lugar de procesar el volumen completo de una sola vez.

El uso de parches permite gestionar de forma eficiente las limitaciones de memoria de la GPU, especialmente en imágenes tridimensionales de alta resolución. Durante la inferencia, los parches se extraen siguiendo una estrategia de solapamiento controlado, de modo que cada voxel del volumen puede ser procesado múltiples veces desde diferentes contextos espaciales. Las predicciones obtenidas se combinan posteriormente mediante un promedio ponderado, lo que contribuye a suavizar artefactos en los bordes de los parches y a mejorar la coherencia global de la segmentación.

En configuraciones donde se ha empleado validación cruzada, *nnU-Net v2* permite además combinar las predicciones generadas por los distintos modelos entrenados en cada *fold* mediante técnicas de ensamblado (*ensembling*). Este enfoque suele traducirse en una mejora adicional del rendimiento, al reducir la variabilidad asociada a modelos individuales y aumentar la robustez de las predicciones finales.

El framework incluye asimismo etapas de postprocesamiento opcionales, orientadas a refinar las segmentaciones obtenidas. Entre las operaciones más habituales se encuentran la eliminación de regiones pequeñas aisladas, que suelen corresponder a falsos positivos, y la selección del componente conectado de mayor tamaño cuando la anatomía o la patología de interés lo justifica. Estas operaciones permiten mejorar la coherencia anatómica de las segmentaciones y facilitar su interpretación clínica.

En este Trabajo de Fin de Grado, el proceso de inferencia se realizó siguiendo el *pipeline* estándar de *nnU-Net v2*, garantizando la consistencia entre los resultados de entrenamiento y evaluación, así como la reproducibilidad de los experimentos realizados.

Este proceso de inferencia y postprocesamiento se enmarca dentro del *pipeline* completamente automatizado de *nnU-Net v2*. Tal y como se ilustra en la Figura 3.11, el framework integra de forma coherente las distintas etapas del flujo de trabajo, desde el análisis inicial del conjunto de datos y el preprocesamiento, hasta el entrenamiento del modelo, la inferencia sobre datos no vistos y la evaluación de los resultados obtenidos.

3.3.2.10 Métricas de evaluación

La evaluación del rendimiento de los modelos se llevó a cabo mediante métricas cuantitativas estándar en segmentación de imágenes médicas, basadas en el solapamiento espacial entre la segmentación predicha y la segmentación de referencia manual.

La métrica principal empleada fue el **Dice Similarity Coefficient (DSC)**, que mide el grado de coincidencia entre ambas segmentaciones:

$$DSC = \frac{2|P \cap G|}{|P| + |G|}$$

donde P representa el conjunto de voxels predichos como lesión y G el conjunto de voxels pertenecientes a la referencia. Esta métrica resulta especialmente adecuada en problemas con fuerte desbalance entre clases, como ocurre en la segmentación de lesiones cerebrales.

De forma complementaria se utilizó la métrica **Intersection over Union (IoU)**, definida como:

$$IoU = \frac{|P \cap G|}{|P \cup G|}$$

El IoU proporciona una medida más estricta del solapamiento, penalizando en mayor medida las discrepancias espaciales entre segmentaciones.

Además, se analizaron directamente los valores derivados de la matriz de confusión voxel a voxel:

- **TP (True Positives)**: voxels de lesión correctamente segmentados.
- **FP (False Positives)**: voxels predichos como lesión que no pertenecen a la referencia.
- **FN (False Negatives)**: voxels de lesión no detectados por el modelo.
- n_{ref} : volumen total de la lesión en la referencia manual.
- n_{pred} : volumen total de la lesión segmentada por el modelo.

Estas medidas permiten interpretar no solo la calidad del solapamiento, sino también el comportamiento del modelo en términos de sobresegmentación (exceso de volumen predicho) o infrasegmentación (pérdida de lesión).

El uso conjunto de métricas de solapamiento y análisis volumétrico proporciona una evaluación más completa del rendimiento del modelo, permitiendo distinguir entre segmentaciones espacialmente precisas y segmentaciones volumétricamente sesgadas.

3.3.2.11 Limitaciones del enfoque nnU-Net

A pesar de su solidez y versatilidad, *nnU-Net v2* presenta una serie de limitaciones que conviene tener en cuenta. En primer lugar, su elevado coste computacional puede suponer una barrera en entornos con recursos limitados, especialmente en configuraciones tridimensionales de alta resolución. El entrenamiento de modelos 3D requiere una cantidad considerable de memoria GPU y tiempos de entrenamiento prolongados, lo que puede restringir su aplicabilidad en contextos no especializados.

Asimismo, aunque el enfoque de configuración automática reduce significativamente la necesidad de ajustes manuales, también limita el control fino sobre determinados aspectos del *pipeline*. En escenarios muy específicos, donde se dispone de conocimiento experto detallado sobre la anatomía o la patología de interés, esta falta de flexibilidad puede dificultar la incorporación de heurísticas personalizadas o estrategias de entrenamiento adaptadas a casos concretos.

Otra limitación relevante es la dependencia del modelo respecto a la calidad y coherencia de los datos de entrada. *nnU-Net v2* asume que las imágenes y las máscaras de segmentación presentan una anotación consistente y un preprocesamiento adecuado.

Errores en las segmentaciones de referencia, inconsistencias entre protocolos de adquisición o una elevada heterogeneidad entre centros pueden afectar negativamente al rendimiento y a la capacidad de generalización del modelo.

Por último, aunque las arquitecturas basadas en convoluciones han demostrado un rendimiento sobresaliente en tareas de segmentación médica, su capacidad para modelar dependencias globales a largo alcance es limitada en comparación con enfoques más recientes. Esta restricción puede ser especialmente relevante en imágenes volumétricas complejas, donde el contexto global juega un papel importante en la correcta delimitación de las lesiones.

Estas limitaciones motivan la exploración de enfoques complementarios y arquitecturas alternativas, como los modelos basados en *state space models*, que se abordan en secciones posteriores de este trabajo con el objetivo de analizar su potencial para superar algunas de las restricciones inherentes a las CNN tradicionales.

3.3.3 U-Mamba

U-Mamba es una arquitectura de segmentación biomédica recientemente propuesta que extiende el framework *nnU-Net v2* mediante la incorporación de **State Space Models (SSM)** como mecanismo principal para el modelado de dependencias espaciales y contextuales [14]. Su objetivo es mejorar la capacidad de representación global de los modelos de segmentación, especialmente en imágenes médicas tridimensionales, donde las relaciones a larga distancia entre regiones anatómicas juegan un papel fundamental.

A diferencia de los enfoques tradicionales basados exclusivamente en convoluciones, *U-Mamba* introduce un paradigma híbrido que combina la eficiencia local de las arquitecturas tipo *U-Net* con la capacidad de modelado secuencial y contextual propia de los modelos de espacio de estados [14]. Esta combinación permite capturar información global del volumen sin incurrir en el elevado coste computacional asociado a otros mecanismos de atención global, como los *Transformers*.

Desde un punto de vista práctico, *U-Mamba* mantiene la filosofía de automatización y robustez de *nnU-Net v2*, integrando los bloques SSM de forma modular dentro del *pipeline* existente. En este trabajo se ha empleado la implementación disponible públicamente por los autores a través de su repositorio, que facilita la reproducción del *pipeline* y la ejecución del *trainer* correspondiente [22]. De este modo, la arquitectura busca aprovechar las ventajas de ambos enfoques —convolucional y secuencial— sin sacrificar la escalabilidad ni la eficiencia computacional necesarias para el procesamiento de imágenes médicas de alta resolución.

3.3.3.1 Motivación y contexto de U-Mamba

Las arquitecturas convolucionales clásicas, como *U-Net* y sus extensiones, han demostrado un rendimiento sobresaliente en tareas de segmentación biomédica, consolidándose como el enfoque dominante durante la última década. Sin embargo, presentan una limitación inherente: su campo receptivo efectivo crece de forma progresiva y depende directamente de la profundidad de la red y del tamaño de los kernels empleados. Como consecuencia, la captura de relaciones espaciales de largo alcance puede resultar limitada, especialmente en volúmenes tridimensionales de gran tamaño.

Este aspecto resulta particularmente relevante en el análisis de resonancias magnéticas cerebrales, donde las lesiones pueden aparecer distribuidas en regiones anatómicas alejadas entre sí, presentar morfologías complejas o mostrar patrones difusos que requieren un contexto global para su correcta delimitación. Aunque el aumento de la profundidad de la red puede mitigar parcialmente este problema, también conlleva un incremento del coste computacional y una mayor dificultad en el

entrenamiento.

Los modelos basados en atención, como los *Transformers*, surgieron como una alternativa capaz de modelar dependencias globales de forma explícita. No obstante, su complejidad computacional y su elevado consumo de memoria los hacen poco eficientes para imágenes médicas volumétricas de alta resolución, limitando su aplicabilidad práctica en muchos escenarios clínicos.

En este contexto, los *State Space Models* (SSM) emergen como una solución intermedia, capaz de capturar dependencias globales y de largo alcance de manera eficiente y escalable. *U-Mamba* se apoya en este principio para integrar bloques SSM dentro de una arquitectura tipo U-Net, manteniendo la estructura encoder–decoder característica de *nnU-Net v2*, pero enriqueciendo su capacidad de modelado contextual sin introducir un sobre coste computacional significativo.

3.3.3.2 *State Space Models en U-Mamba*

Los *State Space Models* (SSM) constituyen una clase de modelos matemáticos diseñados para describir sistemas dinámicos mediante un conjunto de estados internos que evolucionan a lo largo de una dimensión secuencial. Tradicionalmente empleados en el ámbito del control y el procesamiento de señales, estos modelos han sido recientemente adaptados al contexto del *deep learning* como una alternativa eficiente para el modelado de dependencias a largo alcance.

En el contexto de las arquitecturas neuronales modernas, los SSM permiten procesar secuencias largas de forma eficiente, manteniendo una complejidad computacional lineal con respecto al tamaño de la entrada. Esta característica los convierte en una opción especialmente atractiva frente a mecanismos de atención global, como los *Transformers*, cuya complejidad cuadrática limita su escalabilidad en aplicaciones de gran tamaño.

En *U-Mamba*, los SSM se emplean para procesar representaciones intermedias de las imágenes médicas, permitiendo a la red integrar información contextual global a lo largo de dimensiones espaciales sin necesidad de recurrir a mecanismos de atención explícitos. Este enfoque resulta particularmente beneficioso en imágenes de resonancia magnética, donde la coherencia anatómica y la relación entre regiones distantes del volumen son factores clave para una segmentación precisa.

Desde un punto de vista arquitectónico, los bloques Mamba se integran dentro de la estructura encoder–decoder característica de U-Net, complementando las operaciones convolucionales tradicionales. Estos bloques actúan como módulos de procesamiento secuencial que enriquecen las características extraídas por la red, facilitando la propagación de información global y mejorando la capacidad de modelado contextual sin introducir un sobre coste computacional significativo.

3.3.3.3 *Arquitectura general de U-Mamba*

La arquitectura de *U-Mamba* conserva la estructura en forma de “U” característica de *nnU-Net v2*, compuesta por un camino codificador (*encoder*) encargado de extraer representaciones jerárquicas de la imagen de entrada y un camino decodificador (*decoder*) que reconstruye la segmentación a resolución original mediante conexiones de salto (*skip connections*). Estas conexiones permiten combinar información semántica de alto nivel con detalles espaciales finos, manteniendo la precisión en la localización de las estructuras de interés.

La principal diferencia respecto a una U-Net convencional radica en la introducción de bloques **Mamba** dentro de esta estructura. Estos bloques, basados en *State Space Models*, se integran de forma selectiva en determinadas etapas del encoder y/o del decoder, complementando las operaciones convolucionales tradicionales sin alterar la organización general del *pipeline*.

- El sistema estándar de inferencia, postprocesamiento y evaluación de resultados.

Desde un punto de vista práctico, la incorporación de *U-Mamba* en *nnU-Net v2* se realiza mediante un **trainer específico**, que define la arquitectura interna del modelo y sustituye determinados bloques convolucionales por módulos basados en *State Space Models*. Este enfoque permite introducir capacidades adicionales de modelado contextual sin alterar el flujo general del *pipeline*.

Esta estrategia de integración resulta especialmente ventajosa desde el punto de vista experimental, ya que permite comparar directamente *nnU-Net v2* y *U-Mamba* bajo condiciones controladas. Al mantenerse constantes el conjunto de datos, el preprocesamiento, las funciones de pérdida, las métricas de evaluación y el procedimiento de inferencia, cualquier diferencia en el rendimiento puede atribuirse principalmente a la arquitectura del modelo.

En el contexto de este Trabajo de Fin de Grado, esta integración facilita una evaluación objetiva del impacto de los bloques *Mamba* en la segmentación de lesiones cerebrales, evitando factores de confusión derivados de cambios en el *pipeline* o en la preparación de los datos.

3.3.3.5 Proceso de entrenamiento con *U-Mamba*

El proceso de entrenamiento de *U-Mamba* sigue el mismo flujo general que *nnU-Net v2*, apoyándose en su *pipeline* completamente automatizado. Tras la fase de planificación automática del experimento, el entrenamiento se lanza utilizando los scripts estándar del framework, seleccionando el *trainer* específico que incorpora la arquitectura *U-Mamba*.

Desde el punto de vista operativo, el procedimiento de entrenamiento no difiere sustancialmente del de *nnU-Net v2*: el modelo se entrena mediante aprendizaje supervisado a partir de imágenes de resonancia magnética previamente preprocesadas, optimizando la función de pérdida definida por el framework. Esta compatibilidad permite integrar *U-Mamba* de forma directa en el ecosistema *nnU-Net*, sin necesidad de modificar los flujos de datos ni las estrategias de validación.

La principal diferencia reside en la arquitectura interna utilizada durante el entrenamiento. En *U-Mamba*, los bloques convolucionales tradicionales se ven complementados por bloques basados en *State Space Models*, que permiten procesar representaciones intermedias con un mayor contexto global. Durante el entrenamiento, estos bloques facilitan la propagación de información a largo alcance dentro del volumen, lo que puede contribuir a una mejor coherencia espacial en la segmentación.

Este aspecto resulta especialmente relevante en escenarios clínicos donde las lesiones presentan una distribución dispersa o patrones morfológicos complejos, como ocurre en patologías neurológicas con alta variabilidad espacial. En estos casos, la capacidad de integrar contexto global durante el entrenamiento puede favorecer una delimitación más consistente de las regiones patológicas.

Cabe destacar que *U-Mamba* no introduce cambios en las estrategias de optimización, validación cruzada o inferencia respecto a *nnU-Net v2*. El entrenamiento mantiene las mismas políticas de aprendizaje, funciones de pérdida y mecanismos de regularización, garantizando así una comparación directa y justa entre ambos enfoques.

3.3.3.6 Ventajas y limitaciones de *U-Mamba*

El diseño de *U-Mamba* surge como una alternativa a las arquitecturas basadas en mecanismos de atención global, particularmente los Transformers aplicados a segmentación médica (por ejemplo, TransUNet o UNETR). Estos modelos permiten capturar dependencias espaciales de largo alcance, pero presentan un elevado coste computacional y de memoria en imágenes

volumétricas 3D. Los *State Space Models* buscan mantener dicha capacidad de modelado global reduciendo significativamente la complejidad computacional.

La incorporación de *State Space Models* en la arquitectura *U-Mamba* introduce una serie de ventajas relevantes frente a enfoques tradicionales basados exclusivamente en convoluciones. Entre los principales aspectos positivos de este modelo destacan los siguientes:

- **Mayor capacidad de modelado global:** los bloques Mamba permiten capturar dependencias espaciales de largo alcance de forma eficiente, superando la limitación del campo receptivo local inherente a las arquitecturas puramente convolucionales.
- **Eficiencia computacional frente a modelos de atención:** en comparación con arquitecturas basadas en Transformers, *U-Mamba* ofrece una complejidad computacional más favorable y un menor consumo de memoria, lo que resulta especialmente adecuado para imágenes médicas volumétricas de alta resolución.
- **Integración directa con nnU-Net v2:** al implementarse como un *trainer* específico dentro del framework, *U-Mamba* hereda automáticamente el preprocesamiento, la planificación del experimento y las estrategias de evaluación de nnU-Net, facilitando su uso y permitiendo comparaciones experimentales justas.

