



UNIVERSIDAD DE GRANADA

TRABAJO FIN DE GRADO
INGENIERÍA EN INFORMÁTICA

PATRON ML

Patrones de Comportamiento en Trastornos del Neurodesarrollo en Población Infantil usando Machine Learning

Autor

Daniel Carrasco Moreno

Directores

Alberto Fernández Hilario
José Manuel Soto Hidalgo



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, diciembre de 2025

Índice general

1. Introducción	1
2. Fundamentos	6
2.1. Trastornos del neurodesarrollo y TDAH	6
2.1.1. Trastornos del neurodesarrollo	6
2.1.2. TDAH: descripción general e infancia	7
2.2. Machine Learning	12
2.2.1. Validación cruzada	15
2.2.2. Métricas de evaluación para clasificación desbalanceada	15
2.3. Algoritmos de clasificación	19
2.4. Preparación y calidad del dato	21
2.4.1. Imputación univariada mediante K vecinos	21
2.5. Fundamentos de análisis exploratorio y asociaciones	22
2.5.1. Correlación biserial puntual (binaria–continua)	22
2.5.2. Correlación de Spearman (ordinal/binaria–ordinal)	22
2.5.3. Prueba Chi-cuadrado de independencia (categórica–binaria)	23
2.6. Normalización	23
2.6.1. Codificación <i>One-Hot</i> (variables categóricas)	24
2.6.2. Estandarización (normalización <i>Z-score</i>)	24
3. Estado del Arte	25
3.1. Introducción	25
3.1.1. Contexto TDAH	25
3.1.2. Objetivos de la revisión	25
3.1.3. Metodología Revisión del Estado del Arte	26
3.2. Revisión y síntesis de la literatura	26
3.2.1. Métodos tradicionales de análisis de datos en psicología infantil	26
3.2.2. Evolución hacia Machine Learning	26
3.2.3. Trabajos relacionados basados en Machine Learning	27
3.2.4. Limitaciones transversales y lagunas detectadas	28
3.3. Conclusiones de la revisión y encaje de la propuesta	28
4. Planificación y presupuesto	30
4.1. Planificación del proyecto	30
4.1.1. Fases	30
4.1.2. Diagrama de Gantt	35
4.2. Estudio presupuestario	37
4.2.1. Licencias de uso	37
4.2.2. Recursos materiales	38

4.2.3. Costes de personal	38
4.2.4. Costes de despliegue y mantenimiento	40
4.2.5. Otros costes asociados	40
4.2.6. Presupuesto total	41
5. Metodología	42
5.1. Mapa del proceso	43
5.2. Desarrollo metodológico	44
5.2.1. Definición y métricas	44
5.2.2. Ingesta y <i>split</i> de datos	45
5.2.3. Preprocesado (“higiene de datos”)	45
5.2.4. EDA + Feature Engineering + Selección	46
5.2.5. Entrenamiento, validación y calibración	46
5.2.6. Empaquetado e inferencia del pipeline	46
5.2.7. API (FastAPI) + UI (Streamlit)	46
5.2.8. Despliegue con contenedores	46
5.3. Propuesta de Enfoque	47
6. Análisis experimental: Caso NSCH-2023	49
6.1. Descripción del dataset	49
6.2. Preprocesamiento de datos	50
6.2.1. Armonización y controles de calidad	50
6.2.2. Manejo de ausencias y saltos lógicos	50
6.2.3. Codificación numérica para el pipeline	51
6.2.4. Imputación de valores faltantes	51
6.2.5. Resultado	51
6.3. EDA y Feature Engineering	51
6.3.1. Fusión de características	52
6.3.2. Análisis Estadístico Descriptivo y estudio de <i>outliers</i>	52
6.3.3. Discriminación por sexo	54
6.3.4. Análisis de correlaciones respecto a la variable objetivo	54
6.3.5. Hitos de la fase EDA	59
6.3.6. Selección de características	59
6.3.7. Dataset final para entrenamiento del modelo	60
6.4. Machine Learning y entrenamiento	62
6.4.1. Objetivo del entrenamiento y función de pérdida	62
6.4.2. Técnicas utilizadas para el entrenamiento	62
6.4.3. Modelado y evolución del proceso	63
6.4.4. Hiperparámetros, calibración y umbral de decisión	69
6.4.5. Árbol de decisión. Interpretación y versión definitiva	69
6.5. Evaluación del modelo	72
6.5.1. Evaluación del modelo sobre el conjunto de <i>test</i>	72
6.5.2. Comparación del rendimiento por conjuntos de variables	73
6.5.3. Análisis de <i>pruning</i> del <i>Decision Tree</i>	78
6.6. Software, entornos y control de versiones	79
6.6.1. Plataformas de trabajo	79
6.6.2. Programas externos	79
6.6.3. Entorno base y librerías de ciencia de datos	79
6.6.4. Control de versiones	80

7. Desarrollo y Despliegue del Sistema	81
7.1. Diseño del sistema	81
7.1.1. Estructura de carpetas	82
7.1.2. Requisitos funcionales y no funcionales	83
7.2. Implementación	85
7.2.1. Entrenamiento del modelo y selección de variables	85
7.2.2. Backend: API y pipeline de inferencia	85
7.2.3. Frontend: interfaz y comunicación con la API	90
7.2.4. Docker y Docker Compose	97
7.2.5. Ingeniería del software y buenas prácticas	98
7.3. Validación y demostración funcional	100
7.4. Software, entornos y control de versiones	114
7.4.1. Plataformas de desarrollo	114
7.4.2. Frameworks y librerías de la aplicación	114
7.4.3. Control de versiones	114
8. Discusión, conclusiones y trabajos futuros	116
8.1. Conclusiones generales	116
8.2. Líneas de trabajo futuro	117
8.3. Discusión	117
8.3.1. Comparación con el estado del arte	117
8.3.2. Discusión metodológica y conclusiones experimentales	119
8.4. Limitaciones del estudio	120
8.5. Reflexión	120
Bibliografía	122
Apéndices	123
A. Variables utilizadas en el modelo	124
B. Variables del conjunto de entrenamiento final	126
C. Entrenamiento modelo Random Forest	128
D. Entrenamiento y validación del Random Forest	129
E. Variables dataset decodificadas	130
F. Diagramas	131

Resumen

Patrones de Comportamiento del Neurodesarrollo en Población Infantil usando Machine Learning

Daniel Carrasco Moreno

Palabras clave: *TDAH, Aprendizaje Automático, Machine Learning, Sistema Web, Tecnología en salud, Educación y salud, Diagnóstico clínico*

Resumen

El Trastorno por Déficit de Atención e Hiperactividad (TDAH) representa uno de los desafíos más significativos en el ámbito del neurodesarrollo infantil, cuyo diagnóstico depende en gran medida de observaciones clínicas a menudo subjetivas, resultando en posibles inconsistencias y retrasos en el diagnóstico lo que conlleva en la mayoría de casos, problemas tanto para los menores afectados así como para su entorno familiar debido al impacto de este trastorno en su correcto desarrollo social y académico. Este trabajo, bajo el proyecto global '**Patrón ML**', aborda este desafío desde una doble perspectiva: la ciencia de datos y el *Machine Learning (ML)* así como la ingeniería de software.

El objetivo principal no es solo el de desarrollar una aplicación *web* como herramienta de apoyo al diagnóstico, si no realizar un estudio metodológico completo para identificar patrones objetivos y accionables asociados al TDAH en cuestionarios educativos y/o de salud infantil, materializando dichos hallazgos en un prototipo funcional.

