

UNIVERSIDAD POLITÉCNICA DE MADRID  
Escuela Técnica Superior de Ingeniería de Sistemas Informáticos

**Grado en Ingeniería de Computadores**

**TJ-II Assistant for MHD Detection and Shot Data**

# **PROYECTO FIN DE GRADO**

Jaime Rodríguez Lara  
Grado en Ingeniería de Computadores

Bajo la dirección de:  
Dr. Joaquin Gayoso-Cabada

Madrid, 2025

**Título:** TJ-II Assistant for MHD Detection and Shot Data

**Autor:** Jaime Rodríguez Lara  
Grado en Ingeniería de Computadores

**Dirección:**  
Dr. Joaquin Gayoso-Cabada, UPM

## Agradecimientos

Quiero expresar mi más sincero agradecimiento a **Augusto Pereira González**, del *Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT)*, por su constante apoyo y colaboración a lo largo del desarrollo de este proyecto. Ha sido la persona encargada de proporcionarnos toda la información necesaria sobre los datos generados por el **TJ-II** [1], incluyendo archivos `.csv`, enlaces y documentación técnica relevante.

## Resumen

El objetivo principal del proyecto es diseñar y desarrollar un asistente conversacional centrado en la interpretación de datos generados por el TJ-II [1], un dispositivo de confinamiento magnético tipo *stellarator* operado por el CIEMAT [2]. Este experimento produce un gran volumen de información en cada disparo de plasma, cuya interpretación es esencial para el análisis de eventos relevantes como la aparición de inestabilidades Magnetohidrodinámica (MHD) [3].

El proyecto se ha dividido en dos etapas principales. La primera se ha tratado de la obtención y clasificación de espectrogramas previamente procesados. Se han obtenido los espectrogramas de unas determinadas descargas en las cuales se observaba con claridad la existencia, o no, de MHD, y se han clasificado manualmente con "Y"(sí) o "N"(no), con el objetivo de un mejor entrenamiento del modelo en la segunda etapa, más enfocada al entrenamiento automático e interpretación de los datos.

Como se ha mencionado con anterioridad, en la segunda etapa se ha procedido al entrenamiento de un modelo, entrenado con los datos y espectrogramas guardados, para su posterior incorporación al asistente.

Junto a esto se han añadido dos funcionalidades más, no relacionadas con MHD. La primera se centra en la obtención de diagramas a través de la conexión a una web del CIEMAT, y la segunda en la interpretación de los datos de las descargas, guardadas en un fichero Comma-Separated Values (CSV).

Los resultados obtenidos muestran que el asistente es capaz de clasificar correctamente espectrogramas, interpretar datos e imprimir diagramas de disparos concretos, aportando un valor añadido al proceso de análisis y reduciendo la carga manual del equipo de investigación. Esto abre la puerta a futuras aplicaciones de Inteligencia Artificial (IA) en entornos de investigación científica complejos.

**Palabras clave:** Chatbot, TJ-II, Espectrogramas, MHD.

## **Abstract**

The main objective of the project is to design and develop a conversational assistant focused on interpreting data generated by the TJ-II, a stellarator-type magnetic confinement device operated by CIEMAT. This experiment produces a large volume of information with each plasma shot, the interpretation of which is essential for analyzing relevant events such as the occurrence of MHD instabilities.

The project was divided into two main stages. The first involved obtaining and classifying previously processed spectrograms. Spectrograms were obtained from specific discharges in which the presence or absence of MHD was clearly observed, and they were manually classified as "Y"(yes) or "N"(no). This was intended to improve model training in the second stage, which was more focused on automatic training and data interpretation.

As mentioned above, the second stage involved training a model, using the saved data and spectrograms, for subsequent incorporation into the assistant.

In addition, two additional features unrelated to MHD have been added. The first focuses on obtaining diagrams through a connection to a CIEMAT website, and the second on interpreting download data, saved in a CSV file.

The results obtained show that the assistant is capable of correctly classifying spectrograms, interpreting data, and printing diagrams of specific shots, adding value to the analysis process and reducing the manual burden on the research team. This opens the door to future applications of artificial intelligence in complex scientific research environments.

**Keywords:** Chatbot, TJ-II, spectrograms, MHD.

# Índice

<b>1. Introducción</b>	<b>7</b>
1.1. Objetivo . . . . .	8
1.2. Motivación . . . . .	9
1.3. Estructura de la memoria . . . . .	10
<b>2. Estado de la cuestión</b>	<b>11</b>
2.1. Introducción al Análisis de Datos en Fusión Nuclear . . . . .	11
2.2. Procesamiento tradicional de señales en fusión . . . . .	11
2.3. Avances Recientes en Modelos de Lenguaje Natural (LLMs) . . . . .	12
2.4. Aplicaciones de IA en Fusión Nuclear . . . . .	13
2.5. Visualización de Señales . . . . .	13
2.6. Predicción de Inestabilidades MHD con Espectrogramas . . . . .	14
2.7. Arquitectura de la Solución Propuesta . . . . .	16
2.8. Herramientas de automatización y análisis . . . . .	18
2.9. Conclusión del Estado de la Cuestión . . . . .	18
<b>3. Metodología</b>	<b>19</b>
3.1. Visión general del sistema . . . . .	19
3.2. Procesamiento del lenguaje natural . . . . .	19
3.2.1. Clasificación de la intención del usuario . . . . .	20
3.2.2. Extracción de parámetros estructurados . . . . .	20
3.2.3. Consultas SQL generadas por IA . . . . .	21
3.3. Consulta de datos experimentales . . . . .	22
3.3.1. Generación de URL con parámetros . . . . .	22
3.4. Obtención y extracción de datos desde HTML . . . . .	22
3.4.1. Agrupación y visualización de señales . . . . .	23
3.5. Generación y análisis de espectrogramas . . . . .	24
3.5.1. Construcción de espectrogramas a partir de señales crudas . . . . .	24
3.5.2. Transformación espectral y generación de imágenes . . . . .	25
3.5.3. Extracción de características y predicción MHD . . . . .	25
3.6. Interfaz conversacional y flujo de ejecución . . . . .	26
3.6.1. Interacción con el usuario mediante Gradio . . . . .	26
3.6.2. Orquestación de módulos y lógica central . . . . .	27
3.7. Ficheros de configuración y estructura del proyecto . . . . .	28
3.7.1. Ficheros de configuración y estructura del sistema . . . . .	28
<b>4. Análisis de resultados</b>	<b>29</b>
4.1. Resultados analíticos . . . . .	29
4.2. Resultados visuales . . . . .	30
4.2.1. Visualización de espectrogramas con clasificación de inestabilidad MHD . . . . .	30
4.2.2. Visualización de espectrogramas sin clasificación de inestabilidad MHD . . . . .	30
4.2.3. Visualización de señales . . . . .	31
<b>5. Conclusiones y trabajo futuro</b>	<b>33</b>
5.1. Conclusiones generales . . . . .	33
5.2. Líneas futuras . . . . .	33
5.3. Impacto social y medioambiental . . . . .	34
5.3.1. Impacto social . . . . .	34
5.3.2. Impacto medioambiental . . . . .	35

## Índice de figuras

1.	Imagen del TJ-II. Fuente: CIEMAT[2]	11
2.	Ejemplo de visualización de señales. Fuente: TJ-II Data Display[4]	14
3.	Comparación de espectrogramas.	15
4.	Arquitectura general del sistema propuesto.	16
5.	Clasificación e Inteción de la consulta del usuario	20
6.	Agrupación e impresión de señales	23
7.	Construcción de espectrogramas	24
8.	Iterfaz de Gradio	26
9.	Sistema de ficheros	28
10.	Espectrograma 29867 clasificado con inestabilidad MHD	30
11.	Espectrograma 29569 clasificado sin inestabilidad MHD	31
12.	Diagrama TFI y ABOL1 de la señal 57543	31
13.	Diagrama TFI, ABOL1 y ABOL2 de la señal 57543	32
14.	Diagrama ABOL1 y ABOL2 de la señal 57543	32

## Índice de cuadros

1.	Comparativa de algunos modelos LLM recientes.	13
2.	Distribución del conjunto de evaluación	29
3.	Matriz de confusión del clasificador	29
4.	Métricas por clase	29

## 1. Introducción

Actualmente, las tecnologías están enfocadas a facilitar la vida de las personas, reduciendo en gran medida el tiempo que se consume en una tarea determinada. Los asistentes conversacionales entran dentro de este ámbito, facilitando resolver, de forma mucho más rápida, problemas o preguntas a las cuales a un ser humano le supondría un tiempo mucho mayor.

En este contexto, el trabajo se centra en el diseño y desarrollo de un asistente capaz de interpretar tanto datos, como imágenes. Este asistente permite al usuario interactuar en lenguaje natural para obtener los análisis mediante textos e imágenes de la información, sin gran necesidad de conocimientos técnicos mediante el uso de modelos de lenguaje grandes (LLMs).

Los modelos de lenguaje grandes (Large Language Models, LLMs) son sistemas de IA entrenados con grandes cantidades de datos textuales, permitiendo comprender, generar y responder en lenguaje natural. Estos modelos utilizan arquitecturas avanzadas de redes neuronales, como los transformadores, para capturar patrones complejos y relaciones contextuales dentro de los datos. Gracias a su capacidad para entender y generar texto de forma coherente, los LLMs han demostrado ser herramientas muy eficaces en tareas de asistencia conversacional, redacción automática, traducción y análisis de datos, entre otras aplicaciones.

Esto ha ayudado a que el asistente pueda identificar patrones en los espectrogramas, responder preguntas relacionados con los datos y representar diagramas de las señales pertenecientes a los experimentos. Esto proporciona una solución de gran accesibilidad para las tareas de investigación para los trabajadores del CIEMAT [2].