A pesar de estas ventajas, *U-Mamba* presenta también una serie de limitaciones que deben ser consideradas. En primer lugar, al tratarse de una arquitectura relativamente reciente, su comportamiento puede ser más sensible a la configuración de hiperparámetros y a las características del conjunto de datos, especialmente en escenarios con un número reducido de muestras.

Asimismo, la introducción de bloques Mamba incrementa la complejidad arquitectónica del modelo, lo que se traduce en un mayor coste computacional durante el entrenamiento en comparación con nnU-Net v2 estándar. Aunque este incremento es moderado en relación con otros enfoques de modelado global, puede suponer una limitación práctica en entornos con recursos computacionales restringidos.

Finalmente, el beneficio real de *U-Mamba* depende en gran medida del tipo de patología y de la distribución espacial de las lesiones. En casos donde el contexto global juega un papel determinante, como en lesiones difusas o multifocales, el uso de bloques SSM puede resultar especialmente ventajoso. Sin embargo, en escenarios donde la información relevante es principalmente local, las mejoras respecto a nnU-Net v2 pueden ser más limitadas.

Tabla 3.4. Comparación conceptual entre diferentes enfoques de segmentación médica.

Modelo	Modelado global	Coste computacional	Integración en nnU-Net	Escalabilidad
nnU-Net v2	Limitado	Bajo–medio	Nativa	Alta
<i>U-Mamba</i>	Alto	Medio	Nativa (trainer)	Alta
Transformers	Muy alto	Alto	No nativa	Baja en 3D volumétrico

3.3.3.7 Justificación del uso de *U-Mamba* en este trabajo

La inclusión de *U-Mamba* en este Trabajo de Fin de Grado responde a la necesidad de explorar arquitecturas emergentes que vayan más allá de los enfoques puramente convolucionales tradicionales, manteniendo al mismo tiempo un entorno experimental

controlado y reproducible. En este sentido, la comparación entre *nnU-Net v2* y *U-Mamba* permite analizar el impacto real de la incorporación de mecanismos de modelado global basados en *State Space Models* en tareas de segmentación de lesiones cerebrales.

U-Mamba se presenta como una extensión natural de *nnU-Net v2*, ya que se integra directamente en su framework mediante un *trainer* específico, reutilizando el mismo preprocesamiento, la planificación automática del experimento y las estrategias de evaluación. Esta integración garantiza que las diferencias observadas entre ambos enfoques puedan atribuirse principalmente a la arquitectura del modelo, y no a variaciones en el *pipeline* o en la preparación de los datos.

Desde un punto de vista metodológico, el uso de *U-Mamba* permite evaluar si el aumento de capacidad de modelado contextual, especialmente relevante en imágenes de resonancia magnética cerebral con lesiones distribuidas o de morfología compleja, se traduce en mejoras medibles en el rendimiento de segmentación. Asimismo, su estudio aporta una perspectiva exploratoria alineada con las tendencias actuales del estado del arte en segmentación médica, donde se buscan alternativas eficientes a los mecanismos de atención global.

En el contexto de este trabajo, *U-Mamba* no se plantea como un reemplazo directo de *nnU-Net v2*, sino como un complemento que permite profundizar en el análisis comparativo de arquitecturas modernas, aportando valor tanto desde el punto de vista técnico como formativo.

3.4 Diseño experimental

En esta sección se describe el diseño experimental seguido en este trabajo, detallando tanto los pasos previos de preparación de los datos como el flujo completo de ejecución del *pipeline* de *nnU-Net v2*. Se abordan de forma sistemática todas las fases del proceso, desde la organización inicial de los conjuntos de datos y el preprocesamiento previo al entrenamiento, hasta la planificación automática, el entrenamiento de los modelos, la inferencia, el postprocesamiento y la evaluación final de los resultados.

3.4.1 Preparación y organización de los conjuntos de datos

El primer paso del diseño experimental consiste en la descarga, preparación y organización de los conjuntos de datos empleados en el estudio. Cada dataset fue obtenido desde su repositorio oficial y reorganizado para ajustarse estrictamente al formato requerido por *nnU-Net v2*. Esta reorganización resulta imprescindible, ya que el framework asume una estructura de directorios y una nomenclatura de archivos específicas para poder identificar correctamente las imágenes, las modalidades y las máscaras de segmentación.

Durante esta fase se llevaron a cabo las siguientes tareas:

- **Selección de modalidades de imagen:** se emplearon las modalidades **T1** y **FLAIR**, ampliamente utilizadas en neuroimagen clínica por su capacidad para resaltar diferentes tipos de lesiones cerebrales y alteraciones tisulares.
- **Renombrado sistemático de las imágenes:** los volúmenes de resonancia magnética fueron renombrados siguiendo la convención establecida por *nnU-Net*, `CEREBRO_XXXX_0000.nii.gz` y `CEREBRO_XXXX_0001.nii.gz`, donde `XXXX` identifica de forma única a cada sujeto y el sufijo numérico indica la modalidad de imagen correspondiente (0000 → FLAIR, 0001 → T1).

- **Alineación y renombrado de las máscaras de segmentación:** las máscaras asociadas a cada caso fueron renombradas y verificadas para garantizar su correspondencia espacial con las imágenes de entrada, asegurando la coherencia entre datos y etiquetas.
- **División del conjunto de datos:** se realizó una separación explícita entre los datos de entrenamiento y de test, almacenándolos en los directorios `imagesTr`, `labelsTr` e `imagesTs`, conforme a la estructura esperada por el framework.

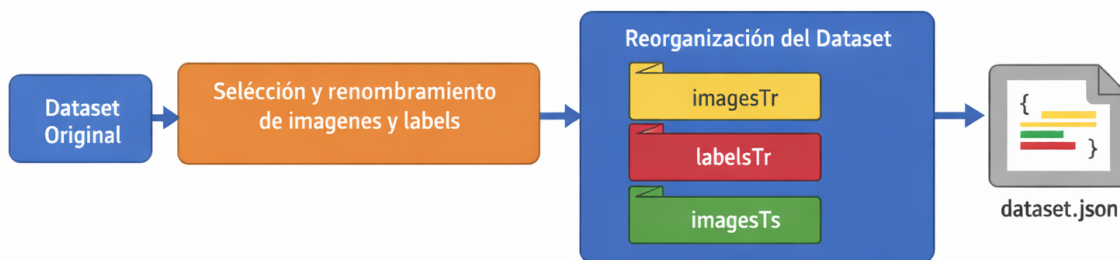


Figura 3.17. Esquema del proceso de preparación y organización de los datasets para su uso en pipeline de *nnU-Net v2*.

Adicionalmente, se generó el archivo `dataset.json`, que describe formalmente el conjunto de datos y actúa como punto de entrada para el *pipeline* de *nnU-Net v2*. Este archivo incluye información clave como las modalidades utilizadas, las clases de segmentación, el número total de casos y las rutas de acceso a los distintos subconjuntos de datos.

La correcta preparación y organización de los conjuntos de datos resulta fundamental para garantizar la reproducibilidad de los experimentos y permitir una comparación objetiva entre los distintos modelos evaluados en este trabajo.

3.4.2 Preprocesamiento previo al entrenamiento

Antes de ejecutar el *pipeline* estándar de *nnU-Net v2*, se llevó a cabo una etapa adicional de preprocesamiento mediante scripts propios implementados en un entorno de tipo *notebook*. Esta fase se concibe como un paso previo al preprocesamiento interno del framework y tiene como objetivo mejorar la calidad y coherencia de las imágenes de entrada antes de la planificación automática del experimento.

Las transformaciones aplicadas en esta etapa corresponden a las técnicas descritas en la Sección de Preprocesamiento, e incluyen la corrección de inhomogeneidades de intensidad mediante *N4 bias field correction*, la eliminación de estructuras extracerebrales (*skull stripping*) y la normalización de intensidades mediante un escalado Min–Max al rango $[0, 1]$. Estas técnicas se emplearon de forma sistemática con el fin de reducir la variabilidad no deseada entre sujetos y adquisiciones, especialmente en conjuntos de datos heterogéneos.

Este preprocesamiento previo no sustituye al preprocesamiento interno de *nnU-Net v2*, sino que lo complementa, proporcionando imágenes espacial y radiométricamente más homogéneas antes de la fase de planificación automática. De este modo, se favorece un entrenamiento más estable y se reduce la influencia de artefactos o variaciones de intensidad que podrían afectar negativamente al aprendizaje del modelo.

3.4.3 Pipeline del algoritmo

Una vez preparados los datos, se ejecuta el *pipeline* completo de *nnU-Net v2*. Este *pipeline* está diseñado para automatizar la mayor parte de las decisiones experimentales, minimizando la intervención manual y favoreciendo la reproducibilidad: a partir de la estructura del dataset, el framework planifica el experimento, preprocesa los datos, entrena el modelo mediante validación cruzada y, finalmente, genera predicciones para su evaluación.

En esta sección se describen las distintas etapas del flujo de trabajo seguido en el presente estudio, indicando los comandos empleados y resumiendo los procesos internos más relevantes en cada fase. Asimismo, se señalan los puntos en los que el uso de *U-Mamba* difiere de *nnU-Net v2*, manteniendo constantes el preprocesamiento y el protocolo de evaluación para asegurar una comparación justa.

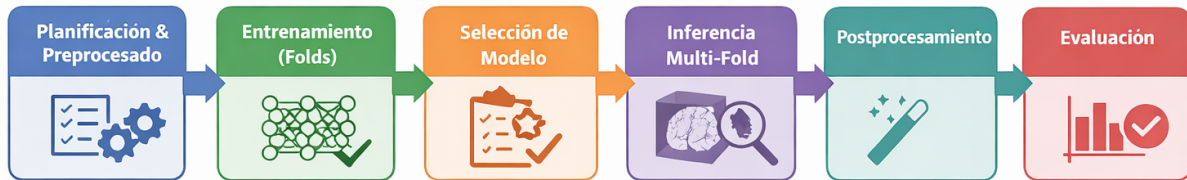


Figura 3.18. Resumen del pipeline empleado con *nnU-Net v2* y *U-Mamba*

3.4.3.1 Verificación del dataset, planificación automática y preprocesamiento interno

El primer paso del *pipeline* se ejecuta mediante:

```
nnUNetv2_plan_and_preprocess -d DATASET_ID -pl PLANNER --verify_dataset_integrity
```

Esta etapa integra tres tareas estrechamente relacionadas: (i) la **verificación de integridad** del dataset, (ii) la **planificación automática del experimento** y (iii) el **preprocesamiento interno** que materializa dicha planificación.

En primer lugar, *nnU-Net v2* comprueba que la estructura del conjunto de datos cumple el formato esperado (carpetas `imagesTr/labelsTr/imagesTs`), que existen los pares imagen-máscara para cada caso de entrenamiento y que los ficheros son legibles. Asimismo, valida que las máscaras contienen etiquetas coherentes (por ejemplo, que el fondo es 0) y que imágenes y etiquetas son compatibles en dimensiones, *spacing* y sistema de coordenadas.

A continuación, el framework analiza el conjunto de datos para extraer un *fingerprint* con propiedades intrínsecas del dataset (tamaño típico de volúmenes, resolución espacial, anisotropía entre ejes y estadísticas básicas de intensidad). A partir de este perfil se generan automáticamente los *plans*, que fijan la configuración del experimento y determinan las transformaciones necesarias para estandarizar los datos.

En este trabajo para el algoritmo *nnUnetv2* se utilizó el planificador `-pl nnUNetPlannerResEncM`, que induce el uso de una variante basada en *Residual Encoder* (ResEnc) durante la definición del *pipeline*. Como consecuencia, los *plans* generados se referencian posteriormente mediante `-p nnUNetResEncUNetMPlans` en las etapas de entrenamiento e inferencia. En

otras palabras, `-p1` selecciona el *planificador* que construye los *plans*, mientras que `-p` especifica el *conjunto de plans* que se utilizará en el resto del *pipeline*.

Entre las decisiones más relevantes que quedan definidas en los *plans* se encuentran:

- La **configuración** de entrenamiento (p. ej., 2D, 3D a resolución completa o cascada).
- El **tamaño del parche** de entrada y, en consecuencia, el **batch size** viable.
- La **resolución objetivo** y el esquema de **remuestreo** (incluyendo el tratamiento de anisotropía).
- La **profundidad** de la red (niveles encoder/decoder) y el número de filtros por nivel.

Finalmente, se ejecuta el preprocesamiento interno conforme a los *plans*: remuestreo de imágenes y máscaras al *spacing* objetivo, recorte de regiones de fondo para reducir el tamaño efectivo del volumen y normalización de intensidades dependiente de la modalidad. El resultado de esta etapa es la generación de los artefactos de planificación y de los datos preprocesados que se utilizarán en el entrenamiento y la inferencia.

3.4.3.2 Entrenamiento del modelo

Una vez completada la planificación y el preprocesamiento interno, se procede al entrenamiento mediante:

```
nnUNetv2_train DATASET_ID CONFIGURATION FOLD -p PLANS_IDENTIFIER -tr TRAINER
```

En esta fase, el framework instancia automáticamente la arquitectura y los hiperparámetros definidos por los *plans* y entrena el modelo siguiendo validación cruzada basada en *folds* (habitualmente $k = 5$). En cada iteración, un *fold* se reserva para validación y los restantes se utilizan para entrenamiento, proporcionando una estimación más robusta del rendimiento y reduciendo el sesgo asociado a una única partición.

Para *nnUnet-v2* especifica el *plans identifier* con el argumento `-p` y valor `nnUNetResEncUNetMPlans`, es decir, el conjunto de planes generado en la fase `plan_and_preprocess` y almacenado en el entorno de resultados de nnU-Net. De este modo, se garantiza que el entrenamiento utiliza exactamente la misma configuración de arquitectura, tamaño de parche, remuestreo y normalización determinada durante la planificación. Por su parte, el planificador indicado en `-p1` durante la fase anterior define cómo se construyen dichos planes, mientras que `-p` selecciona qué planes concretos se emplean en entrenamiento e inferencia.

De forma interna, el entrenamiento incluye:

- **Entrenamiento por parches**: muestreo *patch-based* para ajustarse a memoria y favorecer el muestreo de regiones relevantes.
- **Aumento de datos en tiempo real**: transformaciones aleatorias geométricas y de intensidad para mejorar la generalización.
- **Optimización**: minimización de la función de pérdida definida por el *trainer*, típicamente con SGD con momento y programación de tasa de aprendizaje.
- **Monitorización y checkpoints**: registro de métricas y guardado de modelos intermedios para selección del modelo final y reproducibilidad.

A diferencia de *nnUnetv2* que usa el *nnUNetTrainer* por defecto, en el caso de *U-Mamba*, este paso se ejecuta especificando un *trainer* propio (*nnUNetTrainerUMambaBot*) que modifica la arquitectura (introduciendo bloques basados en *State Space Models*) manteniendo constante el resto del *pipeline* (datos, folds, preprocesamiento y evaluación), permitiendo una comparación directa bajo condiciones equivalentes.