El núcleo del proyecto ha sido un riguroso *pipeline* de Ciencia de Datos. Utilizando el extenso conjunto de datos de la *National Survey of Children's Health* (NSCH) del año 2023, se ejecutó un proceso exhaustivo que abarcó desde la ingestión y una meticulosa preparación de los datos ("higiene de datos"), hasta proceso exhaustivo de ingeniería de características. Se gestionaron explícitamente desafíos críticos como la fuga de información ('*data leakage*') y el severo desbalance de clases que presenta el NSCH.

El modelado se centró en la interpretabilidad como pilar fundamental. Más allá de la simple precisión, se buscaba comprender por qué ciertas variables (conductuales, socioeconómicas y de salud) son relevantes para el diagnóstico del TDAH. Por ello, se seleccionó y optimizó un modelo de Árbol de Decisión.

Dado el desbalance notable en el *dataset* utilizado, el desafío era doble: se debía minimizar el coste de los **falsos negativos**, que representan niños con TDAH no detectados, lo cual exigía una alta Sensibilidad/*Recall*. Pero también, el coste de los **falsos positivos** o niños sanos derivados para evaluación, lo que exigía una alta precisión y evitar así una saturación en el sistema sanitario de la salud mental.

Es por ello que, en lugar de priorizar una métrica sobre la otra, se buscó un equilibrio óptimo entre ambas. Por este motivo, se adoptó como métrica objetivo de referencia el ***F1-score*** para ajustar el umbral de decisión, al ser una media armónica entre las dos medidas anteriores. Tras una calibración probabilística del modelo (mediante un *CalibratedClassifierCV*), se seleccionó el umbral que maximizaba este *F1-score*, logrando así un equilibrio robusto entre la capacidad de detección del modelo y la fiabilidad de sus predicciones.

Paralelamente, para ir más allá de un experimento académico, el proyecto implementó un enfoque de '*MLOps ligero*'. Este se centró en garantizar la trazabilidad y reproducibilidad conectando la fase de experimentación con la de producción.

El resultado más tangible de este estudio es '**TDAHTool**', una aplicación *web* desarrollada como parte integral del resultado del proyecto. Esta herramienta no es el fin en sí misma, sino la demostración de la aplicabilidad del estudio. Desarrollada con una arquitectura moderna y desacoplada (FastAPI para el *backend*, Streamlit para el *frontend*) y desplegable mediante contenedores Docker, TDAHTool sirve como un sistema de apoyo a la decisión. Su verdadero valor reside en las funcionalidades de explicabilidad , como el análisis exploratorio e interactivo con el usuario mediante la manipulación de las principales variables detectadas por el modelo (análisis "*what-if*") y la visualización de la ruta de decisión exacta del modelo, permitiendo al profesional no solo ver una predicción, sino comprender el razonamiento subyacente.

En conclusión, 'Patron *ML*' entrega un prototipo completo, transparente y reproducible que ofrece un apoyo objetivo al diagnóstico, validando el poder del *ML* para arrojar luz sobre patrones complejos del neurodesarrollo.

Abstract

Neurodevelopmental Behavioural Patterns in Child Populations using Machine Learning

Daniel Carrasco Moreno

Keywords: *TDAH, Machine Learning, Web System, Health technology, Education and health, Clinical diagnosis*

Abstract

Attention Deficit Hyperactivity Disorder (ADHD) represents one of the most significant challenges in the field of child neurodevelopment. Its diagnosis depends heavily on clinical observations that are often subjective, resulting in possible inconsistencies and diagnostic delays. In most cases, this leads to problems for both the affected children and their family environment due to the disorder's impact on their proper social and academic development. This work, under the umbrella project 'Patrón ML', addresses this challenge from a dual perspective: data science, machine learning (ML) and software engineering.

The main objective is not only to develop a web application as a diagnostic support tool, but rather to conduct a complete methodological study to identify objective and actionable patterns associated with ADHD in educational and/or child health questionnaires, materializing these findings into a functional prototype.

The core of the project has been a rigorous Data Science pipeline. Using the extensive dataset from the 2023 National Survey of Children's Health (NSCH), an exhaustive process was executed, ranging from data ingestion and meticulous preparation ('data hygiene') to a thorough feature engineering process. Critical challenges such as 'data leakage' and the severe class imbalance present in the NSCH were explicitly managed.

Modeling was centered on interpretability as a fundamental pillar. Beyond simple accuracy, the goal was to understand why certain variables (behavioral, socioeconomic, and health-related) are relevant to the diagnosis of ADHD. Therefore, a Decision Tree model was selected and optimized.

Given the notable imbalance in the dataset used, the challenge was twofold: the cost of false negatives—representing children with ADHD who go undetected—had to be minimized, which demanded high Sensitivity/Recall. But also, the cost of false positives—or healthy children referred for evaluation—which demanded high precision to avoid saturating the mental healthcare system.

For this reason, rather than prioritizing one metric over the other, an optimal balance between the two was sought. Therefore, the F1-score was adopted as the reference target metric for adjusting the decision threshold, as it is a harmonic mean of the two previous measures. After a probabilistic calibration of the model (using a CalibratedClassifierCV), the threshold that maximized this F1-score was selected, thus achieving a robust balance between the model's detection capability and the reliability of its predictions.

In parallel, to go beyond a mere academic experiment, the project implemented a 'lightweight MLOps' approach. This focused on ensuring traceability and reproducibility by connecting the experimentation phase with the production phase.

The most tangible result of this study is 'TDAHTool', a web application developed as an integral part of the project's outcome. This tool is not an end in itself, but rather a demonstration of the study's applicability. Developed with a modern, decoupled architecture (FastAPI for the backend, Streamlit for the frontend) and deployable via Docker containers, TDAHTool serves as a decision support system. Its true value lies in its explainability features, such as the exploratory and interactive analysis with the user by manipulating the main variables detected by the model ('what-if' analysis) and the visualization of the model's exact decision path, allowing the professional to not only see a prediction but to understand the underlying reasoning.

In conclusion, 'Patron ML' delivers a complete, transparent, and reproducible prototype that offers objective support for diagnosis, validating the power of ML to shed light on complex neurodevelopmental patterns.

Agradecimientos

A mi madre y a mi padre por su constante apoyo y ánimo durante toda esta aventura, sin vuestro apoyo esto no hubiera sido posible. También a mi hermano Alejandro.

A Miguel, Juan Manuel, Jesús y Pablo por estar en los peores momentos, así como el resto de amigos que han estado ahí siempre.

Y, por ultimo, gracias a Alberto y José Manuel por toda la ayuda proporcionada durante este TFG y por darme la oportunidad de trabajar con profesionales tan excepcionales. Ha sido un lujo trabajar y aprender con vosotros.

Capítulo 1

Introducción

El Trastorno por Déficit de Atención e Hiperactividad (TDAH) es un trastorno del neurodesarrollo que afecta a un porcentaje relevante de la población infantil y adolescente en la actualidad, con un impacto directo en el rendimiento académico, en la socialización y en el bienestar emocional. A pesar de la extensa experiencia clínica acumulada durante las últimas décadas, el diagnóstico sigue asentado en procedimientos subjetivos (entrevistas, escalas, observación), lo que abre espacios tanto a la heterogeneidad como a posibles retrasos en la detección temprana. En el presente contexto, **PATRON ML** se plantea como una respuesta técnica y metodológica, en el que se tiende un puente entre una gran cantidad de datos y la decisión clínica/educativa, aportando evidencia cuantitativa y herramientas interpretables de apoyo al juicio profesional.