Una de las principales áreas de aplicación de este asistente ha sido en la investigación en energía de fusión, concretamente en los experimentos llevados a cabo en el TJ-II, un dispositivo de fusión nuclear de tipo stellarator ubicado en el CIEMAT [1]. Este dispositivo tiene el objetivo de estudiar el comportamiento del plasma confinado magnéticamente. El plasma es considerado el cuarto estado de la materia, ya que se trata de un gas ionizado, es decir, que sus partículas están cargadas y presentan un comportamiento colectivo, respondiendo de manera contundente a campos eléctricos y magnéticos.

La energía de fusión busca replicar en la Tierra las reacciones que se producen en el interior de las estrellas, como el Sol. Para ello, es necesario calentar gases hasta alcanzar un estado denomina-

do plasma. En el plasma, los átomos se ionizan, es decir, los electrones se separan de los núcleos atómicos, permitiendo que el gas conduzca electricidad y responda a campos magnéticos [3].

El reactor TJ-II está diseñado para confinar estos plasmas mediante campos magnéticos en configuraciones helicoidales complejas, evitando que el plasma entre en contacto con las paredes del contenedor, lo cual permite estudiar y optimizar las condiciones necesarias para una futura generación de energía limpia y prácticamente inagotable [4].

La capacidad del asistente desarrollado para interactuar con los datos generados por este tipo de experimentos, interpretarlos y proporcionar análisis de apoyo, representa una innovación significativa que contribuye a mejorar la eficiencia del trabajo investigador en un campo tan exigente como el de la fusión nuclear.

Para el análisis de los experimentos se ha usado lo que en el ámbito de la física se conoce como espectrograma, se trata de una representación visual que muestra la variación de frecuencia de una señal en función del tiempo. El uso de estas es especialmente en señales cuyo contenido de frecuencia no varía a lo largo del tiempo.

## 1.1. Objetivo

El objetivo principal de este proyecto ha sido diseñar un asistente inteligente capaz de interpretar y analizar datos experimentales del *stellarator* TJ-II [1], con especial énfasis en el procesamiento de espectrogramas generados a partir de señales experimentales. Este asistente proporciona una interfaz conversacional que permite a los investigadores interactuar con los datos de forma natural, facilitando el análisis de descargas del reactor y la detección de inestabilidades MHD.

Dentro de este objetivo general, se han definido los siguientes objetivos específicos:

- Interpretar espectrogramas generados a partir de señales experimentales, con el fin de detectar automáticamente la presencia o ausencia de inestabilidades MHD mediante modelos de IA.
- Analizar los datos recogidos en una base de datos en formato CSV, donde se almacenan las señales y metadatos correspondientes a cada experimento realizado en el TJ-II [1].
- Representar gráficamente las señales experimentales para facilitar su inspección y comparación, generando diagramas claros y contextualizados.

Este desarrollo contribuye a agilizar la exploración de datos y apoyar al personal científico del CIEMAT en la extracción de información útil a partir de grandes volúmenes de datos experimentales.

## **1.2. Motivación**

El análisis de los datos generados en el *stellarator* TJ-II [1], operado por el CIEMAT, resulta crucial para la comprensión del comportamiento que tiene el plasma y detección de fenómenos relevantes como las inestabilidades MHD [3]. Este análisis habitualmente requiere de una inspección manual de grandes cantidades de datos, lo que supone un proceso muy demandante de tiempo, lo que puede derivar en errores humanos y muy limitado a la capacidad de los investigadores.

Por ello, la motivación de este trabajo es automatizar, facilitar y agilizar el acceso a estos datos mediante el desarrollo de un asistente conversacional basado en IA que de la oportunidad a los investigadores del CIEMAT de interactuar con los datos del TJ-II [1] mediante lenguaje natural. Esto proporciona una interpretación más intuitiva y sobre todo eficiente de consultar la información en bases de datos, visualizar señales relevantes, generar espectrogramas y detectar automáticamente la presencia o ausencia de inestabilidades MHD.

Además, la implementación de modelos de lenguaje grandes (LLMs) y modelos de aprendizaje automático para el análisis de espectrogramas se alinea con las tendencias actuales de transformación digital en el ámbito científico. Esta herramienta no solo mejora la eficiencia en la interpretación de resultados, sino que también constituye un ejemplo práctico del impacto positivo que la IA puede tener en el avance de la investigación en física de plasmas.

### **1.3. Estructura de la memoria**

El documento se encuentra estructurado en varios capítulos que permiten abordar de forma ordenada y comprensiva los diferentes aspectos del proyecto.

En primer lugar, el capítulo 2 se dedica a realizar un análisis detallado del estado de la cuestión. En este apartado se presentan los modelos actuales empleados para el análisis y procesamiento de datos, así como una explicación teórica exhaustiva sobre las Magnetohidrodinámicas (MHD), que constituyen una base fundamental para la comprensión y el desarrollo del trabajo.

El capítulo 3 aborda el proceso de preprocesamiento de los datos. Se explica en detalle cómo se han preparado los datos para su posterior análisis, incluyendo las técnicas y procedimientos aplicados.

En el capítulo 4 se presentan los resultados obtenidos. Se analizan los logros alcanzados durante el desarrollo del proyecto. Este análisis crítico permite valorar la efectividad de las soluciones adoptadas.

Finalmente, el capítulo 5 está dedicado a las conclusiones. En él se reflexiona sobre los hallazgos más relevantes del estudio, se evalúa su impacto tanto social como medioambiental, y se proponen líneas futuras de investigación y desarrollo que podrían dar continuidad y enriquecer el trabajo realizado.

## 2. Estado de la cuestión

### 2.1. Introducción al Análisis de Datos en Fusión Nuclear

El estudio del plasma en dispositivos de confinamiento magnético como el TJ-II requiere un análisis profundo de las señales generadas durante las descargas experimentales. Estos datos permiten a los investigadores observar el comportamiento del plasma, identificar inestabilidades, y validar modelos teóricos.

El TJ-II, operado por el CIEMAT, es un stellarator flexible que produce datos de alta complejidad, incluyendo señales provenientes de distintos sensores y diagnósticos. Entre los fenómenos más relevantes se encuentran las inestabilidades magnetohidrodinámicas (MHD), que afectan la estabilidad del plasma. Detectarlas y analizarlas requiere una combinación de visualización de datos y técnicas de aprendizaje automático.

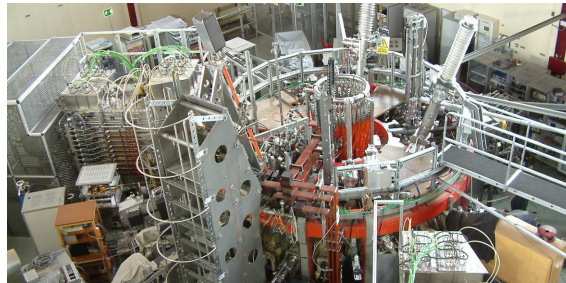


Figura 1: Imagen del TJ-II. Fuente: CIEMAT[2]

### 2.2. Procesamiento tradicional de señales en fusión

Históricamente, el análisis de los experimentos realizados en el dispositivo TJ-II se ha llevado a cabo mediante un enfoque mayoritariamente manual, basado en el uso de herramientas como IDL o MATLAB, y en el acceso directo a los ficheros de datos almacenados en servidores del CIEMAT. Cada experimento, también llamado descarga, genera un conjunto de señales provenientes de diferentes sensores, que deben ser extraídas, procesadas e interpretadas por el investigador.

Este análisis tradicional implica varias etapas encadenadas. En primer lugar, los investigadores deben descargar manualmente los ficheros desde la base de datos del experimento. A continuación, proceden a cargar las señales de interés una a una mediante scripts o funciones personaliza-

das que suelen estar adaptadas al tipo de experimento o diagnóstico. Posteriormente, las señales se visualizan en forma de gráficos temporales, espectrogramas u otras representaciones específicas, dependiendo del tipo de análisis que se desee realizar. Finalmente, es el propio investigador quien interpreta visualmente estas representaciones para identificar eventos relevantes como modos MHD, transiciones de régimen o comportamientos anómalos en el plasma.

En muchos casos, el análisis se ha basado en la inspección directa de espectrogramas generados con herramientas como wavelets [5], que permite visualizar la evolución en frecuencia de una señal. Sin embargo, este tipo de análisis requiere experiencia previa por parte del operador, y puede ser subjetivo, especialmente en casos con señales ruidosas o inestabilidades poco pronunciadas.

Además, debido a que cada investigador suele trabajar con sus propios scripts o rutinas, existe una falta de estandarización que dificulta la replicabilidad de resultados y la automatización de procesos.

Estas condiciones derivan en tiempos elevados para preparar y ejecutar cada análisis, una necesidad constante de intervención humana en todas las fases del flujo de trabajo, y dificultades añadidas a la hora de realizar estudios comparativos entre múltiples disparos o campañas experimentales.

En conjunto, el enfoque tradicional ha sido útil y ha permitido muchos avances, pero presenta limitaciones significativas en cuanto a escalabilidad, reproducibilidad y eficiencia, especialmente frente al volumen creciente de datos generado por cada campaña experimental. Estas limitaciones han motivado la exploración de enfoques más modernos basados en la automatización, la inteligencia artificial y la interacción mediante lenguaje natural, como se propone en este trabajo.

### **2.3. Avances Recientes en Modelos de Lenguaje Natural (LLMs)**

El desarrollo de modelos de lenguaje natural de gran escala ha supuesto una revolución en la forma en la que interactuamos con sistemas informáticos. Modelos como GPT-4, LLaMA 2 y LLaMA 3, PaLM y Claude son capaces de comprender preguntas en lenguaje natural, generar código, resumir documentos y automatizar consultas complejas.