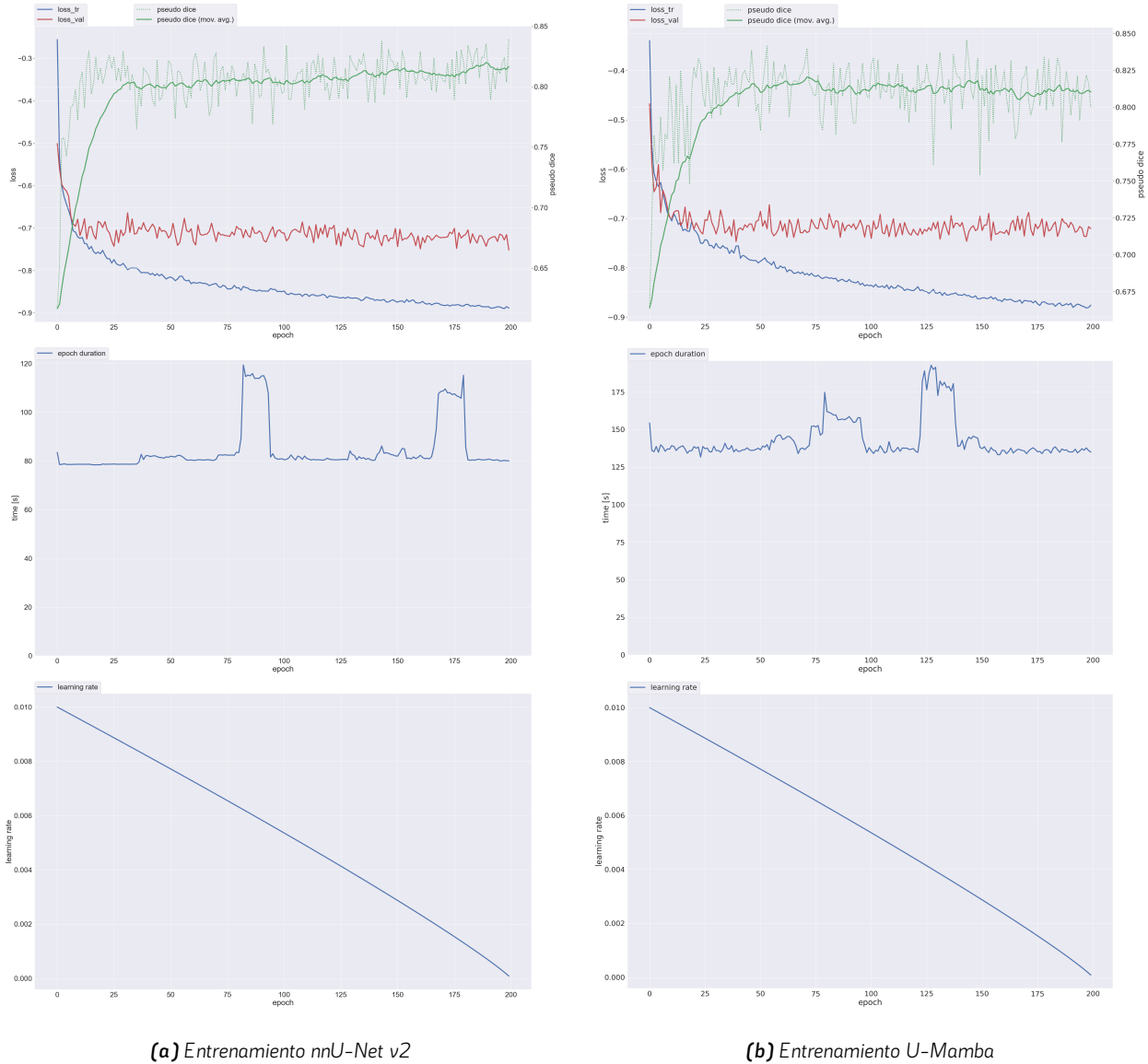


Figura 3.19. Comparación entre uno de los folds de entrenamiento del conjunto de *Dataset550_Esclerosis_MNI152*

La Figura 3.19 muestra la evolución del entrenamiento para ambos modelos. Las gráficas superiores representan la función de pérdida de entrenamiento y validación en función de las épocas, las intermedias indican el tiempo por época y las inferiores la variación del *learning rate* durante el proceso de optimización. Estas curvas permiten analizar el comportamiento de convergencia y el coste computacional del entrenamiento.

Tabla 3.5. Parámetros principales del entrenamiento en *nnU-Net v2* y su impacto. Los valores concretos dependen de los plans generados para cada dataset.

Parámetro	Dónde se define	Impacto principal
Planificador (<i>planner</i>)	<code>plan_and_preprocess(-p1)</code>	Define reglas de diseño automático (p. ej., tamaños objetivo, arquitectura base). Puede activar variantes como ResEnc <i>U-Net</i> según el planificador utilizado.
Trainer (<i>trainer</i>)	<code>-tr</code> en <code>train</code>	Especifica el "cómo" se entrena: arquitectura concreta, función de pérdida, optimizador, política de LR, augmentations y otros detalles. En <i>U-Mamba</i> , el cambio clave suele materializarse aquí.
Patch size	<i>plans</i>	Controla el campo de visión efectivo por muestra. Parches mayores capturan más contexto pero consumen más memoria y suelen reducir el batch size.
Batch size	<i>plans</i>	Afecta a estabilidad del gradiente y a la velocidad de entrenamiento. Se ajusta para encajar en memoria GPU junto con el <i>patch size</i> .
Función de pérdida (Dice + CE, etc.)	Trainer	Equilibra solapamiento global (Dice) y penalización local voxel a voxel (CE). Relevante en desbalance de clases y lesiones pequeñas.
Optimizador y <i>learning rate schedule</i>	Trainer	Condiciona la convergencia y estabilidad del entrenamiento. La política de LR impacta en rapidez y calidad del mínimo alcanzado.
Folds (<i>k</i>)	Definición de splits (<i>nnU-Net</i>)	Aporta robustez en la estimación del rendimiento. Permite entrenar múltiples modelos comparables y, si se desea, combinar predicciones multi-fold.

3.4.3.3 Selección de la mejor configuración

Una vez entrenados los distintos *folds* y (en caso de haberse ejecutado) múltiples configuraciones, *nnU-Net v2* permite seleccionar automáticamente la opción con mejor rendimiento promedio en validación cruzada mediante:

```
nnUNetv2_find_best_configuration DATASET_ID -c CONFIGURATIONS -p PLANS_IDENTIFIER
```

Este paso agrega los resultados de validación de los *folds* para cada configuración evaluada (p. ej., `2d`, `3d_fullres`,

3d_lowres/cascada) y selecciona aquella que maximiza la métrica objetivo (habitualmente el Dice medio) en el conjunto de validación. De este modo, se evita una selección manual y se mantiene un criterio reproducible basado en resultados cuantitativos.

En este trabajo no se utilizó la selección automática de una única configuración final. En su lugar, se entrenaron y evaluaron dos configuraciones representativas del framework: 2d y 3d_fullres. Posteriormente, se realizaron inferencias completas sobre todos los casos del conjunto de prueba con ambas configuraciones, permitiendo una comparación directa de su comportamiento.

Por tanto, la herramienta `nnUNetv2_find_best_configuration` se empleó únicamente como referencia informativa para conocer el rendimiento esperado según la validación cruzada, pero no como criterio de selección del modelo final. De esta forma, el análisis experimental se basa en la comparación objetiva entre aproximaciones bidimensionales y tridimensionales, evaluando sus diferencias en términos de precisión, robustez y capacidad de detección de lesiones.

3.4.3.4 Inferencia

La inferencia sobre nuevos datos se realiza mediante el comando:

```
nnUNetv2_predict -i INPUT_FOLDER -o OUTPUT_FOLDER -d DATASET_ID -c CONFIGURATION \  
-f FOLDS -p PLANS_IDENTIFIER -tr TRAINER
```

donde `-i` y `-o` indican, respectivamente, la carpeta de entrada (habitualmente `imagesTs`) y la carpeta donde se guardarán las predicciones; `-d` identifica el dataset; y `-c` especifica la configuración empleada (por ejemplo, 2d). Adicionalmente, `-f` permite seleccionar los *folders* cuyos modelos se utilizarán para predecir (p.ej., 0 1 2 3 4), mientras que `-p` y `-tr` garantizan que se cargan los mismos *plans* y el mismo *trainer* utilizados durante el entrenamiento.

Durante la inferencia, el modelo procesa cada volumen mediante un esquema de *sliding window* sobre parches potencialmente solapados. Este enfoque permite segmentar volúmenes completos sin exceder la memoria de la GPU. Las predicciones parciales se agregan (habitualmente promediando probabilidades en las zonas solapadas) para obtener un mapa final por voxel, que posteriormente se discretiza en etiquetas.

Cuando se utilizan múltiples *folders*, la inferencia puede ejecutarse de forma *multi-fold*: se generan predicciones con cada modelo entrenado y, según la configuración del *pipeline*, se combinan para producir una salida más robusta. En este trabajo, esta estrategia se utilizó para mantener consistencia con la validación cruzada, almacenando las predicciones resultantes en el directorio de salida especificado.

En el caso de *U-Mamba*, la inferencia sigue exactamente el mismo procedimiento y comandos, diferenciándose únicamente en la arquitectura interna del modelo cargado (definida por el *trainer* empleado durante el entrenamiento).

3.4.3.5 Predicción multi-fold y consolidación de resultados

En este trabajo, la inferencia se realizó utilizando las predicciones de los distintos *folders* entrenados, con el objetivo de obtener resultados más robustos y reducir la variabilidad asociada a una partición concreta de validación. Para ello, se empleó el comando:

```
nnUNetv2_predict -i INPUT_FOLDER -o OUTPUT_FOLDER -d DATASET_ID -c CONFIGURATION \  
-f 0 1 2 3 4 -tr TRAINER -p PLANS_IDENTIFIER
```

El parámetro `-f` permite especificar qué modelos (folds) se utilizarán durante la predicción. En este caso se emplearon los cinco folds (0 1 2 3 4), de modo que el framework genera una predicción por fold sobre cada volumen del conjunto de test. Las predicciones generadas por cada fold pueden consolidarse posteriormente de dos formas:

- **Selección directa:** utilizar la salida de un fold concreto (menos habitual en evaluación final).
- **Consolidación por combinación:** combinar las predicciones de todos los folds, típicamente promediando las probabilidades (*softmax*) y generando después la segmentación final. Esta estrategia suele mejorar la estabilidad del resultado y reducir falsos positivos aislados.

En la práctica, esta fase permite aprovechar el entrenamiento con validación cruzada no solo como método de estimación del rendimiento, sino también como mecanismo para construir un predictor más robusto a partir de varios modelos entrenados.

3.4.3.6 Ensamblado y postprocesamiento

Una vez generadas las predicciones, *nnU-Net v2* permite aplicar etapas opcionales para mejorar la robustez y la coherencia espacial de las segmentaciones: (i) **ensamblado** (*ensemble*) y (ii) **postprocesamiento**.

Ensamblado (*ensemble*).

El *ensemble* consiste en combinar las salidas de **distintos modelos** (por ejemplo, entrenados con diferentes configuraciones, planificadores o arquitecturas) para obtener una predicción final más estable. Para ello se utiliza:

```
nnUNetv2_ensemble -i FOLDER1 FOLDER2 -o OUTPUT_FOLDER
```

En términos prácticos, este comando combina las probabilidades (*softmax*) generadas por cada modelo y produce una segmentación final a partir de la fusión. Aunque es una característica estándar del framework, en este trabajo se priorizó la comparación directa entre configuraciones (y la predicción multi-fold) sin realizar ensamblados entre modelos distintos.

Postprocesamiento.

De manera adicional, *nnU-Net v2* puede aplicar un postprocesamiento automático basado en reglas simples, típicamente aprendido/seleccionado a partir de los resultados de validación cruzada. Este paso se ejecuta con:

```
nnUNetv2_apply_postprocessing -i INPUT -o OUTPUT --pp_pkl_file FILE \  
-np N --plans_json PLANS_JSON
```

donde `--pp_pkl_file` apunta al archivo `postprocessing.pkl` generado durante la validación cruzada (habitualmente en `crossval_results_folds_...`). El postprocesamiento puede incluir, entre otras operaciones:

- **Eliminación de componentes conectadas pequeñas** (reducción de falsos positivos aislados).
- **Selección del componente conectado principal** cuando la anatomía o la tarea lo justifican.
- **Reglas por clase** (si el problema es multiclase), aplicando criterios diferentes según la etiqueta.

Estas operaciones buscan mejorar la plausibilidad anatómica de la segmentación final sin modificar el modelo, actuando únicamente sobre la máscara predicha.

3.4.3.7 Evaluación sobre el conjunto de test

Una vez completado el proceso de inferencia (y, en su caso, el postprocesamiento), se realizó una evaluación cuantitativa de las segmentaciones obtenidas sobre el conjunto de test mediante la herramienta `nnUNetv2_evaluate_folder`. Esta utilidad compara de forma directa las predicciones generadas por el modelo con las máscaras de referencia (*ground truth*), proporcionando métricas objetivas de rendimiento en datos no utilizados durante el entrenamiento.

De forma general, el comando empleado para llevar a cabo esta evaluación presenta la siguiente estructura:

```
nnUNetv2_evaluate_folder \
  <ruta_labelsTs> <ruta_predicciones> \
  -djfile <ruta_dataset.json> \
  -pfile <ruta_plans.json> \
  -o <ruta_salida.json>
```

En este comando, el primer argumento corresponde al directorio que contiene las etiquetas reales del conjunto de test (`labelsTs`), mientras que el segundo argumento indica la carpeta donde se almacenan las segmentaciones predichas por el modelo tras el proceso de inferencia. A partir de estas dos rutas, la herramienta evalúa caso a caso la correspondencia entre las máscaras reales y las predicciones generadas.

El parámetro `-djfile` hace referencia al archivo `dataset.json`, que define la estructura del conjunto de datos, las modalidades empleadas y las clases de segmentación. Este archivo permite a *nnU-Net v2* interpretar correctamente las etiquetas y asociar cada clase con su identificador numérico.

Por otro lado, el parámetro `-pfile` apunta al archivo de *plans* utilizado en el experimento (típicamente un `plans.json` dentro de `nnUNet_results` o, en algunos casos, el `nnUNetPlans.json` generado en `nnUNet_preprocessed`). Dicho archivo contiene información clave sobre el *spacing* objetivo, la configuración interna y transformaciones del preprocesamiento, garantizando que la evaluación se realice en condiciones consistentes con el *pipeline* utilizado.

Finalmente, `-o` permite especificar un fichero JSON de salida donde se almacenan las métricas agregadas de la evaluación, facilitando la comparación sistemática entre experimentos y la trazabilidad de resultados.

Internamente, `nnUNetv2_evaluate_folder` recorre todos los casos del conjunto de test y calcula métricas de segmentación comparando voxel a voxel las máscaras predichas con las reales. Entre las métricas más relevantes se encuentra el *Dice Similarity Coefficient* (DSC). En función de la configuración del dataset y del problema, también pueden calcularse métricas complementarias relacionadas con precisión, sensibilidad (*recall*) y otros indicadores derivados.

Esta evaluación se realiza de forma independiente al proceso de entrenamiento y a la validación cruzada, proporcionando una estimación más realista del rendimiento del modelo en datos no vistos. En este trabajo, este procedimiento permitió comparar de manera homogénea *nnU-Net v2* y *U-Mamba* bajo el mismo conjunto de test y las mismas métricas.

3.4.3.8 Diferencias clave al emplear *U-Mamba*

Aunque el flujo general del *pipeline* es el mismo para *nnU-Net v2* y *U-Mamba*, las diferencias se concentran en la **definición del modelo dentro del trainer**. En particular, *U-Mamba* sustituye parte del procesamiento puramente convolucional por bloques basados en *State Space Models* (SSM), lo que incrementa la capacidad de modelado de dependencias de largo alcance sin recurrir a mecanismos de atención global.

En la práctica, esta diferencia se traduce en que:

- El **preprocesamiento y la planificación** se mantienen dentro del marco de *nnU-Net v2*, de modo que la estructura de datos, el remuestreo, el recorte y la normalización interna son comparables.
- El **comando de entrenamiento e inferencia** se ejecuta de forma análoga, pero especificando el **trainer de *U-Mamba***, que es el componente que define la arquitectura concreta y su configuración.
- Opcionalmente, pueden variar **detalles de inferencia** como el uso de *test-time augmentation* (TTA), que en algunos experimentos puede deshabilitarse para controlar el tiempo de ejecución o mantener condiciones homogéneas.

Gracias a esta integración directa, la comparación entre ambos enfoques se realiza en un entorno experimental controlado: se mantienen constantes el conjunto de datos, la partición en folds, el preprocesamiento y las métricas de evaluación, aislando el efecto de la arquitectura (y del trainer) como factor principal de diferencia. Esto permite una evaluación más justa y reproducible del impacto de *U-Mamba* frente a un baseline consolidado como *nnU-Net v2*.