La pregunta planteada en este proyecto es clara: ¿se puede, con un enfoque riguroso de Ciencia de Datos y *Machine Learning*, objetivar señales relacionadas con el TDAH a partir de un *dataset* amplio, heterogéneo y actual, para operacionarlas en una aplicación *web* usable por personal docente y demás profesionales clínicos? El presente trabajo demuestra que sí, articulando 2 vertientes:

1. Un estudio de Ciencia de Datos y *ML* orientado a descubrir patrones de comportamiento y factores asociados al TDAH con especial énfasis en la interpretabilidad y la calibración del modelo.
2. La materialización aplicada de todo este conocimiento obtenido del proceso de Ciencia de Datos en **TDAHTool**, un prototipo *web* con explicabilidad local, concebido como apoyo a la decisión y no como sustituto del criterio clínico.

El diagnóstico de TDAH se construye hoy a partir de múltiples fuentes (entrevistas, observación en contextos distintos, escalas estandarizadas), con criterios DSM-5 (American Psychiatric Association, 2013) y pruebas específicas como Conners (Conners, 2000), pero no existe una “prueba única”. Esta pluralidad, valiosa, también introduce variabilidad y puede retrasar intervenciones críticas en etapas sensibles del desarrollo. **PATRON ML** no persigue “automatizar el diagnóstico”, sino aportar señales cuantitativas y transparencia que complementen la práctica clínica y educativa, ayudando a priorizar casos y a argumentar derivaciones con mejor trazabilidad.

Introducción

Encaje con el Estado del Arte

La investigación reciente sobre TDAH y *ML* ha recorrido tres caminos que se complementan. En primer lugar, los estudios de gran escala basados en *encuestas poblacionales* (por ejemplo, *National Survey of Children's Health*, NSCH) muestran que es viable extraer patrones clínicos y psicoeducativos con modelos supervisados, identificando factores de riesgo y protectores y alcanzando rendimientos competitivos. Este enfoque, empleado en el trabajo de Maniruzzaman (Maniruzzaman et al. (2022)) que combina selección de señales con clasificadores modernos, ha sido la principal referencia conceptual para este TFG. En nuestro caso, recogemos ese testigo pero **actualizamos** el marco de datos a **NSCH 2023**, ampliamos y depuramos la preparación del dato y priorizamos la **interpretabilidad** desde el diseño del *pipeline*.

En paralelo, una segunda línea se apoya en *pruebas informatizadas de rendimiento atencional* (Slobodin et al. (2020)) con distracciones visuales y auditivas, logrando métricas altas en muestras clínicas escolares al combinar índices de tarea con covariables básicas (edad, sexo, momento de la prueba). Estos trabajos demuestran que la *captura estandarizada del comportamiento* puede alimentar modelos precisos, pero suelen operar con *muestras moderadas y modalidad única* (la propia prueba), lo que limita la generalización fuera del laboratorio. Nuestro enfoque aprende de esa fortaleza, la medición conductual, pero la reintegra en un **conjunto amplio y heterogéneo** (clínico, educativo y contextual) como el de NSCH, buscando señales **más transferibles** a entornos reales.

Un tercer frente explora *plataformas interactivas y serious games* con sensores móviles (acelerómetro/giroscopio) para capturar hiperactividad y patrones de juego (inatención/impulsividad), reportando cifras prometedoras de *precisión*, sensibilidad y especificidad (Zakani et al. (2023)). Aun así, la literatura subraya su talón de Aquiles: *tamaños muestrales reducidos*, dependencia del *hardware* y dudas sobre la *validez externa* y el mantenimiento en contextos diversos. Como respuesta, este trabajo apuesta por **datos representativos** y una **arquitectura reproducible** que pueda convivir con estas nuevas fuentes en el futuro, pero sin depender de ellas para demostrar valor hoy.

En síntesis, **Patrón ML** se inspira en la evidencia de que el *ML* puede complementar el diagnóstico del TDAH, pero propone elevar el listón en tres frentes: **calidad y trazabilidad del dato, interpretabilidad calibrada orientada a decisiones y operacionalización con garantías de reproducibilidad**.

Objetivo general

El proyecto persigue construir y poner a prueba un *pipeline* de *Machine Learning* interpretable y bien calibrado que, a partir de variables clínicas, psicoeducativas y de contexto, apoye el cribado del **TDAH en población infantil**. La finalidad última no es reemplazar el juicio profesional, sino aportar señales claras, trazables y fiables que puedan llevarse a la práctica a través de una herramienta *web* reproducible, sustentada en un conjunto de datos amplio, reciente y bien caracterizado, como es el NSCH.

Objetivos específicos (Ciencia de Datos y *ML*)

En el plano analítico, se establece como primer objetivo comprender y preparar el dato con rigor: decodificar fielmente el *dataset* y preservar su semántica, revisar la calidad, tratar los valores nulos y identificar y plasmar la lógica de los '*logical skips*'; a continuación, codificar según la naturaleza de cada variable (binaria, categórica, ordinal, numérica) e imputar cuando proceda con criterios coherentes, evitando la fuga de información a lo largo del ciclo de validación. Este esfuerzo se apoya en la elección de NSCH 2023 como fuente, por su cobertura temática, actualización y tamaño muestral (55.162 menores), y en una curación inicial de 82 variables informadas por evidencia clínica y revisiones recientes.

Sobre una base depurada, se plantea explorar y convertir el dato en conocimiento mediante una *EDA* cuidadosa que no solo atienda a distribuciones, sino también a relaciones útiles, construcción/depuración de rasgos y convergencia hacia un conjunto explicativo. Con ese conjunto, se entrenará y validará el modelo con validación cruzada estratificada en un escenario desbalanceado, aplicando búsqueda de hiperparámetros y poda cuando sea necesario. La calibración probabilística y el ajuste de un umbral operativo alineado con el uso de cribado garantizarán que las salidas del sistema sean utilizables en contexto clínico/educativo.

Finalmente, se evaluará donde importa: en *test* independiente, con métricas sensibles al problema y con análisis de robustez que permita interpretar los resultados y sus límites.

Objetivos específicos (TDAHTool y MLOps ligero)

Para que el conocimiento sea transferible al punto de decisión, se aspira a empaquetar el *pipeline* de inferencia en un artefacto único que incluya preprocesado, modelo y calibración, junto a un contrato de entrada y metadatos/métricas mínimas que aseguren consistencia entre entrenamiento e inferencia, todo ello en una aplicación *web* final usable por el usuario con perfil psicopedagógico, clínico o docente.

Sobre ese artefacto, se expondrá una *API* (FastAPI) y una interfaz (Streamlit) con explicabilidad local: ruta de decisión por caso y análisis "*what-if*" para apoyar la discusión clínica/educativa. El sistema se desplegará de forma portable mediante Docker Compose, con servicios desacoplados, *healthchecks*, dependencias versionadas y volumen de modelo en solo lectura, siguiendo un *MLOps* ligero (inspirado en CRISP-DM) e integración continua en GitHub (ramas cortas, *pull requests* con revisión, verificaciones automáticas de formato y *build*/arranque de contenedores), de modo que la solución resulte repetible, trazable y portable.