Modelo	Organización	Tamaño	Acceso
GPT-4	OpenAI	170B	Privado/comercial
LLaMA 3	Meta	8B/70B	Abierto
Claude 3	Anthropic	N/D	Comercial
PaLM 2	Google	N/D	Limitado (Bard)

Cuadro 1: Comparativa de algunos modelos LLM recientes.

Estos modelos se han comenzado a aplicar en diversos campos científicos, permitiendo generar consultas SQL a partir de texto natural, automatizar el procesamiento de grandes volúmenes de datos y ofrecer interfaces conversacionales dentro de aplicaciones científicas.

## 2.4. Aplicaciones de IA en Fusión Nuclear

En el *ámbito* de la fusión nuclear, la IA puede aplicarse principalmente a tareas como la predicción de estados del plasma mediante redes neuronales, la clasificación de eventos utilizando técnicas de machine learning, y el control de sistemas en tiempo real a través de modelos predictivos.

Sin embargo, la aplicación de interfaces conversacionales en este contexto es todavía limitada. Algunos proyectos exploran interfaces gráficas simplificadas, pero la integración con modelos de lenguaje natural para consulta de datos sigue siendo un campo en desarrollo.

## 2.5. Visualización de Señales

El análisis visual sigue siendo una herramienta clave para validar la calidad de las descargas y detectar anomalías. En este trabajo se ha desarrollado un sistema capaz de interpretar la solicitud del usuario, acceder a los datos públicos del TJ-II y graficar señales específicas.

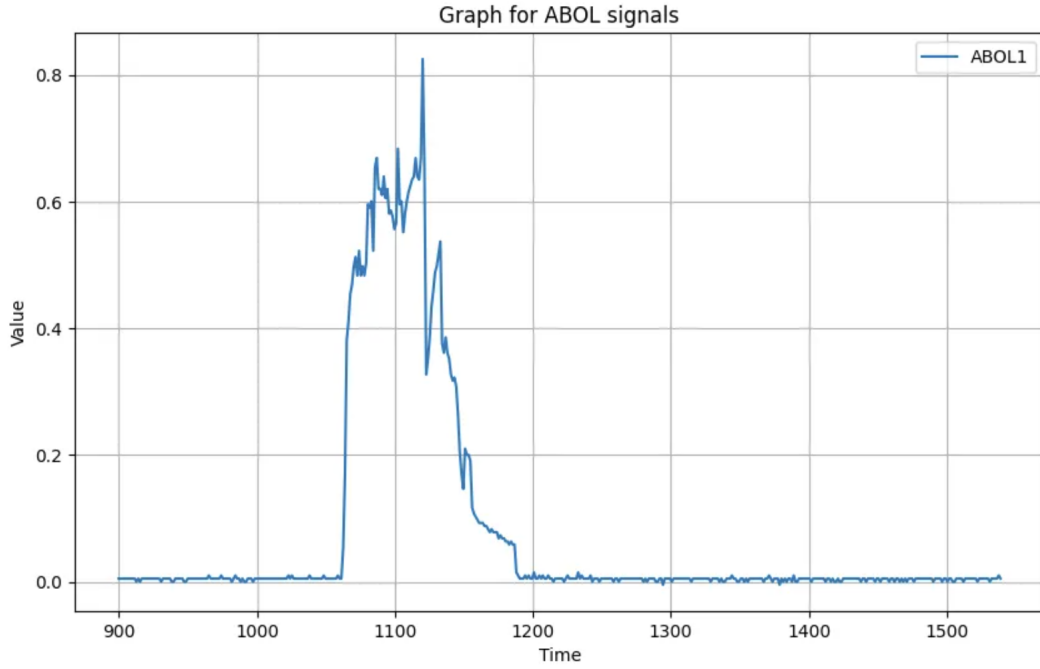


Figura 2: Ejemplo de visualización de señales. Fuente: TJ-II Data Display[4]

## 2.6. Predicción de Inestabilidades MHD con Espectrogramas

Las inestabilidades magnetohidrodinámicas (MHD) se manifiestan como perturbaciones oscilatorias que pueden detectarse en el dominio de la frecuencia a través de análisis espectral. Estas inestabilidades afectan negativamente al confinamiento del plasma y, por tanto, su detección temprana es crucial.

En este trabajo, se utiliza como base el sensor MIR5C, cuya señal se transforma mediante una técnica de espectrograma, convirtiendo la información temporal en una representación tiempo-frecuencia que permite visualizar la evolución de la potencia espectral. Para ello, se parte de una señal cruda  $y(t)$ , medida a intervalos temporales de 1 ms, y se realiza un análisis de Fourier utilizando la función de espectrograma de la biblioteca `scipy.signal`.

Matemáticamente, el espectrograma se obtiene aplicando una transformada de Fourier sobre ventanas deslizantes de la señal, según:

$$S(t, f) = \left| \int_{-\infty}^{\infty} y(\tau) w(\tau - t) e^{-j2\pi f\tau} d\tau \right|^2 \quad (1)$$

Donde  $S(t, f)$  representa el espectrograma en función del tiempo y la frecuencia,  $y(\tau)$  es la señal medida,  $w(\tau - t)$  es la ventana aplicada (en este caso, de tipo Hann), y  $f$  corresponde a la frecuencia.

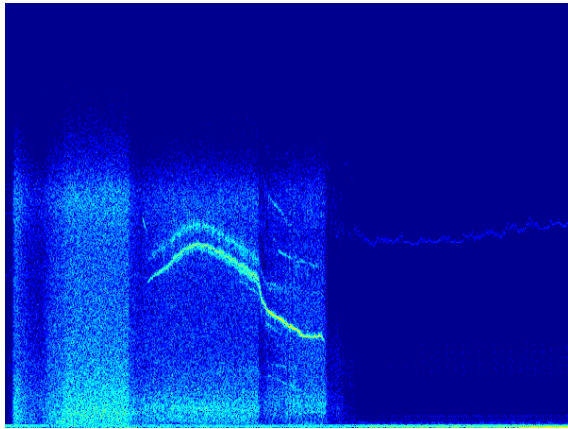
En el script `extttplot_spectrogram.py`, se utiliza una frecuencia de muestreo de  $f_s = 1000$  Hz (derivada de un paso temporal de 1 ms), un tamaño de ventana de  $n = 2^{10}$  y un solapamiento del 80 %. Posteriormente, se calcula la densidad espectral de potencia en decibelios:

$$B_{dB}(t, f) = 10 \cdot \log_{10} (|S(t, f)|) \quad (2)$$

Para mejorar la visualización, se realiza una normalización de la escala de color, adaptándola entre el valor mínimo y máximo del espectro:

$$B_{\text{norm}}(t, f) = \frac{B_{dB}(t, f) - B_{dB}^{\min}}{B_{dB}^{\max} - B_{dB}^{\min}} \quad (3)$$

El resultado se presenta en forma de imagen a color (colormap 'jet'), en escala de grises ('gray'), y como mapa de calor ('turbo').



(a) Espectrograma con MHD.



(b) Espectrograma sin MHD.

Figura 3: Comparación de espectrogramas.

En el espectrograma con MHD se aprecian bandas de alta intensidad en ciertas regiones de frecuencia que no están presentes en condiciones normales. Estas zonas representan oscilaciones coherentes del plasma, indicativas de fenómenos MHD. Para identificar estos patrones automáticamente, se emplea una técnica de extracción de características conocida como Histogram of Oriented Gradients (HOG), muy utilizada en visión por computador.

Esta técnica convierte cada imagen en un vector de características que capta los contornos y estructuras presentes, facilitando la clasificación mediante un modelo previamente entrenado. La clasificación final determina si la descarga presenta o no MHD.

Este enfoque presenta ventajas como la reducción del tiempo necesario para detectar eventos MHD, la automatización completa de un proceso tradicionalmente visual, y la integración directa con el sistema conversacional, lo que permite realizar predicciones a través del lenguaje natural.

## 2.7. Arquitectura de la Solución Propuesta

El sistema desarrollado tiene como objetivo simplificar y automatizar el análisis de los datos experimentales generados en el TJ-II. Para ello combina diversas tecnologías que permiten, por un lado, la interacción mediante lenguaje natural y, por otro, el procesamiento automático de señales y espectrogramas. A grandes rasgos, la solución integra cuatro bloques funcionales principales: la interpretación de lenguaje natural mediante el modelo LLaMA 3 (ejecutado localmente con Ollama), la generación automática de consultas SQL sobre datos en formato CSV, la descarga y visualización gráfica de señales desde el servidor de datos del CIEMAT [2], y la detección de inestabilidades MHD a través del análisis de espectrogramas con descriptores HOG y un clasificador SVM.

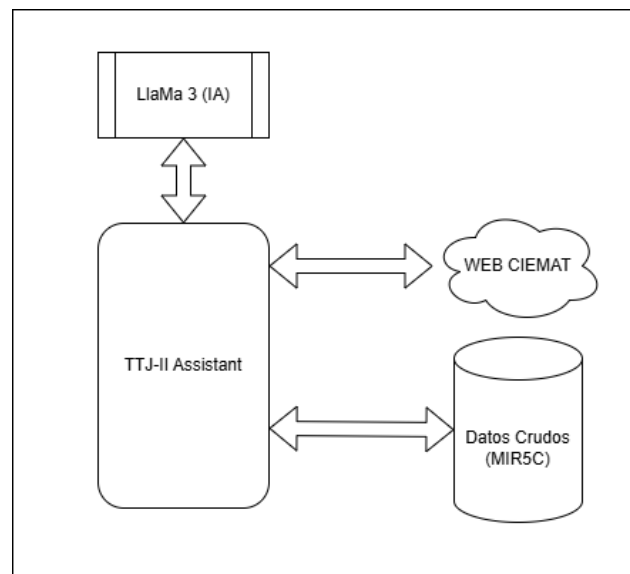


Figura 4: Arquitectura general del sistema propuesto.