4

Resultados

En este capítulo se presentan los resultados obtenidos en los experimentos de segmentación realizados con *nnU-Net v2* y *U-Mamba*. Los entrenamientos se llevaron a cabo sobre dos conjuntos de datos (esclerosis múltiple y metástasis cerebrales), evaluando el impacto de la normalización espacial mediante tres plantillas anatómicas (**MNI152**, **MRI225** y **SRI24**). En todos los casos, se mantuvieron constantes el resto de componentes del *pipeline* (formato de datos, partición *train/test*, configuraciones 2D y 3D_fullres, folds, y procedimiento de inferencia y evaluación), con el fin de garantizar comparaciones homogéneas.

4.1 Resumen de experimentos realizados

Para facilitar la trazabilidad de los experimentos, la Tabla 4.1 resume las combinaciones evaluadas. En total se propone realizar **24 entrenamientos**: 6 con *nnU-Net v2* y 6 con *U-Mamba* (2 datasets × 3 plantillas × 2 configuraciones × 2 algoritmos).

Tabla 4.1. Resumen de entrenamientos realizados por dataset, plantilla, configuración y arquitectura.

Dataset	Plantilla	Configuración	<i>nnU-Net v2</i>	<i>U-Mamba</i>
Esclerosis múltiple	MNI152	2D	✓	✓
Esclerosis múltiple	MNI152	3D_fullres	✓	✓
Esclerosis múltiple	MRI225	2D	✓	✓
Esclerosis múltiple	MRI225	3D_fullres	✓	✓
Esclerosis múltiple	SRI24	2D	✓	✓
Esclerosis múltiple	SRI24	3D_fullres	✓	✓
Metástasis	MNI152	2D	–	–
Metástasis	MNI152	3D_fullres	–	–
Metástasis	MRI225	2D	✓	–
Metástasis	MRI225	3D_fullres	✓	✓
Metástasis	SRI24	2D	–	–
Metástasis	SRI24	3D_fullres	–	–

✓: experimento realizado –: experimento no ejecutado por limitaciones computacionales

4.2 Estructura de la sección de resultados

Los resultados se organizan en dos bloques principales, uno por conjunto de datos. Dentro de cada bloque se analizan de forma progresiva tres factores experimentales:

- El efecto de la **plantilla espacial** (MNI152, MRI225 y SRI24), manteniendo fija la arquitectura y la configuración.
- La comparación entre **configuraciones 2D y 3D_fullres**, con el objetivo de estudiar la influencia del contexto volumétrico en la segmentación.
- La comparación entre ***nnU-Net v2* y *U-Mamba*** bajo las mismas condiciones experimentales.

Para cada experimento se reportan métricas de solapamiento calculadas sobre el conjunto de test, principalmente el **Dice Similarity Coefficient (DSC)** y el **Intersection over Union (IoU)**. Adicionalmente, se incluyen valores derivados de la matriz de confusión a nivel de vóxel (TP, FN, FP, n_{ref} y n_{pred}), que permiten interpretar el comportamiento del modelo en términos de sobresegmentación e infradetección.

Con el fin de complementar la interpretación de los resultados, cuando resulta relevante se analiza también la variabilidad entre casos, destacando la presencia de fallos completos de segmentación o degradaciones significativas del rendimiento en lesiones pequeñas.

4.3 Resultados en esclerosis múltiple

En esta sección se analizan los resultados obtenidos sobre el conjunto de esclerosis múltiple (MSLesSeg). Se evalúan los modelos *nnU-Net v2* y *U-Mamba* bajo tres espacios de referencia anatómicos (**MRI225**, **MNI152** y **SRI24**) y dos configuraciones de entrenamiento (2D y 3D_fullres).

El objetivo es estudiar: (i) el impacto de la dimensionalidad del modelo, (ii) la influencia de la plantilla anatómica y (iii) las diferencias entre arquitecturas.

4.3.1 Resultados cuantitativos globales

La Tabla 4.2 resume los resultados promedio obtenidos en el conjunto de test.

Tabla 4.2. Resultados para el dataset de esclerosis múltiple.

Modelo	Plantilla	Config.	Dice	IoU	TP	FN	FP	n_{ref}	n_{pred}
<i>nnU-Net v2</i>	MRI225	2D	0.6408	0.4922	7810.93	5141.67	1513.73	12952.60	9324.67
<i>nnU-Net v2</i>	MNI152	2D	0.6455	0.4980	8018.00	4934.60	1633.13	12952.60	9651.13
<i>nnU-Net v2</i>	SRI24	2D	0.6431	0.4949	7900.13	5052.47	1617.93	12952.60	9518.07
<i>nnU-Net v2</i>	MRI225	3D	0.7196	0.5728	8958.60	3994.00	1878.40	12952.60	10837.00
<i>nnU-Net v2</i>	MNI152	3D	0.7172	0.5699	8929.00	4023.60	1799.53	12952.60	10728.53
<i>nnU-Net v2</i>	SRI24	3D	0.7123	0.5644	8845.47	4107.13	1831.87	12952.60	10677.33
<i>U-Mamba</i>	MRI225	2D	0.6524	0.5036	8234.93	4717.67	1787.47	12952.60	10022.40
<i>U-Mamba</i>	MNI152	2D	0.6502	0.5017	8241.33	4711.27	1839.00	12952.60	10080.33
<i>U-Mamba</i>	SRI24	2D	0.6516	0.5042	8304.80	4647.80	1819.73	12952.60	10124.53
<i>U-Mamba</i>	MRI225	3D	0.7214	0.5756	9332.20	3620.40	2249.13	12952.60	11581.33
<i>U-Mamba</i>	MNI152	3D	0.7218	0.5761	9339.93	3612.67	2198.20	12952.60	11538.13
<i>U-Mamba</i>	SRI24	3D	0.7205	0.5755	9270.53	3682.07	2127.67	12952.60	11398.20

4.3.2 Impacto de la dimensionalidad

El factor más determinante observado es la dimensionalidad del modelo. El entrenamiento volumétrico (3D) mejora el coeficiente Dice aproximadamente entre 0.07 y 0.08 respecto a los modelos 2D en ambas arquitecturas.

Este comportamiento indica que las lesiones de esclerosis múltiple presentan coherencia espacial tridimensional y que el contexto volumétrico resulta crítico para su correcta segmentación.

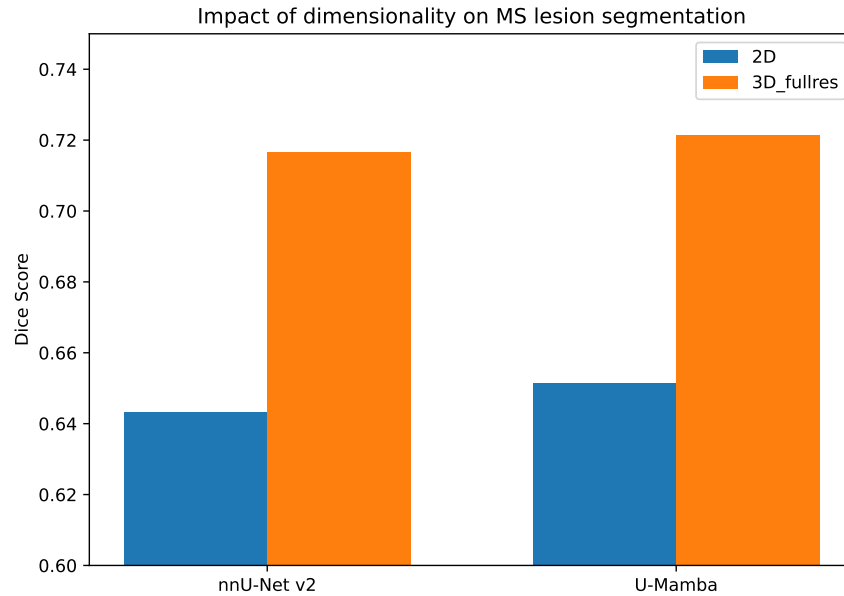


Figura 4.1. Comparación del rendimiento medio (Dice) entre modelos 2D y 3D.

4.3.3 Impacto de la plantilla anatómica

Las diferencias entre plantillas son reducidas (menores a 0.01 Dice en todos los casos). Esto sugiere que la normalización espacial no constituye el factor dominante en este dataset, probablemente debido al carácter pequeño, disperso y heterogéneo de las lesiones.

Por tanto, la elección de plantilla no altera significativamente el rendimiento global del sistema.

4.3.4 Comparación entre modelos

Bajo la configuración óptima (3D_fullres), *U-Mamba* obtiene sistemáticamente valores ligeramente superiores a *nnU-Net v2* en todas las plantillas. La mejora observada es pequeña pero consistente, situándose aproximadamente entre 0.002 y 0.008 Dice.

Este resultado indica que ambas arquitecturas convergen hacia rendimientos muy similares, aunque los bloques basados en *State Space Models* aportan una ligera ventaja en la modelización del contexto global.

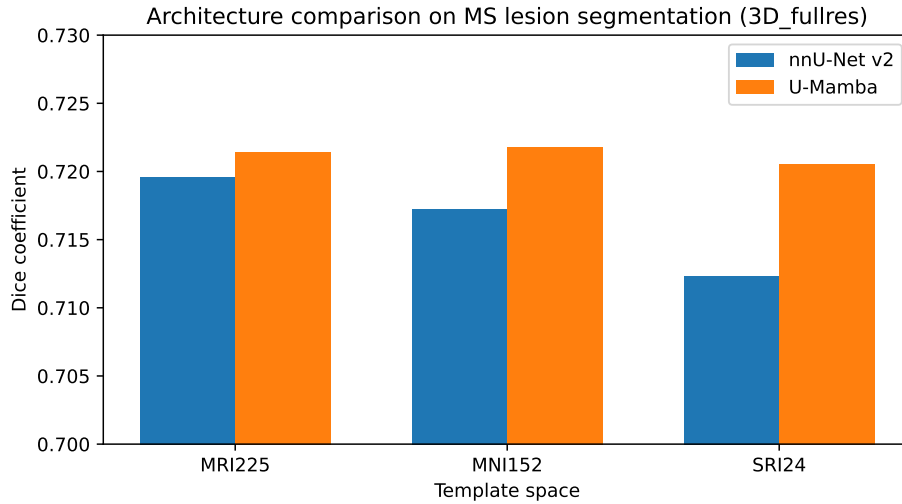


Figura 4.2. Comparación del rendimiento medio entre nnU-Net v2 y U-Mamba.

4.3.5 Análisis cualitativo

En esta sección se analizan cualitativamente distintos sujetos representativos con el fin de complementar la evaluación cuantitativa. Se estudian casos de dificultad creciente para identificar patrones de error, diferencias estructurales entre modelos y el impacto del contexto espacial en la segmentación.

Caso fácil (00038)

En el caso fácil (00038), caracterizado por lesiones extensas y bien delimitadas, ambos modelos logran segmentaciones cercanas a la referencia manual. Sin embargo, *U-Mamba* obtiene un Dice ligeramente superior (≈ 0.82 frente a ≈ 0.80), generando regiones más suaves y continuas.

nnU-Net v2 produce una segmentación correcta pero con mayor fragmentación en los bordes, mientras que *U-Mamba* mantiene mejor la coherencia espacial. Esto sugiere que el modelado de contexto global aporta una ventaja cuando la lesión es claramente visible.

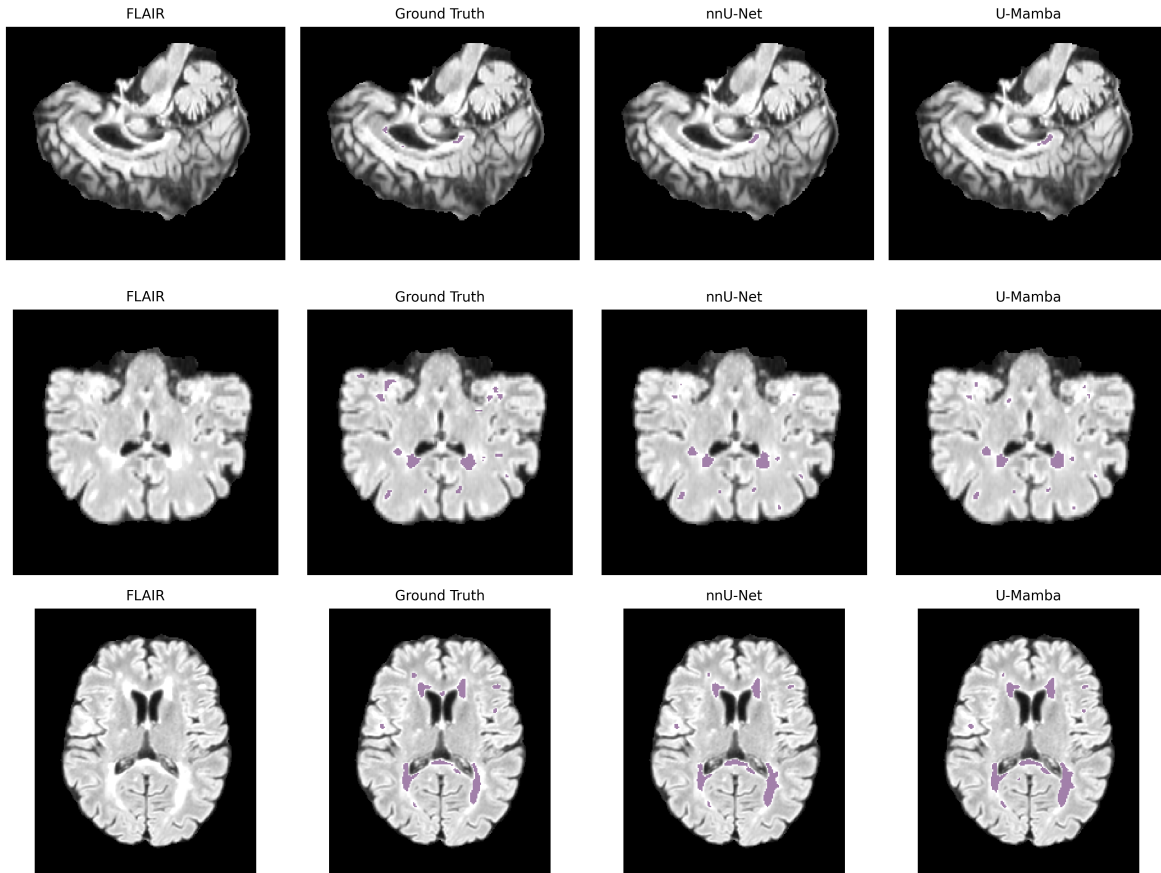


Figura 4.3. Caso fácil: lesiones extensas y bien delimitadas.

Caso intermedio (00067)

En el caso intermedio (00067), donde las lesiones presentan menor contraste y mayor dispersión espacial, aparecen diferencias más claras entre arquitecturas. *U-Mamba* alcanza un Dice superior (≈ 0.77 frente a ≈ 0.75), generando regiones más continuas y menos fragmentadas.

nnU-Net v2 detecta la mayor parte de la lesión pero con mayor discontinuidad espacial, mientras que *U-Mamba* integra mejor el contexto tridimensional. Este comportamiento indica que el modelado global resulta especialmente beneficioso cuando la lesión no está perfectamente definida.