Aportación Diferencial y Enfoque Metodológico

La aportación diferencial del presente proyecto se asienta en cinco pilares metodológicos que conectan la fuente del dato con la herramienta final:

1. Rigor en la Fuente de Datos y su Preparación

El proyecto se fundamenta en un rigor metodológico desde la base. Como fuente de datos, se utiliza **NSCH 2023** (U.S. Census Bureau, 2023), una encuesta nacional de EEUU con una muestra inicial de 55.162 reportes de menores (0-17 años) que recoge un amplio espectro de variables de salud, contexto y rendimiento. Esta amplitud permite abordar el TDAH con una evidencia representativa.

Sobre esta fuente, el primer pilar fue una preparación exhaustiva del dato: la encuesta se decodificó en *STATA* (StataCorp LLC, s.f.) para preservar etiquetas y significados, trabajando en *CSV* para el procesamiento en *Python*. En el preprocesado, se distinguen cuidadosamente los valores ausentes reales de los *logical skips* propios de los cuestionarios; se armonizan tipos, se codifica según la naturaleza de cada variable (binaria, categórica, ordinal, numérica) y como técnica de imputación de valores nulos, se imputan valores con *KNN* cuando procede, documentando el mapeo en un *pipeline* reproducible que impide la fuga de información.

2. Modelado Interpretable y Calibrado en Contexto Desbalanceado

En problemas sensibles como el cribado de TDAH, no basta con 'acertar mucho'. Por ello, se prioriza un clasificador transparente, como el Árbol de Decisión, cuya lógica puede auditarse. Con el terreno limpio (Pilar 1), se realiza una *EDA* para buscar relaciones útiles y depurar rasgos. Sobre esa base, se entrena con validación cruzada estratificada (dado el desbalance de clases), se ajustan hiperparámetros y, crucialmente, se calibran las probabilidades (*CalibratedClassifierCV*). Finalmente, se fija un umbral operativo alineado con un uso de cribado (optimizado con curvas *precision-recall*), para que las predicciones sean fiables y útiles en la práctica.

3. Explicabilidad Accionable en el Punto de Decisión

La explicabilidad no se queda en un gráfico de barras: se vuelve actionable. La herramienta muestra la ruta de decisión concreta que el modelo sigue para cada caso (qué condiciones se cumplen, en qué orden y con qué confianza parcial), permitiendo al profesional ver el razonamiento del sistema y contrastarlo con su propio juicio. A esto se suma un análisis "*what-if*" que facilita explorar escenarios hipotéticos: ¿cómo cambiaría el riesgo si mejorara cierto patrón conductual? Este diálogo con el modelo convierte la predicción en palanca de conversación clínica/educativa.

4. Transferencia Real: de la Ciencia de Datos a TDAHTool

Para cerrar el círculo, el proyecto culmina en **TDAHTool**. Se empaqueta un artefacto único (preprocesado + modelo + calibración + contrato de entrada) y se expone en una aplicación desacoplada: una *API* (FastAPI) y una *UI* (Streamlit), concebida para perfiles psicopedagógicos, clínicos o docentes. El sistema se orquesta con *Docker Compose* para garantizar portabilidad y reproducibilidad de extremo a extremo (red interna, *healthchecks* y modelo en solo lectura).

5. Capa Operativa: *MLOps* Ligero e Integración Continua (*CI*)

Finalmente, se cuida cómo se construye y entrega la solución. Se adopta un enfoque de ***MLOps* ligero** (inspirado en CRISP-DM) que estructura el repositorio de GitHub¹ por capas (ciencia de datos, *backend*, *frontend*) con versionado. Las fases analíticas se organizan en *notebooks* parametrizados. A nivel de *CI*, el flujo de trabajo en GitHub emplea ramas cortas, *pull requests* con revisión y **verificaciones automáticas** (formato, construcción de contenedores y

¹Repositorio del proyecto en GitHub: <https://github.com/darmor99/TDAHTool/tree/main>

Introducción

build/arranque de contenedores).

En conjunto, este *MLOps* conecta ciencia de datos y software, haciendo el sistema **repetible, trazable y portable**.

Resultados y Conclusión de la Propuesta

La evaluación del modelo final en el conjunto de *test* independiente confirma la consistencia y utilidad clínica de la propuesta: se alcanza un *Recall* (TDAH) entre 0,65 y 0,68, una *Especificidad* $\approx 0,94$ y un *F1-score* $\approx 0,61$. Este balance, obtenido tras el ajuste del umbral de decisión, prioriza no dejar pasar casos (sensibilidad) sin disparar un número excesivo de falsos positivos, alineándose con los requisitos de un sistema de cribado.

Aunque las métricas son moderadamente aceptables para este escenario, el resultado se interpreta como un punto de partida sólido: una base reproducible y operativa sobre la que se podrá iterar para perfeccionar el modelo (incorporación de nuevas variables, estrategias avanzadas de muestreo y calibración, evaluación externa) y madurar la aplicación hacia su uso real por profesionales de la salud mental y la educación, integrando progresivamente validaciones clínicas, mejoras de explicabilidad y requisitos de usabilidad y ética.

Capítulo 2

Fundamentos

Este capítulo establece el marco teórico y metodológico que sustenta la memoria. En primer lugar, se presenta una síntesis de los trastornos del neurodesarrollo, con especial énfasis en el TDAH, abarcando su definición clínica, principales subtipos y manifestaciones en la infancia y adolescencia, así como los retos diagnósticos y las implicaciones para la intervención educativa y sanitaria.

A continuación, se exponen los fundamentos de *ML* relevantes para el problema, incluyendo el ciclo de vida del dato, la preparación y codificación de variables, el equilibrio de clases, la validación y la evaluación con métricas apropiadas para clasificación. Finalmente, se revisan los algoritmos de clasificación más utilizados en este ámbito —desde métodos lineales y árboles de decisión hasta ensambles y modelos de margen— destacando sus supuestos, ventajas, limitaciones y criterios de selección en contextos clínicos y educativos.

2.1. Trastornos del neurodesarrollo y TDAH

Esta sección presenta los fundamentos conceptuales de los trastornos del neurodesarrollo, centrando el análisis de forma específica en el TDAH.

Se realiza una descripción general del TDAH, abordando sus presentaciones clínicas (inatenta, hiperactiva/impulsiva y combinada), su sintomatología principal y la complejidad de sus causas, que abarcan desde factores genéticos hasta ambientales.

Finalmente, se analiza el proceso diagnóstico actual, basado en los criterios del DSM-5 y el uso de herramientas como el *Test de Conners*. Se destaca la alta prevalencia del trastorno y la dificultad para diferenciarlo de otras condiciones, lo que subraya la necesidad de explorar nuevas herramientas tecnológicas (como las basadas en *ML*) para mejorar su detección.

2.1.1. Trastornos del neurodesarrollo

Los trastornos del neurodesarrollo constituyen un conjunto de condiciones que se originan en las primeras etapas del desarrollo del sistema nervioso central y que afectan de manera significativa diversas áreas del funcionamiento personal, social, académico o laboral de quienes los padecen. De acuerdo con la American Psychiatric Association (2013), estos trastornos se caracterizan por déficits en el desarrollo que provocan limitaciones en una o más esferas del desempeño cotidiano, y suelen manifestarse durante la infancia, persistiendo a lo largo del ciclo vital.