El sistema se ha desarrollado íntegramente en **Python 3.12**, aprovechando su ecosistema de librerías científicas y de aprendizaje automático. El entorno de desarrollo utilizado ha sido **Visual Studio Code** (Visual Studio Code (VS Code)), por su integración con Git, soporte para extensiones y funcionalidades orientadas al desarrollo ágil.

Las principales librerías y herramientas empleadas son:

- **Gradio:** para la construcción de interfaces web ligeras y funcionales que permiten la interacción directa con el sistema [6].
- **Matplotlib y NumPy:** utilizadas para el cálculo numérico y la generación de gráficos científicos [7, 8].
- **Pandas:** para la manipulación y análisis de ficheros tabulares en formato CSV [9].
- **Scikit-image:** empleada en el procesado de espectrogramas como imágenes [10].
- **Scikit-learn:** utilizada para entrenar y aplicar el clasificador SVM [11].
- **FastAPI y Pydantic:** para la construcción y validación de servicios web mediante APIs REST [12, 13].
- **Certifi:** para la validación de certificados en conexiones seguras [14].
- **Ollama:** herramienta local para ejecutar modelos de lenguaje de gran tamaño (LLMs) de forma eficiente sin depender de servicios externos [15].
- **IBM Watson Machine Learning:** usado opcionalmente para desplegar modelos en la nube y realizar pruebas de inferencia remota [16].

Este conjunto de tecnologías permite desplegar un sistema capaz de procesar señales, generar espectrogramas, realizar predicciones automáticas sobre inestabilidades MHD, y ofrecer todo ello a través de una interfaz intuitiva basada en lenguaje natural.

## **2.8. Herramientas de automatización y análisis**

Existen otras herramientas para la automatización y análisis de las inestabilidades MHD, como puede ser el uso de redes neuronales profundas [17], a pesar de que ha sido un proceso de adopción lento. Existen varias formas de implementar esta técnica, como las redes convoluciones, que son aplicadas sobre los espectrogramas de señales Mirnov [18], ya que son capaces de detectar patrones visuales complejos, a su vez están las redes recurrentes, capaces de detectar secuencias temporales anómalas en las señales del plasma. Por último, están los autoencoders [19] y modelos no supervisados [20], los cuales detectan anomalías sin tener un etiquetado previo.

## **2.9. Conclusión del Estado de la Cuestión**

A lo largo de este capítulo se ha contextualizado la situación actual del análisis de datos en fusión, la irrupción de modelos de lenguaje natural, y el potencial de integrar IA en sistemas de consulta científica. La solución propuesta en este trabajo se posiciona como una herramienta pionera en la aplicación de interfaces conversacionales al entorno de datos del TJ-II, permitiendo una experiencia accesible, rápida y extensible para el análisis de descargas y detección de MHD.

En los capítulos siguientes se describirá con mayor detalle el diseño e implementación del sistema, así como la evaluación de su funcionamiento en distintos escenarios experimentales.

### **3. Metodología**

En esta sección se detalla el proceso seguido para el desarrollo del asistente conversacional, desde la interpretación de la entrada del usuario hasta la generación de gráficos y predicción de fenómenos MHD. Cada módulo se ha diseñado y probado de forma independiente para facilitar la integración final.

#### **3.1. Visión general del sistema**

El sistema ha sido desarrollado en tres módulos principales, lenguaje natural, visualización de datos experimentales y generación y análisis de espectrogramas para predicción de MHD. Esta modularización lo que nos permite es aislar individualmente cada componente, haciendo más fácil su mantenimiento y mejora individualizada.

El lenguaje en el cual se ha desarrollado ha sido Python, con la implementación de las librerías mencionadas en la sección Arquitectura de la Solución Propuesta.

Gradio ha sido la interfaz elegida para implementar el asistente conversacional debido a la flexibilidad de implementar tanto el texto como las imágenes. Junto a esto, Ollama, Matplotlib, Scikit-learn han sido esenciales, para la interpretación del lenguaje natural, la generación de gráficas, y aprendizaje de clasificación de espectrogramas.

La arquitectura ha sido modular, implementada mediante script independientes cuyo módulo central es main.py, el cual se encarga de la coordinación según la intención del usuario.

#### **3.2. Procesamiento del lenguaje natural**

La capacidad de interpretar y comprender las necesidades del usuario a través de lenguaje natural es uno de los pilares fundamentales del sistema. Para ello, la integración del LLM LLaMa 3 ha sido decisiva con su ejecución interna mediante la plataforma de Ollama. Este modelo ayuda a clasificar la intencionalidad, extraer parámetros relevantes de manera estructurada y la traducción de peticiones en instrucciones ejecutables.

Esto permite a los investigadores interactuar de manera intuitiva con los datos del TJ-II.

### 3.2.1. Clasificación de la intención del usuario

La intencionalidad del usuario es el fin con el cual el usuario hace la pregunta. En este caso tenemos 4 intencionalidades: mostrar gráficos de señales (PLOT), responder preguntas sobre los datos dentro de un CSV (CSV), predecir si un espectrograma tiene MHD (PREDICT), o la general, la cual responderá a preguntas generales.

La clasificación de la intencionalidad se hace mediante el uso del LLM, el cual va a clasificar de manera automática a través del uso de la función `determine_intent()`, donde se ha metido el número, nombre y función de intencionalidades.

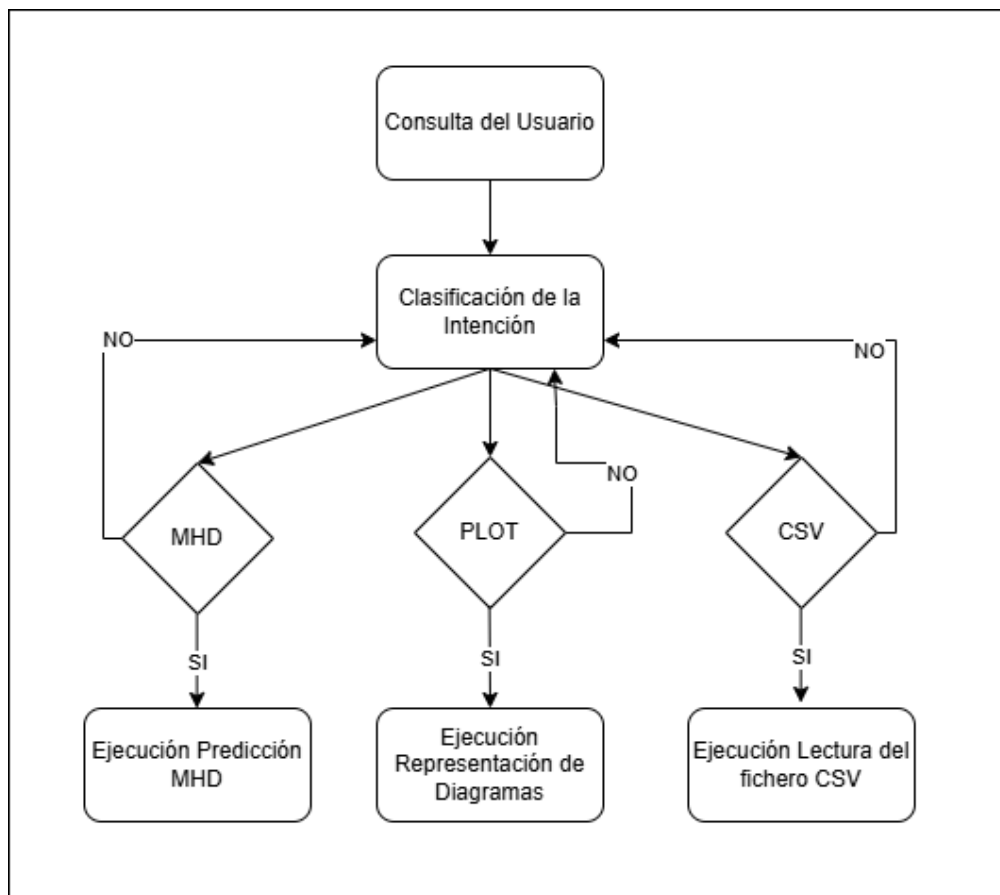


Figura 5: Clasificación de Intención de la consulta del usuario

### 3.2.2. Extracción de parámetros estructurados

En el caso de que el usuario decida obtener una gráfica de una o varias señales, el sistema, mediante el uso de LLM y un fichero de texto, interpreta la petición para obtener los valores im-

portantes de manera independiente, es decir, va a buscar todos los parámetros configurables dentro de la petición del usuario.

Estos parámetros son los siguientes:

- **Shot:** El cual va a determinar el número de descarga con el cual el sistema imprimirá las gráficas.
- **Signals:** Esta variable es la cual se encarga de guardar una lista de señales que queremos que sean impresas en forma de gráfica. Para ello se utiliza un fichero de texto, guardado en la configuración del programa, el cual va a hacer una comparación del nombre de las señales que han sido encontradas por el LLM con las que se tienen guardadas en el fichero. Si el nombre coincide, se podrá imprimir la gráfica de dichas/s señales.
- **Tstart:** Se encarga de determinar el tiempo de inicio de la gráfica, es decir, del tiempo inicial de la representación de la gráfica. Esta variable no es obligatoria ponerla, ya que tiene como valor predeterminado 0 ms, pero en caso de querer elegir un valor distinto, se podría especificar.
- **Tstop:** Al igual que 'tstart' determina el tiempo, pero en este caso el final de la gráfica, es decir, el tiempo final de la representación de la gráfica. No es obligatoria al tener un valor predeterminado de 2000 ms, pero en caso de querer elegir un valor distinto, se podría especificar.

Todas estas variables que son sacadas mediante el uso principalmente del LLM, junto a la ayuda del fichero de texto, son devueltas por el LLM en formato JSON, para su futura configuración a la hora de la llamada para obtener los valores.