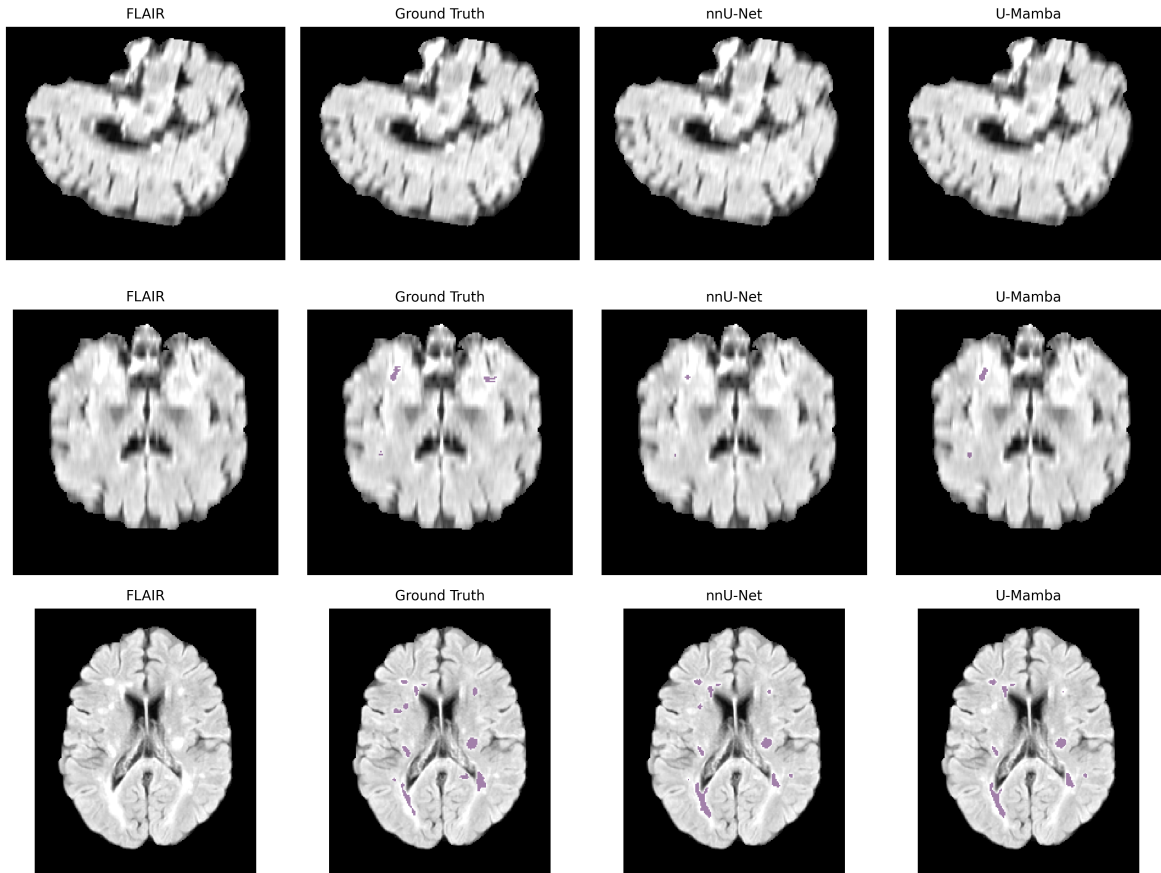


Figura 4.4. Caso intermedio: mayor coherencia espacial en U-Mamba.

Caso difícil (00027)

Finalmente, en el caso difícil (00027), correspondiente a lesiones pequeñas y de bajo contraste, se observa el escenario más desafiante del dataset. La Figura 4.5 muestra cómo ambos modelos presentan una elevada tasa de falsos negativos, lo que confirma que la principal dificultad del problema radica en la detección de lesiones sutiles más que en la delimitación de lesiones extensas.

En este caso, *nnU-Net v2* obtiene un rendimiento ligeramente superior (Dice \approx 0.55) frente a *U-Mamba* (Dice \approx 0.52). Mientras que *U-Mamba* tiende a suavizar la predicción eliminando regiones dudosas, *nnU-Net v2* conserva una mayor proporción de vóxeles lesionados, aunque a costa de mayor fragmentación espacial.

La caída significativa del Dice respecto a los casos anteriores confirma que la limitación principal del dataset no es la segmentación de contornos complejos, sino la propia detectabilidad de lesiones pequeñas y de bajo contraste.

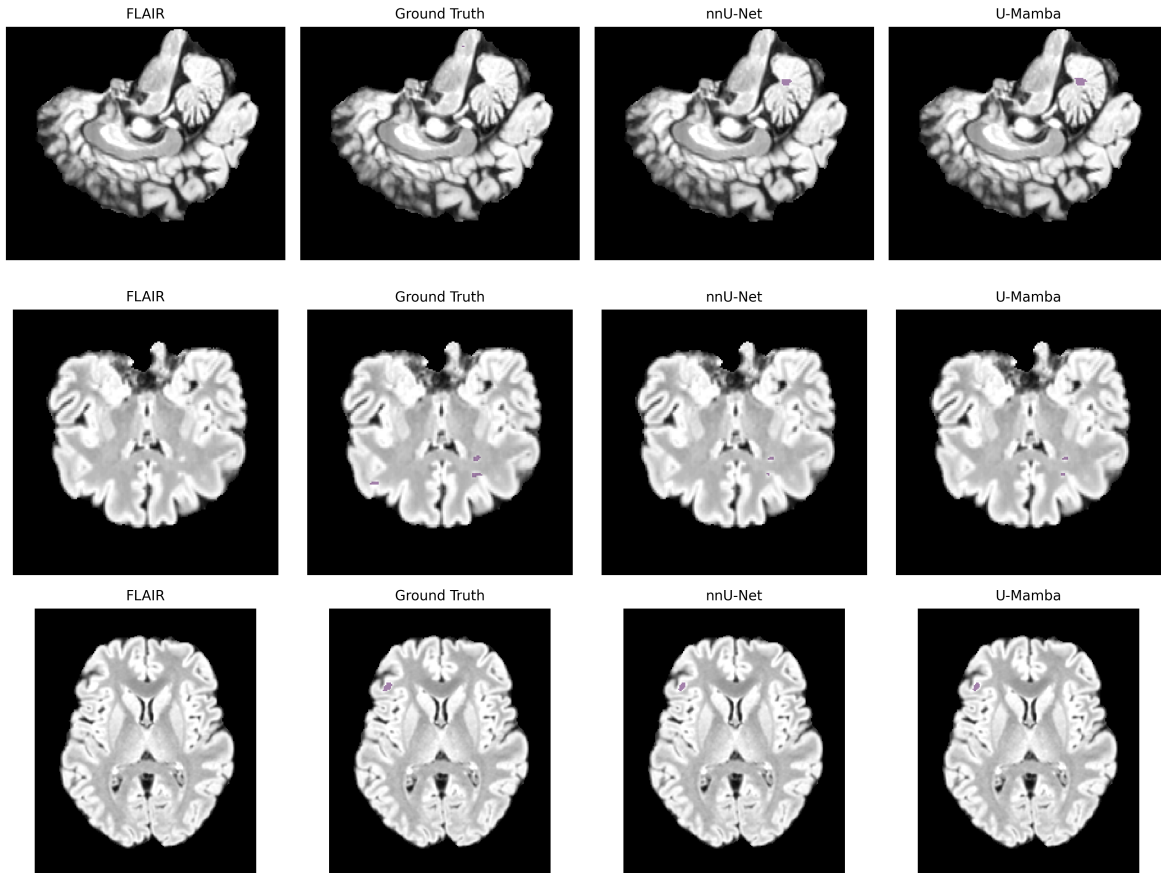


Figura 4.5. Caso difícil: fallos por bajo contraste y lesiones pequeñas.

Para la visualización cualitativa se seleccionó automáticamente el corte correspondiente al **centro de masa** de la lesión en la máscara de referencia. De este modo se evita la selección manual de ejemplos y se garantiza la reproducibilidad del análisis.

4.3.6 Resumen

Los experimentos permiten extraer tres conclusiones principales:

- La dimensionalidad es el factor dominante: los modelos 3D superan claramente a los 2D.
- La plantilla anatómica tiene impacto mínimo en la segmentación de lesiones.
- *U-Mamba* y *nnU-Net v2* presentan rendimientos muy similares, con ligera ventaja para *U-Mamba*.

El mejor resultado global se obtiene con ***U-Mamba 3D_fullres*** en MNI152 (Dice = 0.7218).

4.4 Resultados en metástasis

En esta sección se analizan los resultados obtenidos sobre el conjunto de metástasis cerebrales. A diferencia del dataset de esclerosis múltiple, en este caso únicamente se dispone de experimentos en el espacio de referencia **MRI255**, evaluando configuraciones 2D y 3D_fullres para *nnU-Net v2* y únicamente 3D_fullres para *U-Mamba*.

El objetivo principal es estudiar el comportamiento de ambas arquitecturas en un escenario con lesiones de mayor tamaño y contraste, pero con elevada variabilidad morfológica y presencia de múltiples focos tumorales por paciente.

4.4.1 Resultados cuantitativos globales

La Tabla 4.3 resume los resultados promedio obtenidos en el conjunto de test.

Tabla 4.3. Resultados para el dataset de metástasis cerebrales.

Modelo	Config.	Dice	IoU	TP	FN	FP	n_{ref}	n_{pred}
<i>nnU-Net v2</i>	2D	0.4207	0.3317	4560.80	5600.47	1490.00	10161.27	6050.80
<i>nnU-Net v2</i>	3D	0.5379	0.4575	6224.20	3937.07	2186.13	10161.27	8410.33
<i>U-Mamba</i>	3D	0.7506	0.6656	8454.73	1706.53	2822.20	10161.27	11276.93

Se observa una diferencia sustancial entre arquitecturas: *U-Mamba* supera a *nnU-Net v2* en aproximadamente **+0.21 Dice**, lo que indica un comportamiento claramente distinto al observado en el dataset de esclerosis múltiple.

4.4.2 Impacto de la dimensionalidad

El paso de 2D a 3D en *nnU-Net v2* produce una mejora notable (aproximadamente +0.12 Dice). Este incremento es significativamente mayor que el observado en esclerosis múltiple, lo que indica que la segmentación de metástasis depende en mayor medida del contexto volumétrico.

Las metástasis presentan continuidad espacial más definida y estructuras compactas, por lo que la información tridimensional resulta crítica para diferenciar correctamente tejido tumoral de estructuras anatómicas adyacentes.

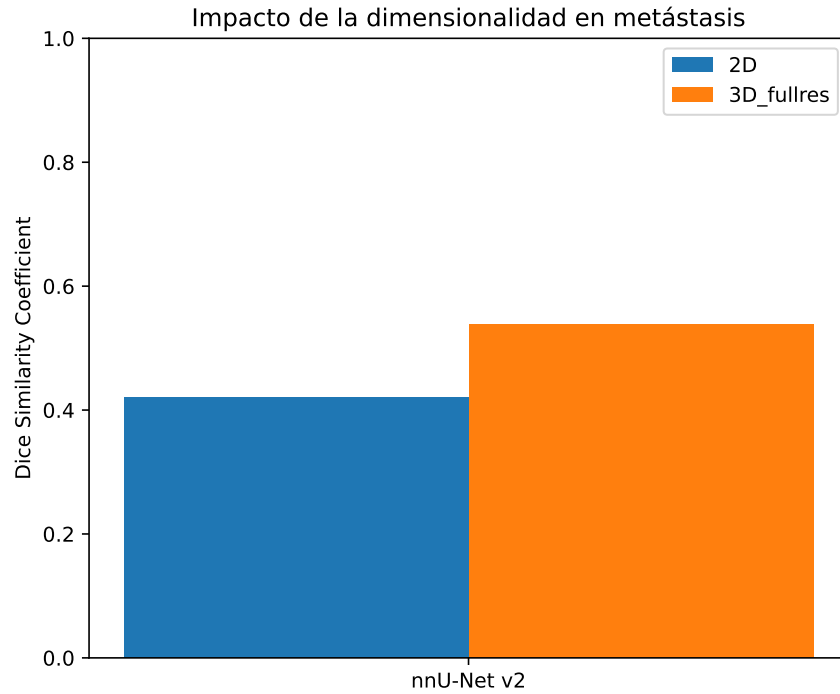


Figura 4.6. Comparación del rendimiento medio (Dice) entre configuraciones 2D y 3D en metástasis.

4.4.3 Comparación entre modelos

La diferencia entre modelos es considerablemente mayor que en el dataset anterior. Mientras que en esclerosis múltiple ambos métodos convergían hacia resultados similares, en metástasis *U-Mamba* muestra una mejora clara y consistente.

El aumento simultáneo de TP junto con un descenso significativo de FN indica que *U-Mamba* detecta un mayor número de lesiones reales, aunque a costa de un ligero incremento de FP. Esto sugiere un comportamiento más sensible, reduciendo la infradetección presente en *nnU-Net v2*.

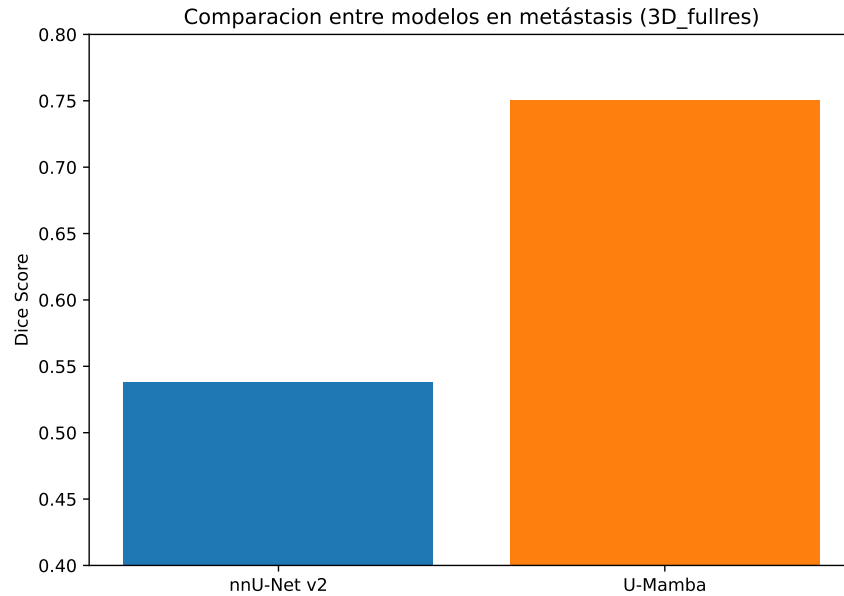


Figura 4.7. Comparación del rendimiento medio entre nnU-Net v2 y U-Mamba en metástasis cerebrales.

4.4.4 Análisis cualitativo

Para complementar el análisis cuantitativo se seleccionaron tres sujetos representativos que ilustran comportamientos característicos del problema: un caso sencillo, un fallo estructural de detección y un escenario clínico difícil con múltiples lesiones pequeñas.

Caso sencillo (00020)

En este caso ambos modelos segmentan correctamente la lesión principal, obteniendo valores de Dice muy elevados (≈ 0.94). La lesión presenta alto contraste y bordes definidos, por lo que el problema queda dominado por la resolución espacial más que por la arquitectura.

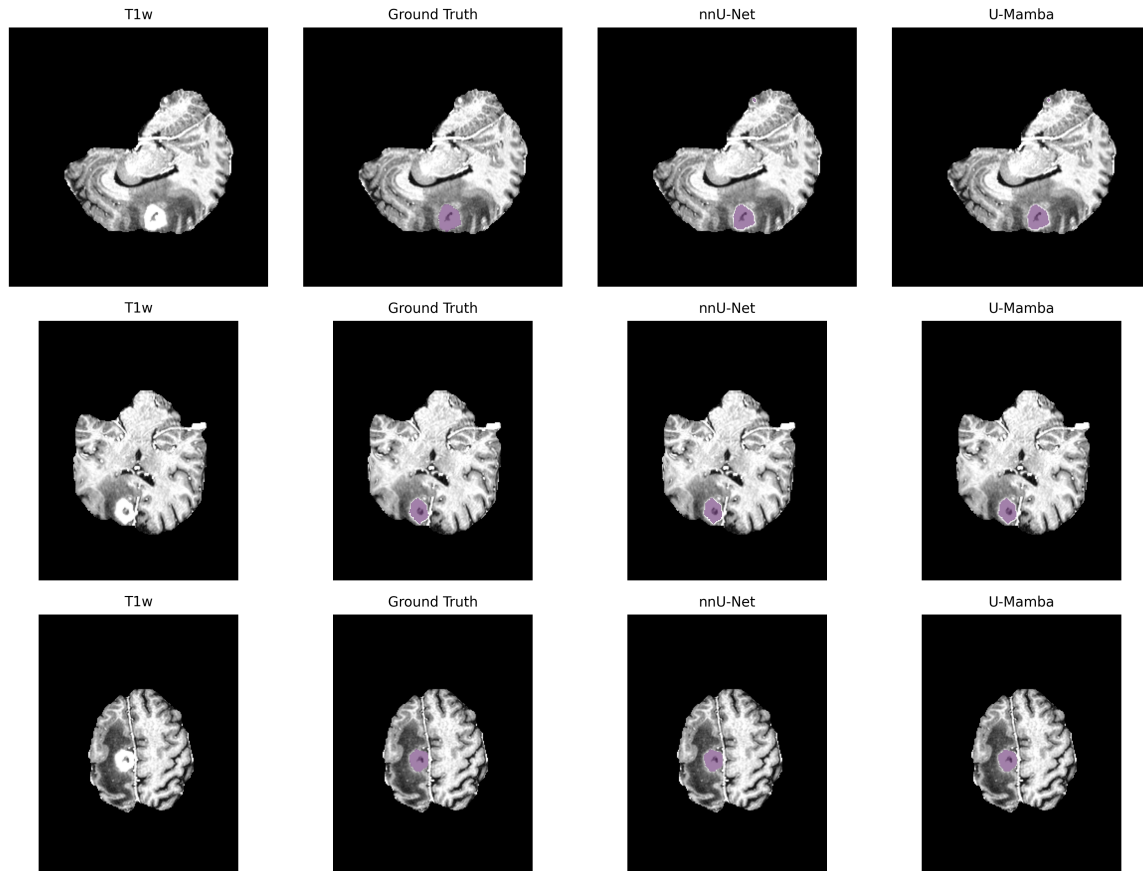


Figura 4.8. Caso sencillo: ambos modelos segmentan correctamente la lesión principal.