Estas condiciones pueden implicar alteraciones en la comunicación, el aprendizaje, el comportamiento, las habilidades motoras o las interacciones sociales, y su **diagnóstico temprano** es fundamental para establecer intervenciones eficaces. Entre los **trastornos del neurodesarrollo** más reconocidos se encuentran el Trastorno del Espectro Autista (TEA), los trastornos del aprendizaje, el trastorno del desarrollo del lenguaje, y, especialmente, el **TDAH**, que constituye el objeto central de este trabajo (Centers for Disease Control and Prevention, 2023).

En los últimos años, el estudio del TDAH ha cobrado una especial relevancia debido a su elevada prevalencia, la complejidad asociada a su diagnóstico y la necesidad de desarrollar herramientas más objetivas, precisas y accesibles para su detección. En este contexto, el avance de la inteligencia artificial (IA), y en particular de las técnicas de aprendizaje automático (*ML*), ha abierto nuevas posibilidades en el ámbito de la salud mental infantil (U.S. National Library of Medicine, 2023).

El uso de algoritmos de clasificación aplicados al análisis de datos clínicos, conductuales o sensoriales ha mostrado resultados prometedores en la identificación temprana de patrones compatibles con el TDAH y otros trastornos del neurodesarrollo. Esta línea de investigación, que combina la informática y la neuropsicología, fundamenta el enfoque adoptado en el presente trabajo, orientado a explorar la aplicabilidad de modelos de *ML* en la detección de este tipo de condiciones.

2.1.2. TDAH: descripción general e infancia

El TDAH es una condición del neurodesarrollo que comienza a manifestarse en etapas tempranas del desarrollo de los menores, durante la infancia comúnmente, y que puede perdurar a lo largo de la adolescencia e incluso hasta la adultez. Se suele manifestar principalmente a través de síntomas continuados de inatención, hiperactividad y/o impulsividad, que afectan de manera notable al rendimiento académico del menor, así como a sus relaciones sociales y a su comportamiento general (U.S. National Library of Medicine, 2023).

El TDAH impacta en múltiples ámbitos del desarrollo infantil:

- En el entorno escolar, los niños pueden tener dificultad para seguir las instrucciones y normas de convivencia del aula, mantener la concentración en tareas que requieren mayor dedicación y organizar actividades de forma adecuada.
- En el ámbito familiar y social, son comunes los problemas del menor para esperar los turnos, controlar sus impulsos o comportarse de manera adecuada según el contexto.
- Estos casos, sumados en conjunto, pueden generar en el menor conflictos interpersonales y sentimientos de frustración al no poder lograr sus objetivos académicos como el resto de compañeros, o baja autoestima derivada de lo anterior.

Clasificación y presentaciones clínicas

Según el *Diagnostic and Statistical Manual of Mental Disorders* (DSM-5), se identifican 3 tipos principales de formas de presentación del TDAH:

- **TDAH predominantemente inatento:** Suelen tener problemas de atención y se distraen con facilidad. Les cuesta organizarse o terminar las tareas. Pueden tener problemas para seguir conversaciones o instrucciones.

- **TDAH predominantemente hiperactivo e impulsivo:** Podemos distinguir 2 casos:
 - **Con hiperactividad:** Sienten necesidad de estar siempre en movimiento. Tienen problemas para quedarse quietos y pueden ser inquietos o hablar demasiado.
 - **Con impulsividad:** Presentan problemas para controlar sus acciones y palabras, sin pensar en las consecuencias, de forma repentina.
- **TDAH combinado:** Es el tipo de TDAH más extendido entre la población infantil y adolescente; combina ambas.

Causas del TDAH

Se desconoce actualmente la causa exacta que origina el TDAH. Tras leer y consultar artículos de expertos en la materia y el artículo sobre el que se basa el presente trabajo sobre *ML* y TDAH, probablemente se deba al resultado de una combinación de factores, como la genética (del lado de la madre) y el entorno socioeconómico. Investigaciones están analizando posibles factores ambientales que podrían aumentar el riesgo de TDAH: lesiones cerebrales, nutrición y entornos sociales.

Tras realizar el estudio de *ML*, basándonos en el conocimiento del autor del artículo sobre el que se basa el presente trabajo, se ha podido llegar a una serie de factores comunes que pueden influir en el desarrollo del TDAH:

1. Factores genéticos

El porcentaje de heredar el TDAH de algún familiar directo se sitúa entre el 70 % y el 80 %. Hay evidencia de que la presencia de un progenitor o un hermano con TDAH multiplica el riesgo en otros miembros de la familia. Por otro lado, la genética molecular (polimorfismos y genes involucrados en neurotransmisión y desarrollo cerebral) opera sobre los circuitos cerebrales relacionados con la atención, la recompensa y funciones ejecutivas (Faraone & Larsson, 2019).

2. Factores ambientales prenatales y perinatales

Aunque la genética explica la mayor parte del riesgo, entre el 10 % y el 40 % de la variación puede atribuirse a factores ambientales. Entre ellos, la exposición prenatal a sustancias y toxinas (fumar, consumir alcohol, drogas o ciertas medicaciones durante el embarazo) así como sustancias contaminantes del aire se refleja en alteraciones neuroconductuales relacionadas con el déficit de atención. Del mismo modo, se han vinculado otros riesgos prenatales como una dieta materna deficiente (rica en grasas, carbohidratos refinados, baja en nutrientes), nacer antes de término o con bajo peso, etc.

3. Salud materna

Las enfermedades maternas que existen antes o durante el embarazo (como la obesidad, trastornos psiquiátricos) y una respuesta inmune alterada se han vinculado con un mayor riesgo de TDAH en el menor.

Aunque hay autores que sostienen la importancia de este enfoque, otros han descubierto que la situación es más compleja de lo previamente considerado. Otros estudios (Rivera, 2016) sugieren que el TDAH se atribuye a una acumulación de factores más que a una causa única. En concreto, el mencionado artículo de F. B. Rivera sostiene que la elevada presencia actual del TDAH no se debe a un solo factor, sino a una combinación de causas. Entre las principales razones identificadas se encuentran, entre otros:

- Disparidad de criterios y herramientas.
- Discrepancia entre informantes.

- Diferencias entre pruebas diagnósticas.
- Cambios en el DSM-5.
- Edad de inicio de los síntomas.

Con lo cual, esto sirve para reforzar el propósito del siguiente proyecto de desarrollar una herramienta que ayude al diagnóstico y que permita reducir el número de *test* y herramientas psicopedagógicas al mínimo, es decir, aunando la tecnología y avances que proporciona el *ML* con el criterio médico profesional, para evitar confusiones y errores en el diagnóstico.

Sintomatología y diagnóstico del TDAH

La sintomatología del TDAH se agrupa principalmente en 2 tipos principales: **déficit de atención** e **hiperactividad-impulsividad**. Los afectados pueden presentar síntomas de una de estas dimensiones o de ambas, lo que da lugar a 3 tipos de presentaciones clínicas del TDAH: presentación predominante de falta de atención, presentación predominante con hiperactividad-impulsividad y presentación combinada (U.S. National Library of Medicine, 2023).

Síntomas de inatención

Las personas con síntomas de inatención suelen mostrar, de manera frecuente, los siguientes comportamientos:

- Dificultad para mantener la atención en tareas o actividades.
- Tendencia a cometer errores por descuido, al no prestar atención a los detalles.
- Aparente falta de escucha cuando se les habla directamente.
- Dificultad para seguir instrucciones o terminar tareas escolares o laborales.
- Problemas para organizar actividades y gestionar el tiempo.
- Evitación o resistencia a tareas que requieren esfuerzo mental sostenido.
- Distraibilidad por estímulos irrelevantes.
- Olvidos frecuentes en actividades cotidianas.