### 3.2.3. Consultas SQL generadas por IA

Una vez el sistema ha reconocido que la intención de la petición es sobre el fichero CSV, se procederá a la creación de la consulta mediante el uso de SQL. Esto lo hace el LLM, se encarga de interpretar y traducir la petición al lenguaje SQL. Una vez la tenemos, se buscará dentro del fichero CSV con la consulta SQL con el uso de la librería Python pandasql [9].

Una vez hemos hecho la consulta, el LLM se encargará de limpiar la respuesta a una respuesta más legible y fácil de interpretar.

### 3.3. Consulta de datos experimentales

#### 3.3.1. Generación de URL con parámetros

Después de haber extraído de los parámetros comentados en el apartado Extracción de parámetros estructurados, procedemos a la construcción de la URL que vamos a usar para la lectura de los datos provenientes de un servidor interno del CIEMAT.

Esta URL está compuesta por la cabecera, la cual será siempre la misma:

```
https://info.fusion.ciemat.es/cgi-bin/TJII_data.cgi
```

a la cual añadiremos las variables que hemos extraído anteriormente.

La URL quedaría de la siguiente forma:

```
{base_url}?shot={shot}&nsignal={nsignal}&signal={signal_name}  
&fact={factor}&tstart={tstart}&tstop={tstop}
```

Esta URL devolverá un HTML con todos los datos obtenidos de la señal con dichos valores.

### 3.4. Obtención y extracción de datos desde HTML

Cuando se realiza una petición al servidor interno del CIEMAT, este responde con una página HTML que contiene los datos asociados a la señal solicitada [4]. La comunicación se lleva a cabo mediante el método HTTP GET, lo que significa que los parámetros de consulta se codifican directamente en la URL.

El HTML recibido no está diseñado específicamente para ser interpretado automáticamente, por lo que se requiere un proceso de análisis estructurado para extraer los datos útiles. En este proyecto, el parseo del contenido HTML se ha realizado mediante el uso de expresiones regulares (*re*), lo que permite identificar patrones textuales dentro del documento sin necesidad de bibliotecas externas específicas para manipulación de HTML.

Concretamente, se buscan bloques de texto que contienen los valores de las señales en formato plano o parcialmente estructurado. A partir de esos bloques, se extraen secuencias de datos numéricos que representan los valores. Estos datos se transforman en listas de puntos, estructuradas en diccionarios, donde la clave es el nombre de la señal y el valor es una lista de tuplas ordenadas.

### 3.4.1. Agrupación y visualización de señales

Una vez extraídos y estructurados, estos datos pueden ser utilizados para su tratamiento posterior, como la generación de gráficas utilizando la biblioteca Matplotlib [7]. Además, se ha implementado un sistema de agrupación de señales por prefijo común mediante expresiones regulares, lo que facilita una representación más clara y organizada de los resultados.

Cada grupo de señales se representa de manera conjunta en un mismo gráfico, ya que ayuda al entendimiento e interpretación de estas. Para la visualización de las gráficas, se almacenan temporalmente en memoria, para convertirlas en imágenes, con el objetivo de mandarlas a través de la interfaz conversacional.

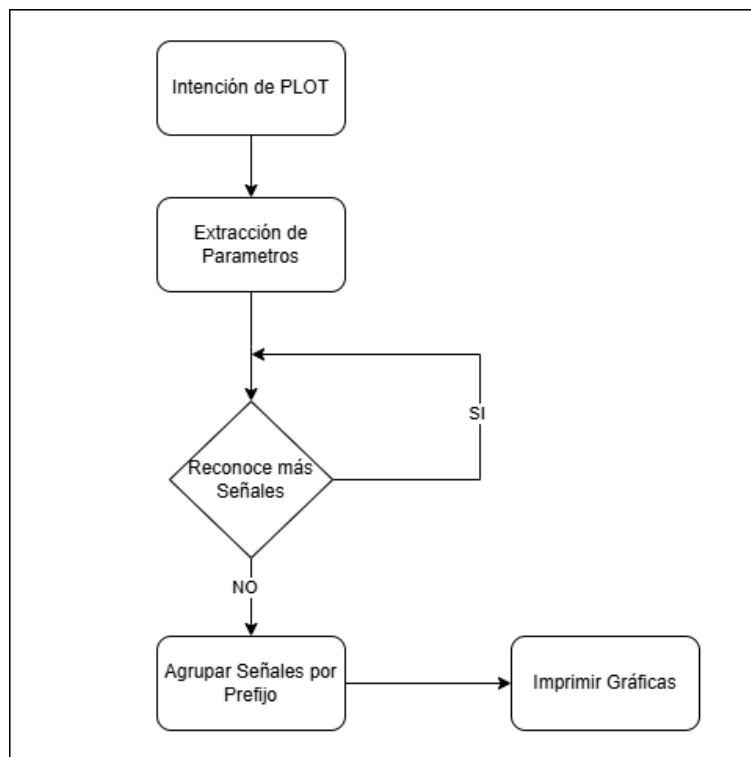


Figura 6: Agrupación e impresión de señales

### 3.5. Generación y análisis de espectrogramas

#### 3.5.1. Construcción de espectrogramas a partir de señales crudas

Para la detección de las inestabilidades MHD, se ha optado por la representación de las señales experimentales mediante el uso de espectrogramas, específicamente de las señales del sensor MIR5C, una señal muy representativa de la observación de las oscilaciones del plasma.

La construcción de los espectrogramas ha empezado mediante la carga de los datos en crudo del disparo, el cual queremos analizar, almacenado en formato .txt, que contiene las muestras de tiempo y amplitud. Seguidamente, se escoge un intervalo de tiempo de interés. Generalmente, se escoge entre 1050 ms y 1250 ms, ya que es empíricamente cuando las inestabilidades se manifiestan de manera más clara.

Una vez se ha escogido el intervalo, aplicamos la transformada de Fourier de ventana deslizante [21], con el objetivo de obtener el espectrograma deseado. Esto se hace con el uso de la función `spectrogram()` de la librería `SciPy.Signal`, configurándola con una ventana de tipo Hann [22].

El resultado de esta transformación es una matriz tridimensional que representa la densidad espectral de la potencia a lo largo del tiempo y la frecuencia. Esta matriz se convierte en una escala logarítmica en decibelios y posteriormente normalizada para facilitar la visualización.



Figura 7: Construcción de espectrogramas

### 3.5.2. Transformación espectral y generación de imágenes

Una vez se ha transformado y obtenido la matriz normalizada, se han generado tres versiones de gráficas del espectrograma

- Una imagen en escala de color tipo `jet` (azul-rojo) que representa directamente la potencia en dB.
- Una versión normalizada en escala de grises.
- Una versión en falso color utilizando el mapa `turbo`, más apropiado para visualizaciones tipo mapa de calor.

Estas son guardadas y utilizadas posteriormente para la predicción automática de las inestabilidades MHD.

### 3.5.3. Extracción de características y predicción MHD

Tras la generación de las imágenes del espectrograma, se procede al análisis automático para la detección de la posibilidad de la presencia de inestabilidades MHD. Esto se hace mediante la extracción de características y clasificación.

Para ello, se cargan las tres versiones de los espectrogramas generados en el apartado anterior. Se convierten en escala de grises y se redimensionan a un formato uniforme (128x128 píxeles) para su posterior tratamiento.

Para la extracción de características se ha utilizado el método HOG, usado para capturar la estructura y orientación de los bordes presentes en las imágenes. Cada una de las tres imágenes aporta vectores de características, y los tres vectores se conectan en un solo vector de 24300 dimensiones.

Este vector resultado se usará para crear un modelo de clasificación de tipo Máquinas de Vectores de Soporte (SVM) [23], el cual será entrenado para distinguir entre espectrogramas que contienen o no MHD. Para esto se ha usado la librería Python `joblib` [24], la cual es un conjunto de herramientas para proporcionar una canalización ligera en Python. Esta librería te proporcionará con un archivo de tipo `.pkl` [25], siendo este el modelo serializado.

Una vez tenemos creado el modelo de clasificación, se podrá usar para la clasificación de cualquier espectrograma, el cual devolverá un texto de tipo "MHD Yes." "MHD No", el cual en pasos futuros el LLM interpretará la salida para pasarlo a una respuesta más elaborada.

### 3.6. Interfaz conversacional y flujo de ejecución

#### 3.6.1. Interacción con el usuario mediante Gradio

La interacción entre el usuario y el asistente es completamente desarrollada a través de una interfaz web de Python conocida como Gradio [6], la cual te permite crear de manera sencilla interfaces gráficas para scripts de Python o modelos de aprendizaje, dando la posibilidad de desplegar un asistente altamente accesible desde cualquier navegador.

La interfaz está compuesta de un cuadro de texto donde el usuario podrá escribir consultas en lenguaje natural, otra cuadro de texto de salida que devolverá el texto en lenguaje natural generado por el LLM, y un último cuadro de imágenes, donde se podrán ver todas las imágenes de salida proporcionadas por el sistema, respondiendo a la consulta del usuario. El sistema es capaz de interpretar consultas como:

- Generar gráficas de señales experimentales (PLOT).
- Consultar datos numéricos desde un fichero CSV (CSV).
- Predecir la presencia de inestabilidades MHD en un disparo determinado (PREDICT).
- Responder preguntas generales mediante un modelo de lenguaje (GENERAL).

Esta interfaz lo que permite es interactuar con gran cantidad de volúmenes de datos sin necesidad de conocimientos técnicos específicos y avanzados, facilitando en gran medida los experimentos del TJ-II [1].