Este ejemplo actúa como control experimental y confirma que las diferencias observadas entre modelos no se deben a problemas de calidad de imagen o anotación.

Fallo de detección en CNN (00032)

Este sujeto muestra una diferencia cualitativa crítica entre arquitecturas. *nnU-Net v2* no detecta la lesión (Dice = 0), mientras que *U-Mamba* la segmenta correctamente (Dice ≈ 0.94).

Este comportamiento indica que el modelo convolucional falla al capturar dependencias globales necesarias para identificar la estructura tumoral, produciendo un fallo completo de detección. Por el contrario, el modelado de contexto global en *U-Mamba* permite recuperar la lesión.

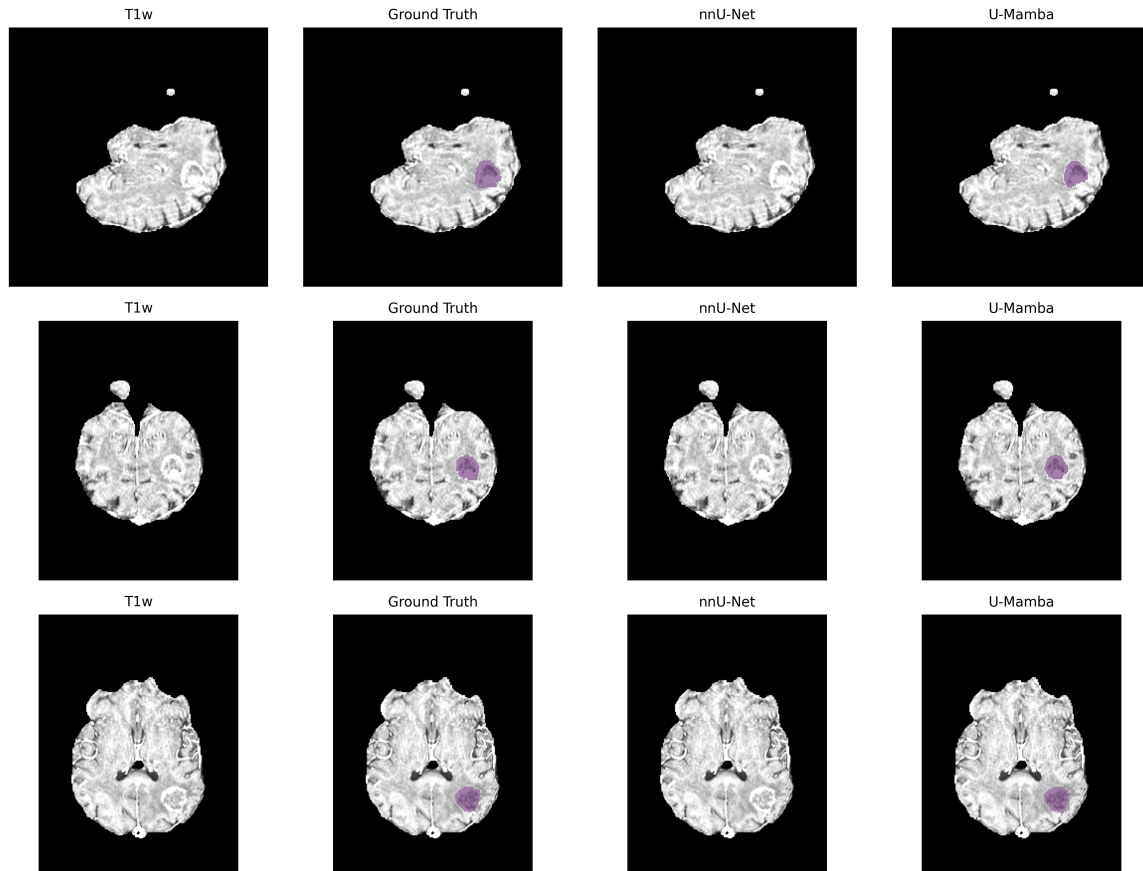


Figura 4.9. Fallo estructural de detección: *nnU-Net v2* no detecta la lesión mientras *U-Mamba* sí la segmenta correctamente.

Este tipo de fallo explica la gran diferencia de rendimiento medio observada en el dataset de metástasis.

Escenario clínico complejo (00041)

Este caso representa un escenario límite caracterizado por múltiples focos extremadamente pequeños cercanos al umbral de resolución de la imagen. En esta situación el problema deja de ser puramente de segmentación y pasa a ser principalmente de detectabilidad.

Ambos modelos muestran una degradación severa del rendimiento. *nnU-Net v2* presenta fallos completos de detección (Dice ≈ 0 en 2D y ≈ 0.06 en 3D), mientras que *U-Mamba* alcanza un valor ligeramente superior (Dice ≈ 0.08) al identificar parcialmente alguna región compatible.

Sin embargo, se observa un fenómeno relevante: las predicciones no solo omiten lesiones reales, sino que también activan regiones anatómicas con textura similar al tejido tumoral. Esto indica que, en el límite de resolución, el problema dominante no es la delimitación de contornos sino la discriminación entre patrón tumoral y variabilidad estructural normal.

El comportamiento de *U-Mamba* sugiere una mayor sensibilidad al contexto global, mientras que *nnU-Net v2* resulta más conservador. No obstante, ninguno de los modelos logra resolver completamente el caso, confirmando que la limitación principal del dataset en metástasis no es la complejidad morfológica sino la propia detectabilidad de lesiones cercanas al límite de resolución efectiva de la imagen.

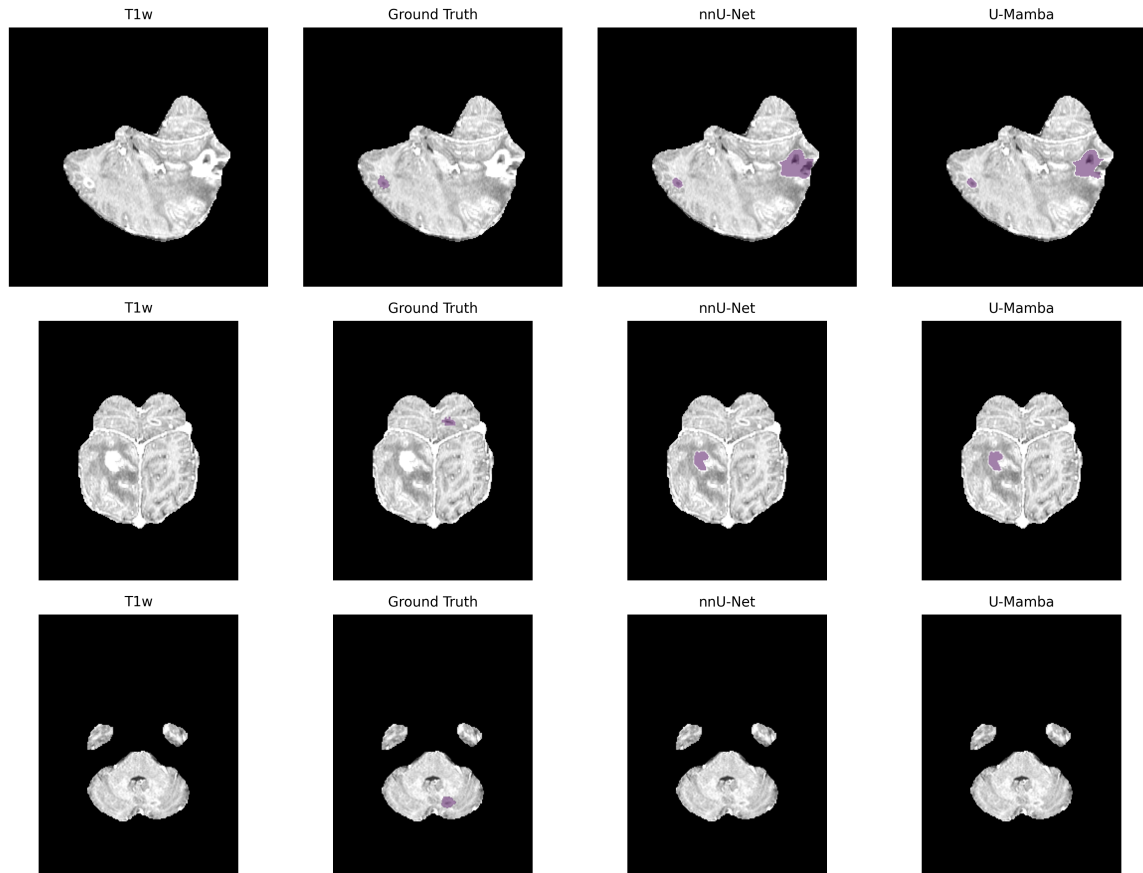


Figura 4.10. Caso complejo: lesiones cercanas al límite de resolución con aparición de falsos positivos estructurados.

4.4.5 Resumen

Los experimentos permiten extraer cuatro conclusiones principales:

- El contexto volumétrico es crítico: el modelo 3D mejora significativamente al 2D.
- Existe alta variabilidad entre pacientes debido al número y tamaño de lesiones.
- *nnU-Net v2* presenta tendencia a la infradetección.
- *U-Mamba* muestra una mejora sustancial en sensibilidad y rendimiento global.

El mejor resultado global se obtiene con *U-Mamba 3D_fullres* (Dice = 0.7506).

4.5 Discusión comparativa

En esta sección se analizan conjuntamente los resultados obtenidos en los dos conjuntos de datos considerados —esclerosis múltiple y metástasis cerebrales— con el objetivo de interpretar el comportamiento de las arquitecturas evaluadas más allá de las métricas numéricas aisladas.

4.5.1 Influencia de la dimensionalidad

El efecto más consistente observado en ambos datasets es la influencia de la dimensionalidad del modelo. En todos los experimentos, las configuraciones **3D_fullres** superan claramente a sus equivalentes **2D**, independientemente de la arquitectura empleada.

Este comportamiento era esperable desde un punto de vista clínico y geométrico: las lesiones cerebrales son estructuras volumétricas cuya continuidad espacial entre cortes contiene información relevante para su identificación. Los modelos 2D procesan cada corte de forma independiente, perdiendo coherencia inter-corte, mientras que los modelos 3D incorporan contexto anatómico completo.

No obstante, el impacto de la dimensionalidad no es igual en ambos datasets:

- En **esclerosis múltiple**, la mejora es moderada pero consistente. Las lesiones suelen ser pequeñas y numerosas, distribuidas por la sustancia blanca, por lo que el contexto local sigue siendo suficiente en muchos casos.
- En **metástasis**, la mejora es mucho más marcada. Las lesiones presentan mayor variabilidad morfológica y aparecen como masas tridimensionales bien definidas, por lo que el contexto volumétrico resulta crítico para evitar falsos negativos y fragmentaciones.

Esto confirma que la elección entre 2D y 3D depende en gran medida de la naturaleza espacial de la patología, más que de la arquitectura concreta.

4.5.2 Comparación entre arquitecturas

Los resultados muestran comportamientos diferenciados según el dataset.

Esclerosis múltiple

En este caso, *nnU-Net v2* y *U-Mamba* alcanzan rendimientos muy similares, con una ligera ventaja sistemática de *U-Mamba*. Sin embargo, la diferencia es reducida.

Esto sugiere que, para lesiones pequeñas y dispersas, el rendimiento está principalmente limitado por la resolución de la imagen y el contraste radiológico más que por la capacidad de modelado global de la arquitectura. El problema se comporta como una detección de patrones locales repetitivos, donde las convoluciones tradicionales ya resultan altamente eficaces.

Metástasis

En contraste, en el dataset de metástasis se observa una diferencia sustancial entre modelos, donde *U-Mamba* supera claramente a *nnU-Net v2*.

Las metástasis presentan estructuras compactas, con bordes definidos y dependencia contextual respecto al tejido circundante. En este escenario, la capacidad de los *State Space Models* para capturar relaciones espaciales de largo alcance permite mejorar la coherencia global de la segmentación y reducir fallos completos de detección.

Por tanto, mientras que en esclerosis múltiple la arquitectura no es el factor dominante, en metástasis sí lo es.

4.5.3 Sobresegmentación e infrasegmentación

El análisis de TP, FN y FP permite identificar tendencias sistemáticas:

- En esclerosis múltiple, ambos modelos tienden a **infrasegmentar**, omitiendo lesiones pequeñas o de bajo contraste.
- En metástasis, *nnU-Net v2* presenta con frecuencia fallos completos de detección, mientras que *U-Mamba* detecta la lesión pero en ocasiones introduce mayor **sobresegmentación**.

Esto sugiere que los errores en esclerosis son principalmente de sensibilidad, mientras que en metástasis están más relacionados con la capacidad de modelado global.

4.5.4 Coste computacional

El análisis de los tiempos de entrenamiento revela un comportamiento no trivial. Contrariamente a lo esperable, las configuraciones 3D no resultan más lentas que las 2D en *nnU-Net v2*. De hecho, el tiempo medio por época es aproximadamente un 20–25% menor en 3D.

Este efecto se debe a la estrategia de muestreo del propio framework. En la configuración 2D el modelo procesa cada corte de forma independiente, generando un mayor número de iteraciones por época. Por el contrario, en 3D se utilizan parches volumétricos que cubren mayor región anatómica por iteración, reduciendo el número total de batches necesarios para completar una época.

Por tanto, aunque el coste computacional por forward pass es mayor en 3D, el número total de iteraciones disminuye, resultando en épocas más cortas.

En cuanto a la comparación entre arquitecturas, *U-Mamba* presenta sistemáticamente tiempos de entrenamiento superiores a *nnU-Net v2*. En el dataset de esclerosis múltiple, el entrenamiento 2D requiere aproximadamente el doble de tiempo por época, mientras que en 3D el incremento se sitúa en torno a un 70–80%.

Este aumento se debe a la mayor complejidad de los bloques basados en *State Space Models*, que introducen operaciones adicionales de modelado contextual global.

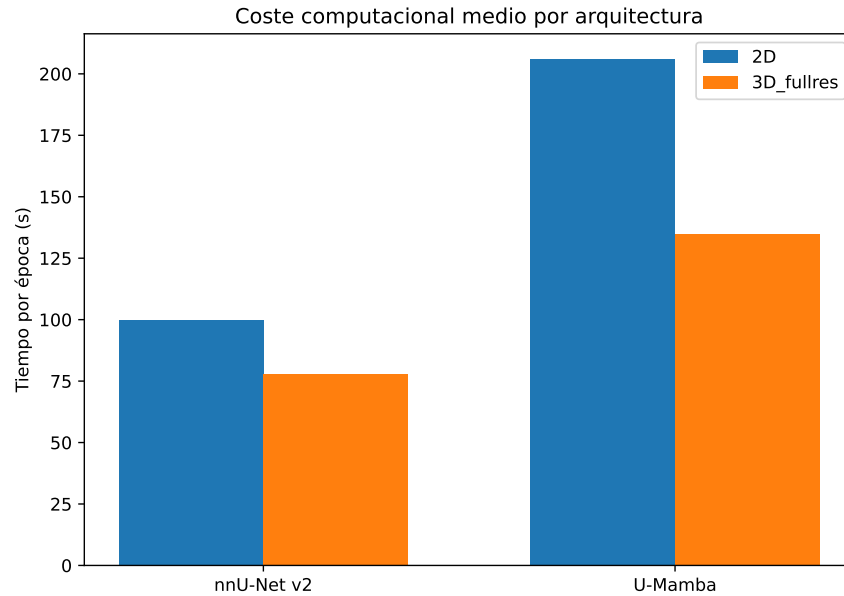


Figura 4.11. Tiempo medio de entrenamiento por época para cada arquitectura y dimensionalidad.