Síntomas de hiperactividad e impulsividad

En esta dimensión, los síntomas más comunes incluyen:

- Inquietud motora (mover constantemente las manos o los pies, levantarse en situaciones donde se espera que estén sentados).
- Correr o trepar en situaciones inapropiadas (en niños) o sentirse internamente inquieto (en adolescentes y adultos).
- Dificultad para permanecer en un lugar, en situaciones que requieren quietud.
- Dificultad para involucrarse en actividades tranquilas.
- Dificultad para esperar el turno en actividades grupales o juegos.
- Interrumpir e inmiscuirse constantemente en conversaciones ajenas.

Diagnóstico del TDAH

El diagnóstico del TDAH debe ser realizado por un profesional de la salud mental (psicólogo clínico o médico psiquiatra) a través de un proceso integral que incluye: entrevistas clínicas, observación directa, cuestionarios estandarizados (BRIEF, Conners) y los antecedentes del paciente (cita). No existe una prueba “única” que confirme el diagnóstico de TDAH. Por ello, se recomienda realizar inicialmente una evaluación física para descartar posibles causas médicas que puedan generar sintomatología similar, como: problemas de audición, visión, trastornos del sueño, cuadros de ansiedad, depresión o dificultades específicas del aprendizaje.

De entre los instrumentos clínicos utilizados por los psicólogos, destacan las escalas de evaluación conductual (como la Conners Rating Scale) completadas por padres, maestros y, en ocasiones, el propio paciente. Estas herramientas sirven para ayudar a medir la frecuencia y la severidad de los síntomas y a determinar si cumplen con los criterios diagnósticos establecidos por el *DSM-5*.

Los criterios diagnósticos del *DSM-5* establecen que para diagnosticar TDAH deben cumplirse las siguientes condiciones (Association, 2013):

- Presencia de varios síntomas antes de los 12 años.
- Presencia de al menos seis síntomas de inatención y/o hiperactividad-impulsividad mantenidos por al menos seis meses.
- Los síntomas deben presentarse en dos o más contextos (hogar, escuela, trabajo).
- Los síntomas deben interferir de manera notable en el funcionamiento académico, social o laboral.

Test de Conners

Una de las herramientas más utilizadas para la evaluación del TDAH en adolescentes y adultos es el *Conners' Continuous Performance Test II* (CPT-II). Se trata de una prueba informatizada diseñada para medir variables relacionadas con la atención sostenida, la impulsividad y el control inhibitorio (Conners, 2000). Este *test* forma parte del conjunto de herramientas psicométricas desarrolladas por Keith Conners, ampliamente empleadas en contextos clínicos y de investigación para apoyar el diagnóstico del TDAH.

Durante la prueba del *test* de CPT-II probada, el *test* consistía en responder a una secuencia de estímulos visuales durante 15 minutos (normalmente letras) que aparecen de forma rápida y secuencial en la pantalla de un ordenador. El *test* que fue probado fue el *Continuous Performance Test - Short version*, de la página de Millisecond (Millisecond Software, 2025), donde por medio de la instalación de un software dedicado (Inquisit, by Millisecond), se instalaba una demo del *test* CPT corta, la cual consistía en 2 tipos de tareas:

- En la primera fase, se le solicita al sujeto que presione una tecla cada vez que aparezca una letra determinada, evaluando así la capacidad de respuesta sostenida.
- En la segunda fase (más compleja), el participante debe responder únicamente cuando aparezca una letra específica precedida de otra (por ejemplo, una “X” que va seguida de una “A”). Así se evalúa el control y la vigilancia.

Este *test* ha sido empleado en multitud de estudios clínicos y también se ha utilizado en programas como HYPERAKTIV, que se centran en la investigación del TDAH. Aunque se recomienda su uso complementario junto a otras fuentes de información clínica para obtener un diagnóstico más preciso.

Evolución y tendencia del TDAH

El TDAH es un trastorno que ha ido aumentando su presencia en la población mundial. En las últimas dos décadas ha habido un incremento continuado en los diagnósticos. Por ejemplo, las encuestas nacionales de EEUU (como el caso de NSCH) reflejan un aumento de la prevalencia de TDAH del 6.1 % al 10.2 % desde 1997 hasta 2019. En el caso de Estados Unidos, el porcentaje es el mayor del mundo, donde la presencia de TDAH actualmente oscila en torno al 5.8 % (unos 5.8 millones de pacientes pediátricos). Aunque el TDAH desde siempre ha sido considerado un trastorno de la infancia, en torno al 90 % de los niños con TDAH continúan presentando síntomas en la etapa adulta (Abdelnour et al., 2022).

Este trastorno neurobiológico afecta en torno al 2–12 % de la población pediátrica mundial. En Europa, afecta a 1 de cada 20 niños o adolescentes en edad escolar (Polanczyk et al., 2007). Mientras que en España, se ha estimado que tiene una prevalencia global del 6,8 %, siendo mayor en varones que en mujeres (Catalá-López et al., 2012).

Otros trastornos relevantes en el contexto del trabajo

Aunque este trabajo se centra principalmente en el TDAH, existen otros trastornos del neurodesarrollo que resultan especialmente relevantes en el análisis del comportamiento infantil. La identificación precisa de estos trastornos resulta esencial en entornos clínicos y educativos, donde un diagnóstico erróneo o incompleto puede dificultar una intervención adecuada.

A continuación, se describen brevemente algunos de los trastornos más relevantes en este contexto:

1. *Trastorno del Espectro Autista (TEA)*

El TEA se caracteriza por dificultades constantes en la comunicación social y en la interacción, así como por patrones de comportamiento, intereses o actividades restrictivos y/o repetitivos. En el ámbito infantil, el TEA se manifiesta desde edades muy tempranas y, en muchos casos, presenta solapamiento con síntomas del TDAH (p. ej., dificultades en la atención o impulsividad), lo que puede confundir con TDAH en las primeras etapas del desarrollo.

2. *Trastornos del aprendizaje*

Dificultades específicas en habilidades académicas como la *dislexia*, la *disgrafía* o la *discalculia*. No afectan de forma directa a la inteligencia del menor, pero pueden generar importantes dificultades escolares y emocionales. Es común que los niños con TDAH presenten también trastornos del aprendizaje.

3. *Trastorno del desarrollo de la coordinación (TDC)*

También *dispraxia*. Afecta a la ejecución de movimientos motores, impacta en habilidades motrices y suele pasar desapercibido o confundirse con “torpeza infantil”. En ocasiones coexiste con TDAH o TEA.

4. *Trastornos de la comunicación*

Dificultades en el lenguaje (expresivo, habla, TEL) que interfieren en el desarrollo social y académico. Su coexistencia con TDAH o TEA es habitual.

5. *Trastorno de ansiedad infantil*

La ansiedad en la infancia puede tener una fuerte repercusión en el comportamiento observable. Inquietud, nerviosismo o evitación pueden confundirse con hiperactividad o déficit de atención.

2.2. Fundamentos de *Machine Learning* para el análisis de datos

El aprendizaje automático o *ML* es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos capaces de aprender a partir de los datos y realizar predicciones o tomar decisiones sin ser programados para cada tarea específica.

Mediante el entrenamiento con datos históricos o de ejemplo, estos algoritmos detectan **patrones, relaciones y estructuras** subyacentes en los datos, utilizables para extraer información relevante sobre nuevas observaciones (Mitchell, 1997).