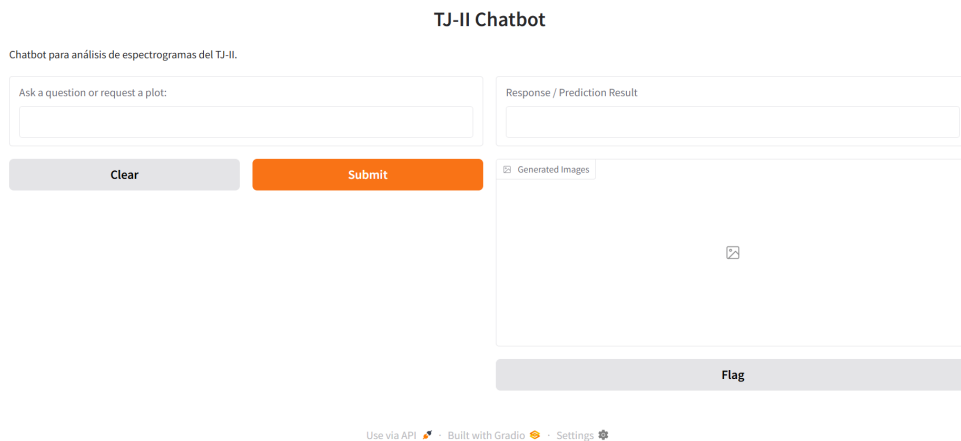


Figura 8: Interfaz de Gradio

### 3.6.2. Orquestación de módulos y lógica central

La orquestación del sistema está controlada por el script `main.py`, el cual tiene la función principal encargada de distribuir las funciones según la intención de la consulta del usuario. El sistema está modularizado, lo que permite la escalabilidad de las funciones.

El flujo de orquestación es el siguiente:

- **Consulta del usuario:** El usuario escribe en lenguaje natural una consulta.
- **Interpretación de la consulta:** Se determina la intención con la cual el usuario ha hecho la consulta
- **Extracción de datos:** Se extraen todos los datos importantes encontrados en la consulta del usuario.
- **Invocación del módulo:** Para la resolución de la consulta se invoca el módulo relacionado con la intención en conjunto con los datos extraídos.
- **Muestra de los resultados:** El sistema se encarga de mostrar en forma de texto y/o imágenes el resultado de la consulta.

Todo esto permite al usuario el uso de manera sencilla de todas las posibilidades que da el sistema.

### 3.7. Ficheros de configuración y estructura del proyecto

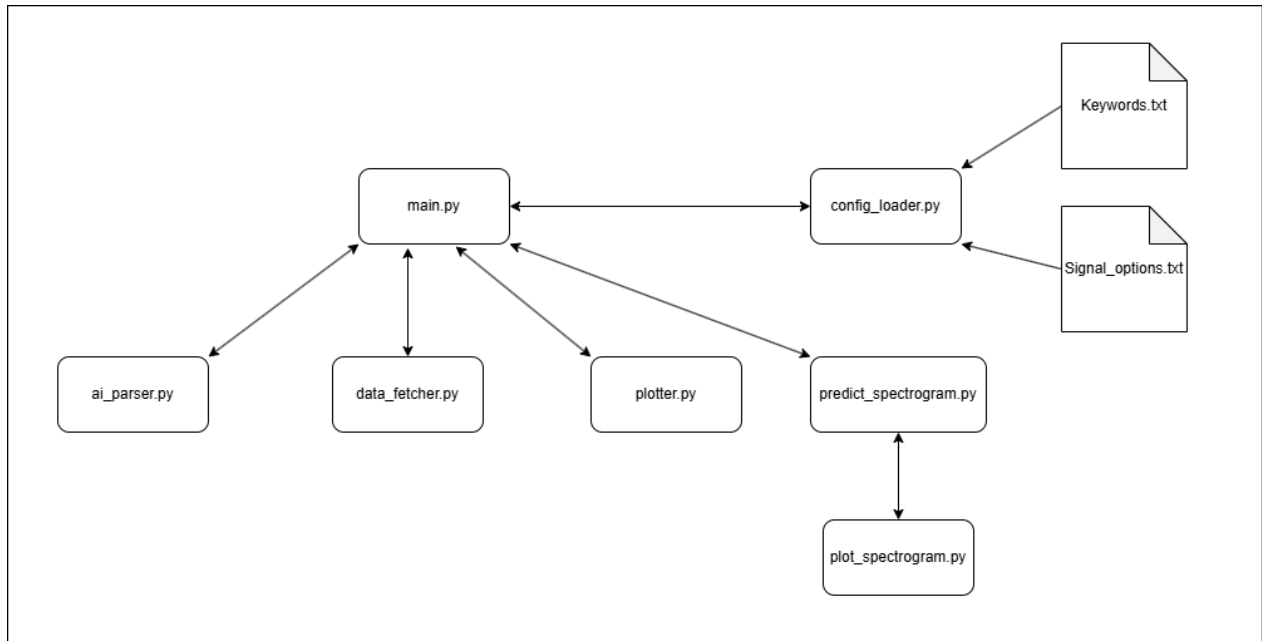


Figura 9: Sistema de ficheros

#### 3.7.1. Ficheros de configuración y estructura del sistema

El sistema está diseñado de forma modular y configurable. Para facilitar su uso y mantenimiento, utiliza dos archivos de texto (`Keywords.txt` y `Signal_options.txt`) que permiten definir palabras clave y señales disponibles sin necesidad de modificar el código. Además, cada componente del sistema está separado en ficheros con responsabilidades específicas: `main.py` gestiona la interfaz y el flujo de la aplicación; `ai_parser.py` procesa lenguaje natural y genera consultas SQL; `data_fetcher.py` descarga y prepara los datos; `plotter.py` y `plot_spectrogram.py` generan las representaciones gráficas; `predict_spectrogram.py` realiza la inferencia sobre inestabilidades MHD mediante SVM; y `config_loader.py` se encarga de cargar los ficheros de configuración.

## 4. Análisis de resultados

En esta sección se presentan los resultados obtenidos después de haber implementado todas las funciones del sistema. Los resultados se han dividido en dos partes principales: resultados analíticos, que cuantifican el rendimiento del sistema, y resultados visuales, que permiten verificar de manera cualitativa el funcionamiento.

### 4.1. Resultados analíticos

Para evaluar cuantitativamente el funcionamiento del sistema, se ha empleado un conjunto de 124 espectrogramas etiquetados manualmente (99 con inestabilidad MHD y 25 sin ella). La Tabla 2 resume su distribución.

Cuadro 2: Distribución del conjunto de evaluación

Etiqueta	Número de espectrogramas	Porcentaje
MHD (Y)	99	79.84 %
No MHD (N)	25	20.16 %

El sistema ha procesado cada espectrograma, generado su espectrograma de tiempo–frecuencia, extraído descriptores HOG y aplicado el clasificador SVM entrenado previamente. Posteriormente, se han comparado las predicciones con las etiquetas reales para construir la matriz de confusión que se muestra en la Tabla 3.

Cuadro 3: Matriz de confusión del clasificador

RealPred.	MHD (Y)	No MHD (N)
MHD (Y)	92	7
No MHD (N)	1	24

A partir de esta matriz se han obtenido las métricas de precisión (*precision*), sensibilidad (*recall*) y F1-score que se recogen en la Tabla 4.

Cuadro 4: Métricas por clase

Clase	Precision	Recall	F1-score
MHD (Y)	0.99	0.93	0.96
No MHD (N)	0.77	0.96	0.86
<b>Macro-media</b>	0.88	0.94	0.91

La precisión global es del 94

Estas métricas confirman la validez del sistema para la detección automática de inestabilidades; sin embargo, los falsos positivos y falsos negativos identificados (Tabla 3) indican margen de mejora, aspecto que se comenta en el apartado de resultados visuales.

## 4.2. Resultados visuales

En este apartado se verán los resultados obtenidos de forma visual:

### 4.2.1. Visualización de espectrogramas con clasificación de inestabilidad MHD

Como se puede observar en la imagen, el espectrograma ha sido clasificado con inestabilidad de tipo MHD, ya que se puede ver la franja de frecuencia y duradera en un periodo extenso de tiempo, característico de una inestabilidad MHD. Esto determina la presencia de oscilaciones.

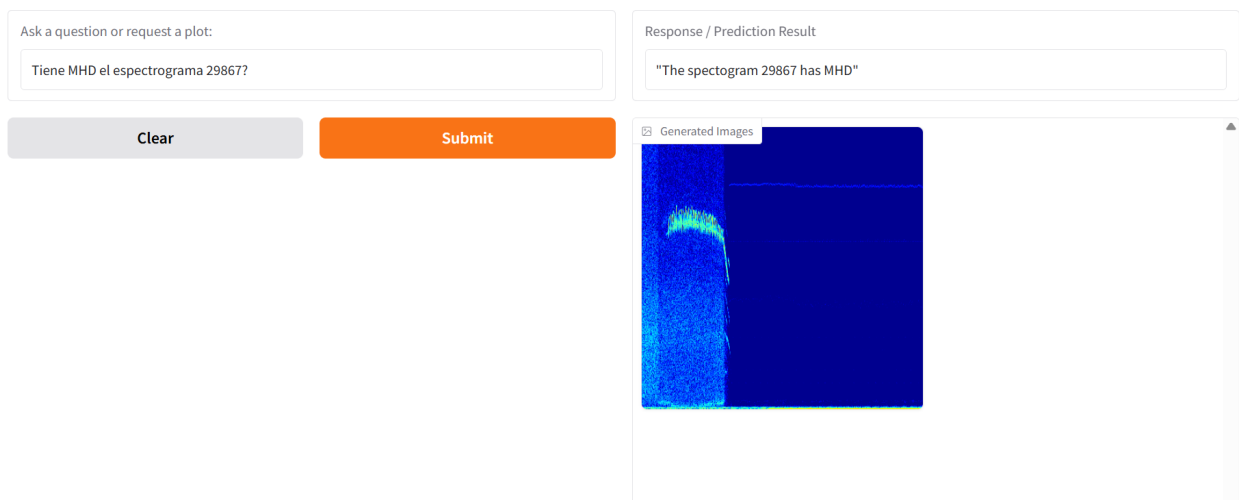


Figura 10: Espectrograma 29867 clasificado con inestabilidad MHD

### 4.2.2. Visualización de espectrogramas sin clasificación de inestabilidad MHD

La imagen representa un espectrograma clasificado como estable, es decir, que el sistema no ha encontrado franjas de frecuencia persistentes en el tiempo.