En consecuencia, se observa un compromiso claro entre precisión y eficiencia:

- **nnU-Net v2** ofrece el mejor equilibrio entre rendimiento y tiempo de entrenamiento.
- **U-Mamba** proporciona mayor precisión en escenarios complejos (metástasis) a costa de mayor coste computacional.

4.5.5 Interpretación global

El análisis conjunto de ambos datasets muestra que el rendimiento depende principalmente del tipo de lesión y no únicamente de la arquitectura empleada.

En esclerosis múltiple, caracterizada por lesiones pequeñas, numerosas y dispersas, ambos modelos alcanzan rendimientos muy similares. El problema se comporta esencialmente como una detección local de patrones de intensidad, donde la capacidad de modelado global aporta beneficios limitados.

En metástasis cerebrales ocurre lo contrario. Las lesiones presentan estructuras volumétricas compactas cuya correcta delimitación depende del contexto anatómico circundante. En este escenario, la capacidad de *U-Mamba* para modelar dependencias espaciales de largo alcance reduce fallos completos de detección y mejora significativamente el coeficiente Dice.

Por tanto, los resultados sugieren que:

- No existe una arquitectura universalmente superior.
- Las CNN tradicionales son suficientes para **lesiones difusas y locales**.
- Los modelos con contexto global resultan ventajosos en **lesiones estructuradas y volumétricas**.

Esto indica que futuras líneas de investigación podrían orientarse hacia modelos adaptativos capaces de modificar dinámicamente su campo receptivo en función de la morfología de la lesión.

En consecuencia, el rendimiento observado no depende únicamente de la arquitectura, sino de la relación entre el tamaño de la lesión y el campo receptivo efectivo del modelo. Cuando la patología puede describirse mediante patrones locales repetitivos, las CNN resultan suficientes; cuando requiere integración espacial de largo alcance, los modelos con contexto global presentan ventajas claras.

5

Conclusiones y líneas futuras

En este capítulo se resumen los principales hallazgos del trabajo y se discuten las posibles líneas de mejora y continuidad.

5.1 Conclusiones

El objetivo principal de este trabajo ha sido analizar el impacto del modelado de contexto global en segmentación de lesiones cerebrales mediante la comparación entre una arquitectura convolucional consolidada (*nnU-Net v2*) y una arquitectura basada en *State Space Models* (*U-Mamba*). Para ello se evaluaron ambos enfoques en dos problemas clínicos con características radiológicas distintas: esclerosis múltiple y metástasis cerebrales.

Los resultados obtenidos muestran que la arquitectura no es, por sí sola, el factor dominante del rendimiento. En su lugar, el comportamiento del modelo depende principalmente de la naturaleza espacial de la patología.

En esclerosis múltiple, caracterizada por lesiones pequeñas, numerosas y dispersas, ambos modelos alcanzan rendimientos prácticamente equivalentes. Esto indica que el problema se comporta como una detección de patrones locales de intensidad, donde las convoluciones tradicionales ya resultan suficientes para capturar la información relevante. En este escenario, el modelado de dependencias de largo alcance aporta únicamente mejoras marginales.

Por el contrario, en el dataset de metástasis cerebrales se observa una diferencia sustancial entre arquitecturas. Las lesiones presentan estructuras volumétricas compactas cuya identificación depende del contexto anatómico circundante. En este caso, *U-Mamba* reduce fallos completos de detección y mejora significativamente el coeficiente Dice, evidenciando la utilidad del modelado global.

Adicionalmente, se confirma que la dimensionalidad del modelo constituye el factor más influyente de todos los evaluados. Las configuraciones 3D superan sistemáticamente a las 2D en ambos datasets, demostrando que la continuidad espacial entre cortes es crítica en segmentación médica.

Por tanto, la principal aportación del trabajo no radica en identificar una arquitectura universalmente superior, sino en establecer un criterio de aplicabilidad: las CNN tradicionales resultan adecuadas para lesiones difusas dominadas por patrones

locales, mientras que los modelos con contexto global muestran ventajas claras en lesiones estructuradas cuya detectabilidad depende de relaciones espaciales amplias.

Finalmente, el análisis cualitativo indica que el principal límite del sistema no reside únicamente en la delimitación precisa de contornos, sino en la propia detectabilidad de ciertas lesiones en la imagen. En los casos más complejos, la señal radiológica de la lesión es similar al tejido sano circundante, situándose cerca del umbral de resolución efectiva. En estas situaciones el problema deja de ser puramente de segmentación y pasa a ser de identificación, introduciendo una limitación inherente al dato que no puede resolverse únicamente mediante cambios en la arquitectura del modelo.

5.2 Líneas futuras

Los resultados obtenidos sugieren que el rendimiento de los modelos de segmentación médica no depende únicamente de la arquitectura empleada, sino en gran medida de la calidad, representatividad y riqueza de la información contenida en los datos. A partir de ello se proponen las siguientes líneas de continuación:

Calidad y resolución de imagen

El análisis cualitativo ha mostrado que el principal límite del sistema aparece en lesiones cercanas al umbral de detectabilidad radiológica. Por ello, una línea directa de mejora consiste en el uso de datasets con mayor resolución espacial, menor ruido y protocolos de adquisición homogéneos. La disponibilidad de imágenes de mayor calidad permitiría evaluar el verdadero límite de los modelos, diferenciando errores algorítmicos de limitaciones inherentes al dato.

Mejoras en el preprocesamiento

Otra línea relevante es el estudio sistemático de estrategias de preprocesamiento orientadas a aumentar la separabilidad entre tejido sano y patológico. Entre ellas destacan normalizaciones de intensidad específicas por modalidad, corrección de bias field, denoising basado en aprendizaje profundo y técnicas de super-resolución. Este enfoque podría resultar especialmente útil en lesiones pequeñas, donde la detectabilidad condiciona más el rendimiento que la arquitectura del modelo.

Uso de múltiples modalidades

El presente trabajo emplea una única modalidad por dataset, sin embargo en la práctica clínica la interpretación radiológica es multimodal. La incorporación conjunta de secuencias T1, T2 y FLAIR permitiría aportar información complementaria sobre la estructura tisular, mejorando la discriminación entre lesión real y variabilidad anatómica normal.

Generalización a otras patologías

Los resultados muestran que el comportamiento de cada arquitectura depende fuertemente de la morfología de la lesión. Por ello, resulta de interés evaluar los modelos en otros escenarios clínicos (tumores infiltrativos, edema difuso, lesiones isquémicas, etc.) con el fin de caracterizar en qué tipos de patología resulta más beneficioso el modelado de contexto global frente a las aproximaciones convolucionales tradicionales.

Aprendizaje adaptativo al contexto

Los resultados sugieren que no existe una arquitectura universalmente óptima, sino que su eficacia depende de la estructura espacial de la lesión. Como continuación natural, sería interesante investigar modelos híbridos o adaptativos capaces de modificar dinámicamente su campo receptivo o su mecanismo de atención en función de la morfología detectada durante la inferencia.

Evaluación clínica y métricas avanzadas

Finalmente, una extensión relevante consiste en incorporar métricas clínicamente orientadas (detección por lesión, sensibilidad por paciente, volumen tumoral o métricas topológicas) que complementen al coeficiente Dice. Estas medidas permitirían evaluar el impacto real del sistema en un entorno asistencial más allá del solapamiento voxel a voxel.



Apéndice I: Manual de usuario

El presente apéndice describe los pasos necesarios para reproducir los experimentos realizados en este trabajo en un equipo nuevo, partiendo desde un sistema sin configurar.

A.1 Requisitos del sistema

Para realizar el mismo entorno que en este trabajo se recomienda un sistema Windows con WSL2 habilitado y GPU NVIDIA compatible con CUDA (Aunque es posible que en un solo sistema Ubuntu 20.04 se pueda ejecutar ambos algoritmos correctamente).

Se recomienda revisar los requisitos hardware especificados en la documentación de nnUNetv2 [16]

A.2 Creación de entornos virtuales

Para llevar a cabo este TFG, se utilizó una división mediante entornos virtuales, para poder satisfacer las diferentes dependencias del código y los algoritmos utilizados.

Instalación y Configuración del Entorno

Para garantizar la reproducibilidad de los experimentos, es fundamental configurar correctamente los entornos de ejecución. Mientras que *nnU-Net v2* puede ejecutarse de forma nativa en Windows, el modelo *U-Mamba* requiere de un entorno basado en Linux para la correcta compilación de sus núcleos específicos (*kernels*).

1. Entorno para *nnU-Net v2* (Windows Nativo)

La instalación oficial puede consultarse en el repositorio del framework [16]. En este trabajo, el procedimiento se resume en los siguientes puntos:

A. Controladores y Compatibilidad

Es imperativo instalar los controladores NVIDIA compatibles con la arquitectura de la GPU. Para una correcta comunicación con *PyTorch*, el driver debe ser igual o superior a la versión 560.94, asegurando soporte para CUDA 12.6. Una instalación incorrecta impedirá que el hardware sea detectado para aceleración por GPU.

B. Instalación del Framework

Se recomienda el uso de un entorno virtual de Python. Una vez activado, se clona el repositorio y se instala en modo editable (*editable mode*):

```
1 # Clonar repositorio oficial
2 git clone https://github.com/MIC-DKFZ/nnUNet.git
3 cd nnUNet
4
5 # Instalar en modo editable
6 pip install -e .
```

Nota: El modo editable permite realizar ajustes en el código fuente, como la modificación manual del número de épocas por fold, sin necesidad de reinstalar el paquete.

C. Configuración de Variables de Entorno

nnU-Net utiliza tres rutas globales obligatorias para organizar el flujo de datos. Estas deben configurarse en las variables de entorno del sistema operativo:

- **nnUNet_raw**: Ubicación de los datos originales en formato NIfTI.
- **nnUNet_preprocessed**: Carpeta donde se almacenarán los datos tras el tipado y normalizado.
- **nnUNet_results**: Ruta donde el framework guardará los pesos (*weights*) del modelo entrenado.

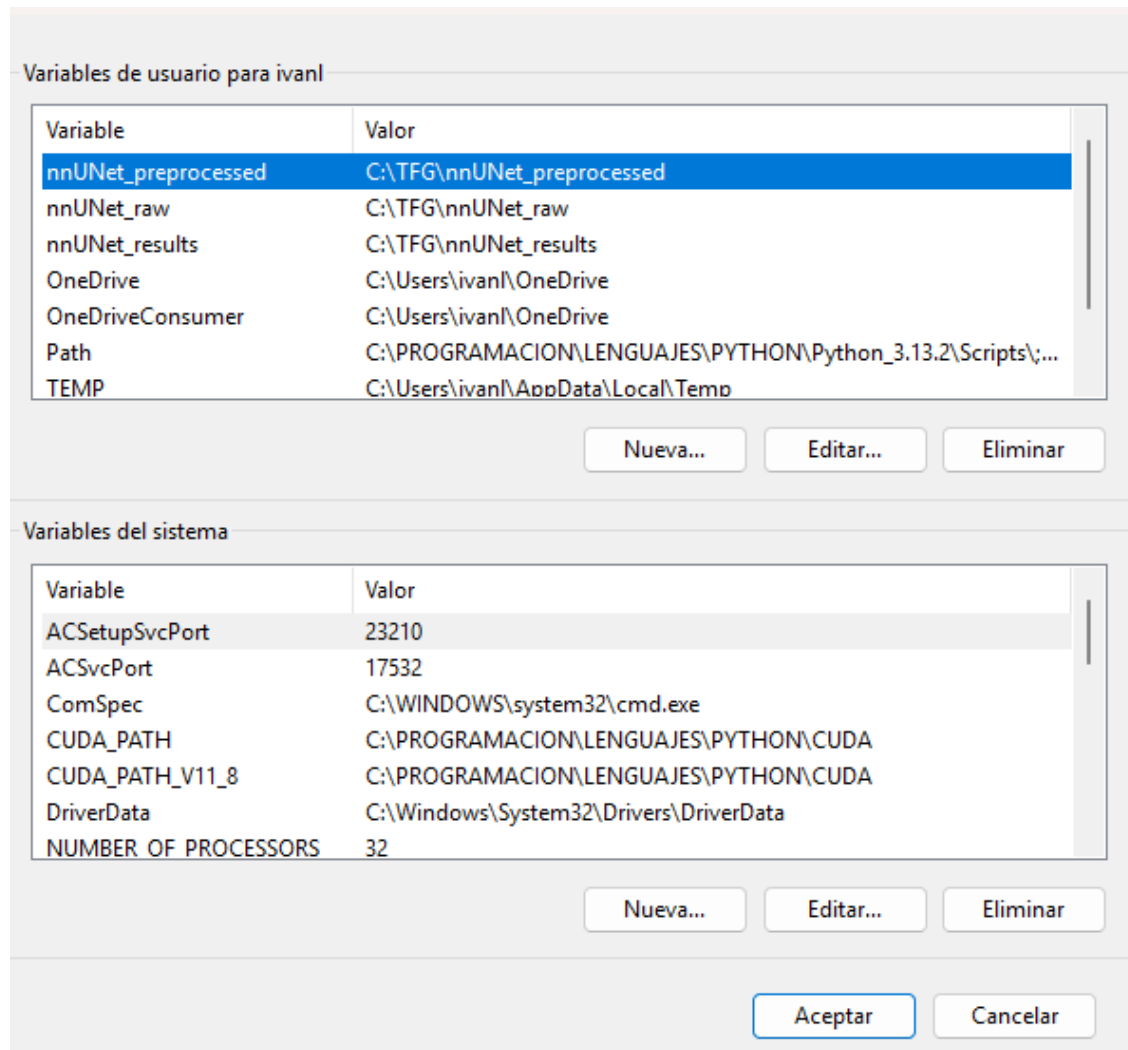


Figura A.1. Configuración de variables de entorno necesarias para nnU-Net v2 en Windows.

2. Entorno para U-Mamba (WSL2 - Ubuntu 20.04)

U-Mamba depende de capas de convolución causal y escaneo selectivo (*Selective Scan*) que requieren compilación de bajo nivel en C++ y CUDA. Por ello, se utiliza el Subsistema de Windows para Linux (WSL2).

A. Preparación de la Distribución

Se debe instalar una distribución limpia de Ubuntu 20.04 desde PowerShell para evitar conflictos con otras instalaciones de Linux:

```
1 wsl --install -d Ubuntu-20.04
```

B. Gestión de Entorno con Conda

Dentro de la terminal de Ubuntu, se instala *Miniconda* para gestionar las dependencias de forma aislada:

```
1 # Descarga e instalacion de Miniconda
2 wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
3 bash Miniconda3-latest-Linux-x86_64.sh # Seleccionar 'yes' para inicializar conda
4
5 # Creacion del entorno especifico
6 conda create -n entorno_umamba python=3.10 -y
7 conda activate entorno_umamba
```

C. Instalación de Dependencias Críticas

Para replicar el entorno funcional de este trabajo, se instala *PyTorch* con soporte para CUDA 11.8. La compilación de los módulos de Mamba debe realizarse desactivando el aislamiento de construcción (*build isolation*) para asegurar que detecten la instalación de *PyTorch* activa:

```
1 # Instalacion de PyTorch y herramientas de compilacion
2 pip install torch==2.0.1 torchvision==0.15.2 --index-url https://download.pytorch.org/whl/cu118
3 pip install ninja wheel
4
5 # Compilacion de modulos especificos (causal-conv1d y mamba-ssm)
6 pip install causal-conv1d==1.2.0.post2 --no-build-isolation
7 pip install mamba-ssm==1.2.0 --no-build-isolation
```

D. Instalación de la Arquitectura

Finalmente, se descarga el código de *U-Mamba* y se instala como extensión de *nnU-Net*:

```
1 git clone https://github.com/bowang-lab/U-Mamba
2 cd U-Mamba/umamba
3 pip install -e .
```

A.3 Descarga y Organización de los Datasets

Una vez configurados los entornos de ejecución, se procede a la obtención de los datos médicos. Los conjuntos de datos públicos utilizados en este trabajo deben descargarse desde sus fuentes oficiales:

- **Esclerosis Múltiple:** Acceso según las directrices del apartado 3.1.2.
- **Metástasis Cerebrales:** Acceso según las directrices del apartado 3.1.3.