En esencia, es la ciencia de programar máquinas para que aprendan con el dato. Un ejemplo cotidiano de *ML* es el filtro de spam de correos electrónicos, que aprende a distinguir entre mensajes legítimos y no deseados a partir de **ejemplos previos**.

En el contexto de este Trabajo de Fin de Grado (detección de patrones asociados al TDAH en la infancia), *ML* proporciona una vía potente para identificar relaciones complejas entre variables, muchas veces imperceptibles a través de métodos estadísticos tradicionales.

Fundamentos del aprendizaje automático

Los conceptos clave son:

- *Conjunto de entrenamiento (training set)*: conjunto de ejemplos que el sistema utiliza para aprender. Cada ejemplo o muestra recibe el nombre de **instancia de entrenamiento**.
- *Modelo*: parte del sistema que ha aprendido a partir de los datos y es capaz de realizar predicciones. Ejemplos: redes neuronales artificiales, árboles de decisión o *Random Forests*.
- *Medidas de rendimiento*: se utilizan para evaluar la calidad del modelo; además de *accuracy*, en contextos clínicos son centrales sensibilidad, especificidad y AUC.

El proceso general en *ML* sigue un enfoque estructurado que incluye: recolección de datos, selección de características relevantes, EDA (Análisis Exploratorio de Datos), entrenamiento del modelo, validación mediante métricas seleccionadas y finalmente despliegue y aplicación práctica.

Tipos de sistemas de *Machine Learning*

Los sistemas de *ML* pueden clasificarse desde distintas perspectivas (Géron, 2019):

Según la supervisión del entrenamiento

▪ *Aprendizaje supervisado*

Algoritmos entrenados con datos etiquetados (variables de entrada + salida esperada). Las dos tareas principales son **clasificación** y **regresión**. Para ilustrarlo de forma visual, en la Figura [2.1] se muestra un esquema típico de aprendizaje supervisado, donde a partir de un conjunto de características y sus etiquetas se entrena un modelo que, una vez ajustado, puede predecir la salida para nuevos datos.

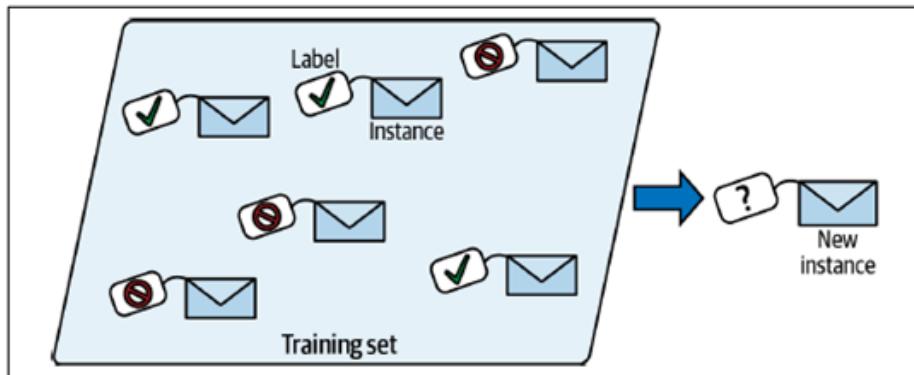


Figure 1-5. A labeled training set for spam classification (an example of supervised learning)

Figura 2.1: Ejemplo de aprendizaje supervisado (clasificación). Adaptado de Géron, 2019.

Clasificación y regresión en *Machine Learning*

1. Clasificación

El objetivo es predecir una categoría o clase. Son **valores discretos** (conjunto finito de posibles resultados). Ej.: diagnóstico positivo/negativo. Los algoritmos aprenden patrones que permiten asignar correctamente una clase a nuevas entradas.

2. Regresión

En la regresión (véase Figura [2.2]) se predicen **valores continuos**. Ej.: precio de vivienda en función de tamaño, ubicación, etc. El objetivo es modelar la relación entre variables independientes y la variable dependiente (valor a predecir).

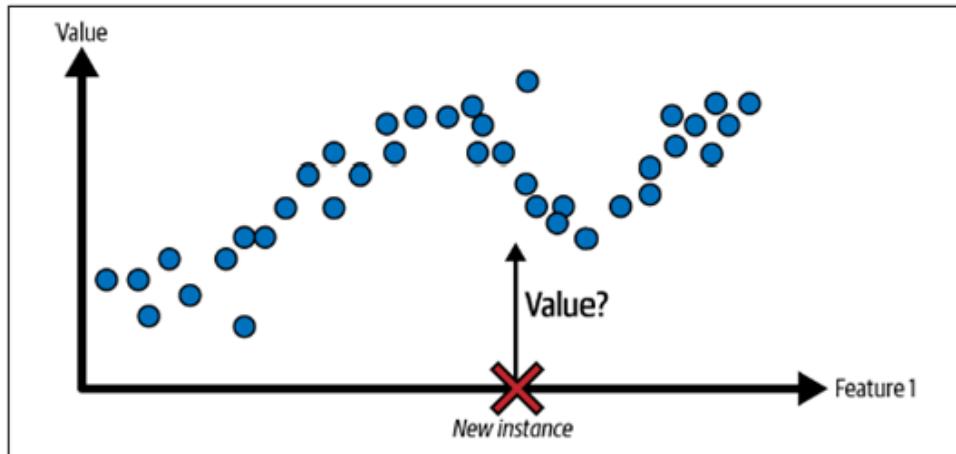


Figure 1-6. A regression problem: predict a value, given an input feature (there are usually multiple input features, and sometimes multiple output values)

Figura 2.2: Regresión Supervisada. Adaptado de Starmer, 2022.

- *Aprendizaje no supervisado*

Entrenado con datos no etiquetados: el modelo identifica **patrones inherentes**. Tareas típicas: *clustering*, reducción de dimensionalidad, detección de anomalías y reglas de asociación. Ejemplo de flujo en Figura [2.3].

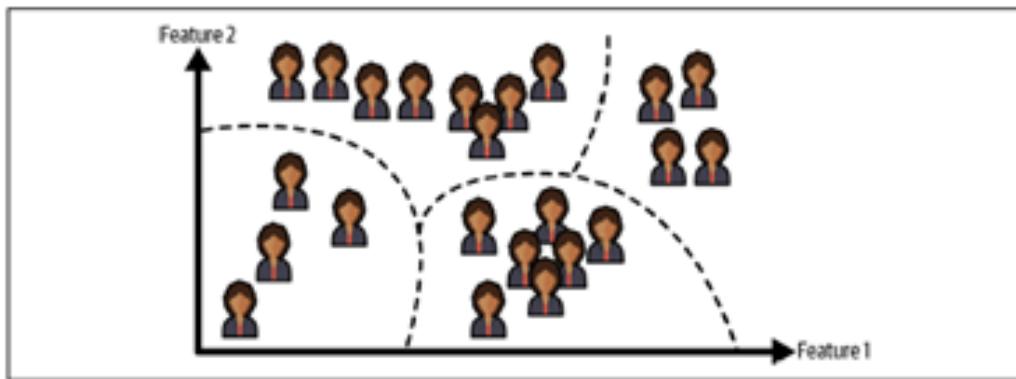


Figura 2.3: Flujo de trabajo típico en aprendizaje supervisado. Adaptado de Géron, 2019.