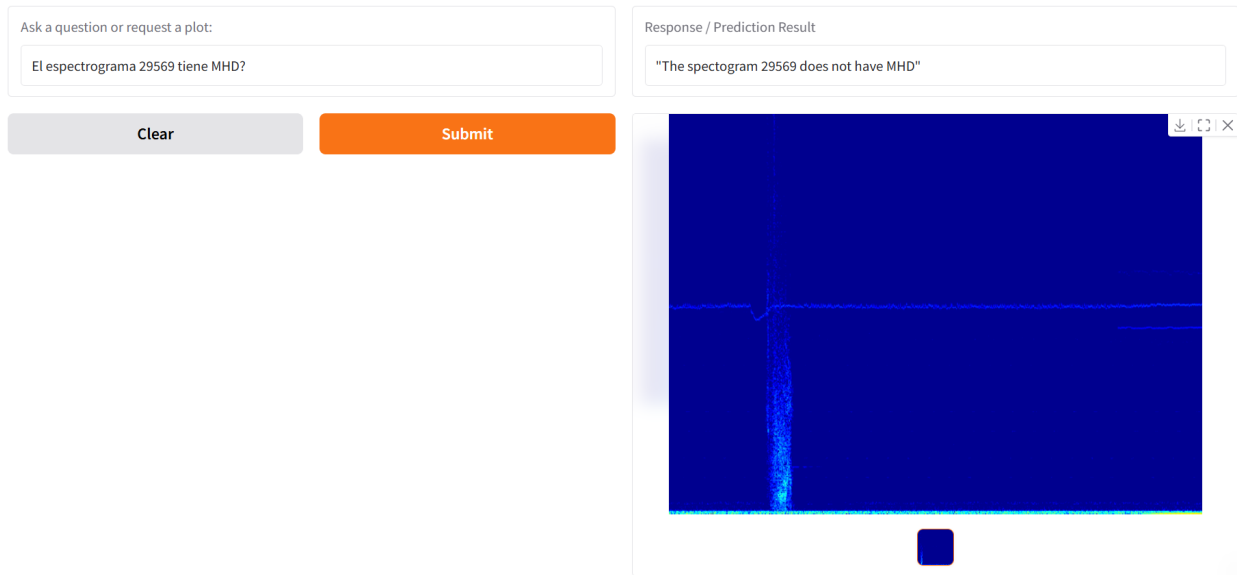


Figura 11: Espectrograma 29569 clasificado sin inestabilidad MHD

#### 4.2.3. Visualización de señales

El sistema, como comentado anteriormente, es capaz de generar gráficas a partir de comandos en lenguaje natural. Esto ha sido implementado y probado con éxito, aunque con ciertas limitaciones, como la identificación de los nombres de señales, los cuales actualmente tienen que ser exactos.

La siguiente imagen muestra cómo el sistema reconoce las señales para ser capaz de representarl

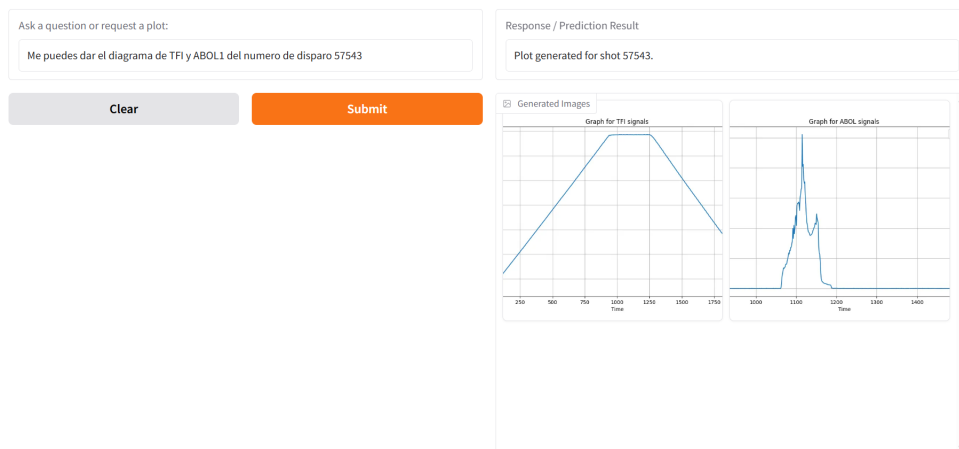


Figura 12: Diagrama TFI y ABOL1 de la señal 57543

En el caso de representar señales en las que tienen prefijo común, se representaría de la siguiente forma:

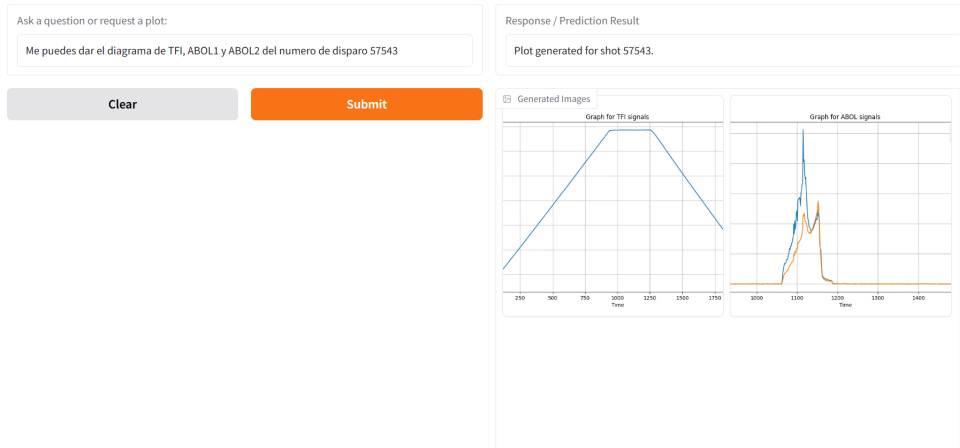


Figura 13: Diagrama TFI, ABOL1 y ABOL2 de la señal 57543

Como se puede observar, las señales de prefijo común se representan en un mismo diagrama con una leyenda, para diferenciar cuál es cuál.

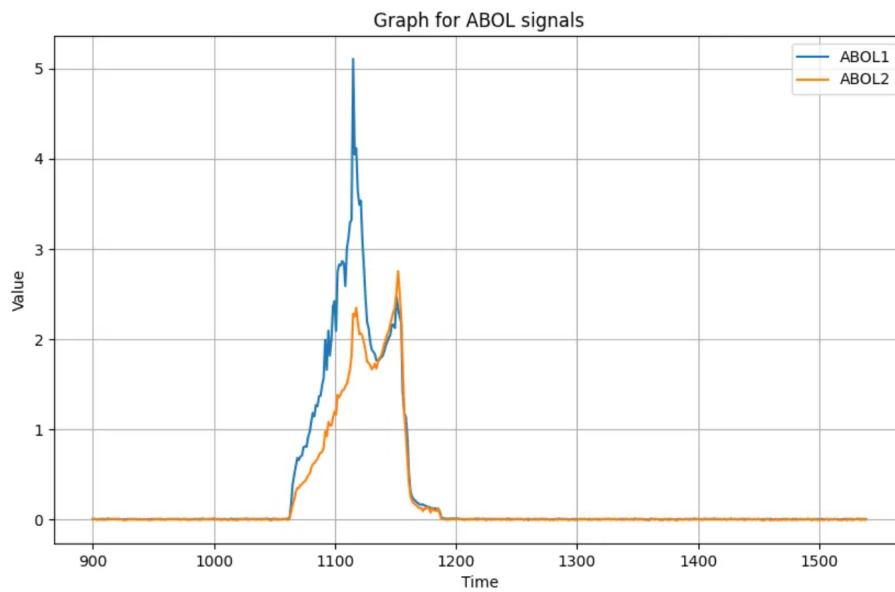


Figura 14: Diagrama ABOL1 y ABOL2 de la señal 57543

## **5. Conclusiones y trabajo futuro**

### **5.1. Conclusiones generales**

Durante el desarrollo de este Proyecto de Fin de Grado (PFG), se ha observado que, la implementación de IA en ámbitos en los que los procesos son repetitivos, con una cantidad de datos grande y sobre todo técnicos, es muy beneficiosa. Todo este progreso conseguido con la IA, ha cambiado por completo la forma de resolver y automatizar una cantidad grande de tareas.

La implementación de LLMs como el usado en la aplicación, ha resultado ser altamente eficaz para mejorar la calidad y rapidez de obtener resultados. A pesar de ciertas limitaciones que pueden tener los modelos de IA, los resultados obtenidos han sido prometedores y confirman que el desarrollo y uso de los mismos va a ser clave en un futuro cercano.

A su vez, el impacto que tiene el uso de estos modelos tanto social como medioambiental es grande, ya que contribuyen con ciertos objetivos dentro de los Objetivos de Desarrollo Sostenible (ODS).

Este proyecto está teniendo mucha repercusión en el área de la fusión nuclear, ya que se está haciendo con el objetivo de implementar inteligencia artificial en los dispositivos de fusión nuclear, como el ITER [26].