Tras la descarga, es necesario normalizar la estructura de archivos. Para ello, se han desarrollado scripts de automatización en Python (adjuntos en la memoria) que realizan el renombrado y la jerarquización requerida por el framework:

- DATASET_ESCLEROSIS.py
- DATASET_METASTASIS.py

A.4 Pipeline de Preprocesamiento

El preprocesamiento es una etapa crítica para homogeneizar las imágenes de diferentes centros médicos. Este proceso se gestiona a través del entorno virtual `entorno_preprocesar` mediante el notebook `Preprocessing.ipynb`.

Configuración de Plantillas (Templates)

Dependiendo del objetivo del experimento, se debe seleccionar una plantilla de registro espacial. En el código, esto se define mediante la variable `template` (se adjuntan en el código):

```
1 # Opción 1: Plantilla MRI225 (Recomendada)
2 template = r'C:\TFG\PC_NUEVO_PRUEBA\Dataset\Templates\MRI225\NMRI225_Flair.nii'
3
4 # Opción 2: Plantilla MNI152 (Estándar)
5 # template = #template = r'C:\TFG\PC_NUEVO_PRUEBA\Dataset\MNI152\mni_icbm152_t1_tal_nlin_sym_09a.
6     nii'
7
8 # Opción 3: Plantilla SRI24
9 #template = r'C:\TFG\PC_NUEVO_PRUEBA\Dataset\SRI24\T1.nii'
```

Listing A.1. Selección de plantillas de registro

Ejecución según el Dataset

Asimismo, se debe descomentar la función correspondiente al volumen de datos a tratar:

```
1 # Para el dataset de Esclerosis (Secuencia Flair + T1)
2 preprocess_dataset(input_dir, base_dir, template, participants_file, id_col, col, col_value, case,
3     idx=idx, sequence="flair+t1")
4
5 # Para el dataset de Metástasis (Secuencia T1)
6 # preprocess_dataset(input_dir, base_dir, template, participants_file, id_col, col, col_value, case,
7     idx=idx, sequence="t1")
```

Los archivos resultantes se almacenarán automáticamente en la ruta de entrada para el entrenamiento, por defecto (modificar): `C:\TFG\PC_NUEVO_PRUEBA\Resultados_preprocesamiento\nnUNet\input`.

A.5 Diseño Experimental y Entrenamiento

Una vez preprocesados los datos, la carpeta resultante debe trasladarse al directorio `nnUNet_raw` para iniciar el flujo de trabajo de los modelos.

A.6 Diseño Experimental y Ejecución

Una vez preprocesados los datos, la carpeta resultante debe trasladarse al directorio `nnUNet_raw` para iniciar el flujo de trabajo. A continuación, se detallan los comandos necesarios para completar el ciclo de vida del modelo, desde el entrenamiento hasta la evaluación cuantitativa.

1. Ejecución en *nnU-Net v2* (Entorno: `entorno_nnUNet`)

El proceso sigue una secuencia lógica de planificación, entrenamiento multifold e inferencia con postprocesamiento optimizado.

A. Planificación y Entrenamiento

Se extraen las características del dataset y se lanzan los 5 *fold*s de la validación cruzada:

```
1 # Planificacion (usando el Planner optimizado para ResEnc)
2 nnUNetv2_plan_and_preprocess -d 450 -pl nnUNetPlannerResEncM --verify_dataset_integrity
3
4 # Entrenamiento de los 5 folds
5 nnUNetv2_train 450 2d 0 -p nnUNetResEncUNetMPlans --npz
6 nnUNetv2_train 450 2d 1 -p nnUNetResEncUNetMPlans --npz
7 nnUNetv2_train 450 2d 2 -p nnUNetResEncUNetMPlans --npz
8 nnUNetv2_train 450 2d 3 -p nnUNetResEncUNetMPlans --npz
9 nnUNetv2_train 450 2d 4 -p nnUNetResEncUNetMPlans --npz
```

B. Inferencia y Evaluación de Resultados

Tras el entrenamiento, se ejecutan los comandos para generar las predicciones sobre el conjunto de test y aplicar las correcciones morfológicas:

```
1 # 1. Identificar la mejor configuracion
2 nnUNetv2_find_best_configuration 450 -c 2d -p nnUNetResEncUNetMPlans
3
4 # 2. Inferencia (Predict) sobre imagenes de test
5 nnUNetv2_predict -d Dataset450_Esclerosis_MRI225_BUENA -i C:\TFG\nnUNet_raw\
   Dataset450_Esclerosis_MRI225_BUENA\imagesTs -o C:\TFG\nnUNet_inference\450_predict -f 0 1 2 3 4
   -tr nnUNetTrainer -c 2d -p nnUNetResEncUNetMPlans --save_probabilities
6
7 # 3. Aplicar Postprocesamiento (basado en el archivo .pkl generado en el entrenamiento)
```

```

8 nnUNetv2_apply_postprocessing -i C:\TFG\nnUNet_inference\450_predict -o C:\TFG\nnUNet_inference\450
  _postprocessing -pp_pkl_file C:\TFG\nnUNet_results\Dataset450_Esclerosis_MRI225_BUENA\
  nnUNetTrainer__nnUNetResEncUNetMPlans__2d\crossval_results_folds_0_1_2_3_4\postprocessing.pkl -
  np 8 -plans_json C:\TFG\nnUNet_results\Dataset450_Esclerosis_MRI225_BUENA\
  nnUNetTrainer__nnUNetResEncUNetMPlans__2d\crossval_results_folds_0_1_2_3_4\plans.json
9
10 # 4. Evaluacion final del test set (Generacion de metricas Dice, Hausdorff, etc.)
11 nnUNetv2_evaluate_folder C:\TFG\nnUNet_raw\Dataset450_Esclerosis_MRI225_BUENA\labelsTs C:\TFG\
  nnUNet_inference\450_postprocessing -djfile C:\TFG\nnUNet_raw\
  Dataset450_Esclerosis_MRI225_BUENA\dataset.json -pfile C:\TFG\nnUNet_results\
  Dataset450_Esclerosis_MRI225_BUENA\nnUNetTrainer__nnUNetResEncUNetMPlans__2d\
  crossval_results_folds_0_1_2_3_4\plans.json -o C:\TFG\nnUNet_inference\450_postprocessing\
  evaluacion_450_2d.json

```

2. Ejecución en *U-Mamba* (Entorno: WSL2 - entorno_umamba)

Para *U-Mamba*, el procedimiento es análogo, pero adaptado a la jerarquía de directorios del Subsistema de Linux y utilizando el *trainer* específico que integra los bloques de *Mamba*.

A. Entrenamiento de la Arquitectura *Mamba*

```

1 # Planificacion y entrenamiento multifold
2 nnUNetv2_plan_and_preprocess -d 450 --verify_dataset_integrity
3
4 nnUNetv2_train 450 2d 0 -tr nnUNetTrainerUMambaBot
5 nnUNetv2_train 450 2d 1 -tr nnUNetTrainerUMambaBot
6 nnUNetv2_train 450 2d 2 -tr nnUNetTrainerUMambaBot
7 nnUNetv2_train 450 2d 3 -tr nnUNetTrainerUMambaBot
8 nnUNetv2_train 450 2d 4 -tr nnUNetTrainerUMambaBot

```

B. Pipeline de Inferencia y Post-análisis

Debido a la integración de *U-Mamba* sobre la base de *nnU-Net*, los comandos de evaluación son compatibles pero requieren las rutas montadas en `/mnt/c/`:

```

1 # 1. Inferencia (Predict)
2 nnUNetv2_predict -d Dataset450_Esclerosis_MRI225_BUENA -i /mnt/c/TFG/U-Mamba/data/nnUNet_raw/
  Dataset450_Esclerosis_MRI225_BUENA/imagesTs -o /mnt/c/TFG/U-Mamba/data/nnUNet_inference/450
  _predict -f 0 1 2 3 4 -tr nnUNetTrainerUMambaBot -c 2d -p nnUNetPlans --save_probabilities
3

```

```

4 # 2. Postprocesamiento de segmentaciones
5 nnUNetv2_apply_postprocessing -i /mnt/c/TFG/U-Mamba/data/nnUNet_inference/450_predict -o /mnt/c/TFG/
  U-Mamba/data/nnUNet_inference/450_postprocessing -pp_pkl_file /mnt/c/TFG/U-Mamba/data/
  nnUNet_results/Dataset450_Esclerosis_MRI225_BUENA/nnUNetTrainerUMambaBot__nnUNetPlans__2d/
  crossval_results_folds_0_1_2_3_4/postprocessing.pkl -np 8 -plans_json /mnt/c/TFG/U-Mamba/data/
  nnUNet_results/Dataset450_Esclerosis_MRI225_BUENA/nnUNetTrainerUMambaBot__nnUNetPlans__2d/
  crossval_results_folds_0_1_2_3_4/plans.json
6
7 # 3. Generacion de archivo de metricas evaluacion_umamba.json
8 nnUNetv2_evaluate_folder /mnt/c/TFG/U-Mamba/data/nnUNet_raw/Dataset450_Esclerosis_MRI225_BUENA/
  labelsTs /mnt/c/TFG/U-Mamba/data/nnUNet_inference/450_postprocessing -djfile /mnt/c/TFG/U-Mamba
  /data/nnUNet_raw/Dataset450_Esclerosis_MRI225_BUENA/dataset.json -pfile /mnt/c/TFG/U-Mamba/data
  /nnUNet_results/Dataset450_Esclerosis_MRI225_BUENA/nnUNetTrainerUMambaBot__nnUNetPlans__2d/
  crossval_results_folds_0_1_2_3_4/plans.json -o /mnt/c/TFG/U-Mamba/data/nnUNet_inference/450
  _postprocessing/evaluacion_450_2d.json

```

Nota: Como se describió en la estructura del algoritmo (Sección 3.1.4), los resultados cuantitativos se localizan en los archivos JSON generados al final de cada pipeline, permitiendo la comparación directa de métricas entre el modelo base y el modelo basado en Mamba.

A.7 Visualización y Análisis de Resultados

Para la validación cualitativa de las segmentaciones, se ha empleado la herramienta **BioImage Suite Editor** [20]. El procedimiento consiste en cargar la imagen original (MRI) y superponer la máscara predicha mediante la función *Objectmap*, lo que permite inspeccionar visualmente la precisión de la detección frente al *ground truth*.

Bibliografía

- [1] AFNI (NIMH). *NIfTI-1 data format documentation (AFNI)*. <https://afni.nimh.nih.gov/pub/dist/doc/html/doc/nifti/format.html>. Accessed: 2026-01-17. 2026.
- [2] Zeynettin Akkus et al. "Deep learning for brain MRI segmentation: state of the art and future directions". In: *Journal of Digital Imaging* 30 (2017), pp. 449–459. DOI: 10.1007/s10278-017-9983-4. URL: <https://link.springer.com/article/10.1007/s10278-017-9983-4>.
- [3] Michela Antonelli et al. "The Medical Segmentation Decathlon". In: *Nature Communications* 13 (2022), p. 4128. DOI: 10.1038/s41467-022-30695-9.
- [4] Özgün Çiçek et al. "3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation". In: *MICCAI* (2016). URL: <https://arxiv.org/abs/1606.06650>.
- [5] Daniel García-Lorenzo et al. "A review of automatic segmentation methods for multiple sclerosis lesions". In: *NeuroImage: Clinical* (2018). URL: <https://www.sciencedirect.com/science/article/pii/S2213158218301880>.
- [6] *Grand Challenges in Biomedical Image Analysis*. <https://atlas.grand-challenge.org/>. Accedido en 2025. 2024.
- [7] Albert Gu and Tri Dao. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces". In: *arXiv preprint arXiv:2312.00752* (2023). URL: <https://arxiv.org/abs/2312.00752>.
- [8] Francesco Guarnera et al. *MSLesSeg: Multiple Sclerosis Lesion Segmentation Dataset*. Accedido en noviembre de 2025. 2024. URL: https://springernature.figshare.com/articles/dataset/MSLesSeg_baseline_and_benchmarking_of_a_new_Multiple_Sclerosis_Lesion_Segmentation_dataset/27919209.
- [9] MICCAI Educational Initiative. *nnU-Net: The no-new-Net for automatic segmentation*. <https://medium.com/miccai-educational-initiative/nnu-net-the-no-new-unet-for-automatic-segmentation-8d655f3f6d2a>. Accedido en 2025. 2020.
- [10] Fabian Isensee et al. "nnU-Net: a self-configuring method for deep learning-based biomedical image segmentation". In: *Nature Methods* 18.2 (2021), pp. 203–211. DOI: 10.1038/s41592-020-01008-z. URL: <https://www.nature.com/articles/s41592-020-01008-z>.
- [11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539. URL: <https://www.nature.com/articles/nature14539>.
- [12] Fei-Fei Li, Andrej Karpathy, and Justin Johnson. *CS231n: Convolutional Neural Networks for Visual Recognition*. <https://cs231n.github.io/convolutional-networks/>. Curso de la Universidad de Stanford. Accedido en 2025. 2017.

- [13] Geert Litjens, Thijs Kooi, et al. "A survey on deep learning in medical image analysis". In: *Medical Image Analysis* 42 (2017), pp. 60–88. DOI: 10.1016/j.media.2017.07.005. URL: <https://www.sciencedirect.com/science/article/pii/S1361841517301135>.
- [14] Jun Ma, Feifei Li, and Bo Wang. "U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation". In: *arXiv preprint arXiv:2401.04722* (2024). URL: <https://arxiv.org/abs/2401.04722>.
- [15] *MetaS-RD: Brain metastasis MRI dataset*. MoLab datasets portal. Accessed: 2026-01-17. URL: <https://molab.es/datasets-brain-metastasis-1/?type=metasrd>.
- [16] MIC-DKFZ. *nnU-Net (GitHub repository)*. <https://github.com/MIC-DKFZ/nnUNet>. Accessed: 2026-01-17. 2026.
- [17] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation". In: *3D Vision (3DV)* (2016). URL: <https://arxiv.org/abs/1606.04797>.
- [18] Neuroimaging Informatics Technology Initiative. *NIfTI: Neuroimaging Informatics Technology Initiative (NITRC project page)*. <https://www.nitrc.org/projects/nifti/>. Accessed: 2026-01-17. 2026.
- [19] Beatriz Ocaña-Tienda et al. "A comprehensive dataset of annotated brain metastasis MR images with clinical and radiomic data". In: *Scientific Data* 10 (2023), p. 208. DOI: 10.1038/s41597-023-02123-0. URL: <https://www.nature.com/articles/s41597-023-02123-0>.
- [20] Xenophon Papademetris and Dustin Scheinost. *BioImage Suite Web: Medical Image Viewer*. <https://bioimagesuiteweb.github.io/webapp/editor.html>. Accedido en noviembre de 2025. 2023.
- [21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* 9351 (2015), pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28. URL: <https://arxiv.org/abs/1505.04597>.
- [22] tiffany9056. *U²-Mamba / UU-Mamba (GitHub repository)*. <https://github.com/tiffany9056/uu-mamba>. Accessed: 2026-01-17. 2026.



UNIVERSIDAD
DE MÁLAGA

| uma.es

ETS. de Ingeniería Informática
Bulevar Louis Pasteur, 35
Campus de Teatinos
29071 Málaga