### **5.2. Líneas futuras**

Durante este PFG, el cual se ha basado en el diseño e implementación de un sistema automático capaz de detectar e interpretar los datos experimentales generados por el TJ-II [1]. Con el objetivo de crear una herramienta capaz de agilizar y facilitar el proceso de analizar datos en un entorno orientado a la fusión, donde se recoge una cantidad de datos muy grande. En cuanto a los resultados obtenidos, han sido muy relevantes, pero puede haber cabida de mejora de los mismos.

Como comentado en otros apartados, se ha conseguido desarrollar un sistema que no solo interprete y analice inestabilidades MHD, sino que sea capaz de imprimir gráficas de datos de las distintas señales dentro del dispositivo e interpretar los datos de un CSV donde se van guardando datos por cada disparo que se hace.

En todas estas funciones es posible la mejora de rendimiento y precisión.

Por parte de análisis de inestabilidades HMD, como el modelo de tipo SVM [23], el cual ha sido entrenado con un número relativamente pequeño de espectrogramas, podría ser entrenado con muchas más descargas con el objetivo de refinar todavía más la precisión a la hora de clasificar si contiene o no dicha inestabilidad.

En cuanto a la interpretación de los datos del CSV, depende mucho el modelo de IA que se use. Durante este proyecto se ha usado un modelo instalado de manera local, lo que reduce la capacidad y velocidad de cómputo del modelo, por lo que los tiempos aumentan considerablemente. El objetivo es conseguir implementar un modelo en la nube, el cual ahorra tiempo de cómputo y espacio, ya que no tienes que tener descargado en local un modelo, los cuales suelen ocupar un espacio considerable. A su vez, la claridad de los prompts donde declaras la finalidad de la función dentro del modelo, es clave para obtener los mejores resultados posibles y es un punto donde cabe la posibilidad de la mejora.

En la representación de gráficas, el punto a mejorar es a la hora de reconocer la señal que el usuario desea representar, ya que se puede mejorar la exactitud. Actualmente, existe un fichero de configuración, que tiene el nombre de todas las señales posibles, pero lo ideal sería hacerlo mediante una petición directa al servidor del CIEMAT [4], donde devolvería los nombres de las posibles señales existentes. A su vez, otra de las cosas que se puede mejorar es que el modelo LLM interprete y encuentre la señal más parecida a la que el usuario ha introducido, ya que en caso de que el usuario escriba de manera errónea el nombre de la señal, el sistema puede no reconocerlo y no imprimir ninguna gráfica.

### **5.3. Impacto social y medioambiental**

La implementación de IA para la automatización de procesos repetitivos y con grandes cantidades de datos está muy ligada con algunos de los ODS [27]. Los ODS son un conjunto de objetivos globales adoptados por todos los estados pertenecientes a las Naciones Unidas para abordar desafíos globales.

#### **5.3.1. Impacto social**

El uso de IA en ámbitos de fusión nuclear beneficia a la hora de reducir el tiempo de trabajo manual, y mejora la innovación en el sector, directamente relacionado con el ODS 9: Industria,

innovación e infraestructura. El ODS pretende construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación [28].

### **5.3.2. Impacto medioambiental**

Este proyecto además ayuda directamente a la posibilidad de mejorar la investigación en todo este sector el cual tiene el objetivo de crear fuentes de energía sostenibles capaces de abastecer al mayor número de personas posibles, por lo que está relacionado con el ODS 7: Energía asequible y no contaminante. El ODS 10 pretende garantizar el acceso a una energía limpia y asequible, clave para el desarrollo de la agricultura, las empresas, las comunicaciones, la educación, la sanidad y el transporte [29].

Los modelos de IA (LLMs) son característicos por tener un consumo energético muy elevado, debido a la ejecución para la inferencia, además del consumo previo, con la preparación y el entrenamiento del modelo. Los costos de entrenar un modelo pueden superar los 500000 kWh, como ha sido el caso Llama 3, el modelo utilizado en este proyecto. Este valor puede ser aproximado al consumo de un avión comercial grande en un viaje de aproximadamente 7 horas. La única diferencia es que el entrenamiento se hace una sola vez, ya que puede ser reutilizado en múltiples instancias. [30]

Las prácticas como estas cuyo objetivo de combatir el cambio climático y sus efectos están relacionadas con el ODS 13: Acción por el clima. El ODS 13 tiene como objetivo adoptar medidas urgentes para combatir este problema, que afectará a las personas de todos los países de todos los continentes de alguna forma [31].

## Referencias

- [1] CIEMAT. Tj-ii – dispositivo de fusión nuclear tipo stellarator, 2024. URL <https://www.fusion.ciemat.es/inicio/tj-ii/>.
- [2] CIEMAT. Centro de investigaciones energéticas, medioambientales y tecnológicas, 2024. URL <https://www.ciemat.es/>.
- [3] Wikipedia Contributors. Magnetohidrodinámica, 2024. URL <https://es.wikipedia.org/wiki/Magnetohidrodin%C3%Almica>.
- [4] CIEMAT. Tj-ii experimental data access, 2025. URL [https://info.fusion.ciemat.es/cgi-bin/TJII\\_data.cgi](https://info.fusion.ciemat.es/cgi-bin/TJII_data.cgi).
- [5] Universidad Nacional Autónoma de México (UNAM). Transformada wavelet, 2025. URL [https://virtual.cuautitlan.unam.mx/intar/?page\\_id=1108](https://virtual.cuautitlan.unam.mx/intar/?page_id=1108).
- [6] Gradio Team. Gradio, 2024. URL <https://gradio.app>.
- [7] Matplotlib Developers. Matplotlib, 2024. URL <https://matplotlib.org>.
- [8] NumPy Contributors. Numpy, 2024. URL <https://numpy.org>.
- [9] Pandas Team. Pandas, 2024. URL <https://pandas.pydata.org>.
- [10] S. van der Walt et al. Scikit-image, 2024. URL <https://scikit-image.org>.
- [11] F. Pedregosa et al. Scikit-learn, 2024. URL <https://scikit-learn.org>.
- [12] Sebastián Ramírez. Fastapi, 2024. URL <https://fastapi.tiangolo.com>.
- [13] Pydantic Developers. Pydantic, 2024. URL <https://docs.pydantic.dev>.
- [14] Certifi Developers. Certifi, 2024. URL <https://pypi.org/project/certifi/>.
- [15] Ollama Team. Ollama, 2024. URL <https://ollama.com>.
- [16] IBM Cloud. Ibm watson machine learning, 2024. URL <https://www.ibm.com/cloud/machine-learning>.
- [17] IBM Corporation. ¿qué es el deep learning?, 2025. URL <https://www.ibm.com/es-es/think/topics/deep-learning>.

- [18] Wikipedia. Oscilaciones de mirnov, 2025. URL [https://en.wikipedia.org/wiki/Mirnov\\_oscillations](https://en.wikipedia.org/wiki/Mirnov_oscillations).
- [19] Medium. Desmitificando las redes neuronales: detección de anomalías con autoencoder, 2025. URL <https://medium.com/@weidagang/demystifying-anomaly-detection-with-autoencoder-neural-networks-1e235840>
- [20] IBM Corporation. ¿qué es el aprendizaje no supervisado?, 2025. URL <https://www.ibm.com/de-de/think/topics/unsupervised-learning>.
- [21] Wikipedia. Transformada de fourier, 2025. URL [https://es.wikipedia.org/wiki/Transformada\\_de\\_Fourier](https://es.wikipedia.org/wiki/Transformada_de_Fourier).
- [22] Wikipedia. Ventana de hann, 2025. URL [https://es.wikipedia.org/wiki/Ventana\\_\(funci%C3%B3n\)\\_r](https://es.wikipedia.org/wiki/Ventana_(funci%C3%B3n)_r).
- [23] IBM Corporation. ¿qué es una máquina de vectores de soporte (svm)?, 2025. URL <https://www.ibm.com/es-es/think/topics/support-vector-machine>.
- [24] Joblib Developers. Persistence and serialization with joblib, 2025. URL <https://joblib.readthedocs.io/en/latest/persistence.html>.
- [25] FILEExt. Extensión de archivo pkl – ¿qué es un archivo .pkl y cómo abrirlo?, 2025. URL <https://filext.com/es/extension-de-archivo/PKL>.
- [26] ITER ORG. Energía de fusión, 2025. URL <https://www.iter.org/>.
- [27] United Nations. Objetivos de desarrollo sostenible, 2025. URL <https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>.
- [28] United Nations. Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación, 2025. URL <https://www.un.org/sustainabledevelopment/es/infrastructure/>.
- [29] United Nations. Objetivo 7: Garantizar el acceso a una energía asequible, segura, sostenible y moderna, 2025. URL <https://www.un.org/sustainabledevelopment/es/energy/>.

- [30] Communications of the ACM. La huella energética de los humanos y los grandes modelos lingüísticos, 2025. URL <https://cacm.acm.org/blogcacm/the-energy-footprint-of-humans-and-large-language-models/>.
- [31] United Nations. Objetivo 13: Adoptar medidas urgentes para combatir el cambio climático y sus efectos, 2025. URL <https://www.un.org/sustainabledevelopment/es/climate-change-2/>.

## **Lista de Siglas**

**CIEMAT** Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas. 2–4, 9

**CSV** Comma-Separated Values. 3, 4, 8, 16, 17, 20, 21, 26

**HOG** Histogram of Oriented Gradients. 15, 16, 25, 29

**IA** Inteligencia Artificial. 3, 7–9, 13, 33–35

**LLMs** modelos de lenguaje grandes. 9, 33, 35

**MHD** Magnetohidrodinámica. 3, 4, 8–11, 14–16, 18–20, 24–26

**ODS** Objetivos de Desarrollo Sostenible. 33–35

**PGF** Proyecto de Fin de Grado. 33

**SVM** Máquinas de Vectores de Soporte. 25

**VS Code** Visual Studio Code. 17