- *Aprendizaje semi-supervisado*

Combina una pequeña cantidad de datos etiquetados con una gran cantidad de datos no etiquetados, aprovechando ambos para mejorar el aprendizaje.

Según la forma en que aprenden los modelos

Si atendemos a la forma en la que los modelos aprenden de los datos, distinguimos:

- *Aprendizaje por lotes (batch learning)*: se entrena con el conjunto completo; se reentrena al disponer de nuevos lotes.
- *Aprendizaje en línea (online learning)*: el modelo actualiza parámetros continuamente a medida que recibe nuevos datos.

Según el enfoque de aprendizaje

- *Aprendizaje basado en instancias (instance-based)*: almacena ejemplos y predice por comparación directa (p. ej., *k-NN*).
- *Aprendizaje basado en modelos (model-based)*: construye una representación del modelo de los datos (regresión lineal, redes neuronales, árboles de decisión).

Principales desafíos en el aprendizaje automático

Los sistemas de *ML* enfrentan diversos retos técnicos y metodológicos que, si no se abordan, comprometen rendimiento y generalización (Domingos, 2012).

Uno de los más comunes es la **escasez de datos de entrenamiento**. Se requieren ejemplos suficientes para extraer patrones útiles y generalizables; con datos limitados, el modelo puede no aprender correctamente y fallar ante nuevos casos.

Otro reto es la **representatividad de los datos**. Una muestra amplia pero poco representativa induce *sample bias*.

La **calidad de los datos** es determinante: errores, atípicos o incompletos introducen ruido; por ello la limpieza y el preprocesamiento son críticos.

La **relevancia de las características** también importa: si las variables no contienen información útil o están altamente correlacionadas, el rendimiento se resiente. La **ingeniería de características** construye representaciones más informativas.

En entrenamiento pueden surgir **sobreajuste** (*overfitting*, memoriza ruido) y **subajuste** (*underfitting*, modelo demasiado simple). Ajustar complejidad, seleccionar variables y validar adecuadamente es clave.

2.2.1. Validación cruzada

La **validación cruzada** (CV) es un procedimiento de remuestreo para estimar el error de generalización y guiar la selección de hiperparámetros sin usar el conjunto de *test*. Reduce la varianza de depender de una única partición de entrenamiento/validación.

Concepto

Se divide el conjunto de entrenamiento en k pliegues. Para cada pliegue f , se entrena con los $k-1$ restantes y se evalúa en f ; la métrica (p. ej., ROC AUC, sensibilidad) se promedia sobre los k pliegues.

Buenas prácticas

- Ajustar imputación, escalado y codificación *solo* con los datos de entrenamiento de cada pliegue (evita **fuga de información**).
- Usar ***Stratified k-fold*** en datos desbalanceados.
- Fijar hiperparámetros maximizando la métrica media de CV y reentrenar con todo el entrenamiento al final.

2.2.2. Métricas de evaluación para clasificación desbalanceada

La matriz de confusión se muestra en la **Figura 2.4**. Las definiciones y fórmulas de *accuracy*, *precision*, *recall* y *specificity* pueden consultarse en las Ecuaciones (2.1). La definición de la métrica F1 está recogida en la Figura 2.5.

En problemas desbalanceados (como TDAH), la *accuracy* puede resultar engañosa. Se priorizan métricas sensibles a la clase positiva (TDAH) y medidas independientes del umbral.

		Pred. 0	Pred. 1
Real 0	TN	FP	
	FN	TP	

Figura 2.4: Matriz de confusión binaria. TP: verdaderos positivos, TN: verdaderos negativos, FP: falsos positivos, FN: falsos negativos.

Ecuaciones 2.1 - Definiciones de las métricas básicas.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1a)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2.1b)$$

$$\text{Recall (Sens.)} = \frac{TP}{TP + FN} \quad (2.1c)$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (2.1d)$$

$$F1 = 2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Figura 2.5: Definición de la puntuación F1 como media armónica entre *precision* y *recall*. Útil en escenarios desbalanceados al equilibrar ambas dimensiones.

AUC-ROC: definición, usos e interpretación

La curva **ROC** (*Receiver Operating Characteristic*) representa, para todos los umbrales de decisión, la **Sensibilidad/TPR** en el eje Y frente a la **tasa de falsos positivos/FPR** en el eje X, tal y como muestra la Ecuación 2.2.

Ecuaciones — Curva ROC y AUC.

$$\text{TPR} = \frac{TP}{TP + FN} \quad (2.2a)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (2.2b)$$

La ROC traza TPR frente a FPR al variar el umbral de decisión. El AUC $\in [0.5, 1]$ resume la capacidad discriminativa del modelo de manera independiente del umbral.

El **AUC-ROC** es el área *bajo la curva*, con valores en $[0.5, 1]$: 0.5 equivale a azar, 1.0 a separación perfecta, dicha área se puede ver de manera más gráfica en la Figura 2.6. De forma equivalente, AUC-ROC es la **probabilidad** de que el modelo asigne una puntuación mayor a un positivo que a un negativo elegidos al azar (relación con el test U de Mann–Whitney).

Resume la **capacidad discriminativa global** del modelo *independientemente del umbral* y es **invariante ante transformaciones monótonas** de la puntuación (p. ej., aplicar una sigmoidal o escalar no cambia el AUC). Es útil cuando interesa el **poder de ranking** general y cuando la prevalencia no es extrema.

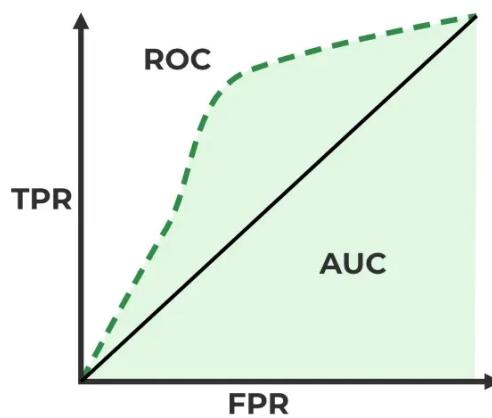


Figura 2.6: Curva ROC-AUC. Adaptado de “AUC ROC Curve in Machine Learning”, 2025.

Cómo interpretarla.

- *Global vs. local:* AUC-ROC promedia el rendimiento en *todos* los umbrales. Un AUC alto no garantiza buen desempeño en el **punto operativo** de interés; conviene inspeccionar la curva y el umbral elegido.
- *Zonas de operación:* si sólo toleras FPR muy bajos, es más informativo reportar **partial AUC** (pAUC) en el rango $FPR \in [0, \varepsilon]$.
- *Curvas iso-coste:* en ROC es posible trazar líneas de igual coste (según costes de FP/FN y prevalencia) para elegir el umbral donde la curva toca la iso-línea deseada.

Limitaciones en clases desbalanceadas. Con clases muy desbalanceadas, FPR puede parecer “pequeña” pese a muchos FP (el denominador $TN+FP$ es grande). Por ello, AUC-ROC puede resultar **optimista** en positivos muy raros. En esos casos conviene complementar con **Precision–Recall** (Subsección 2.2.2).

Curva Precision–Recall (PR): definición, usos e interpretación

La curva **Precision–Recall** traza, al variar el umbral, la **Precisión (PPV)** en el eje Y frente al **Recall (TPR)** en el eje X.

Su área (**AUC-PR** o *Average Precision*) resume el rendimiento priorizando la clase positiva, tal y como se puede apreciar de una manera gráfica en la Figura 2.7.

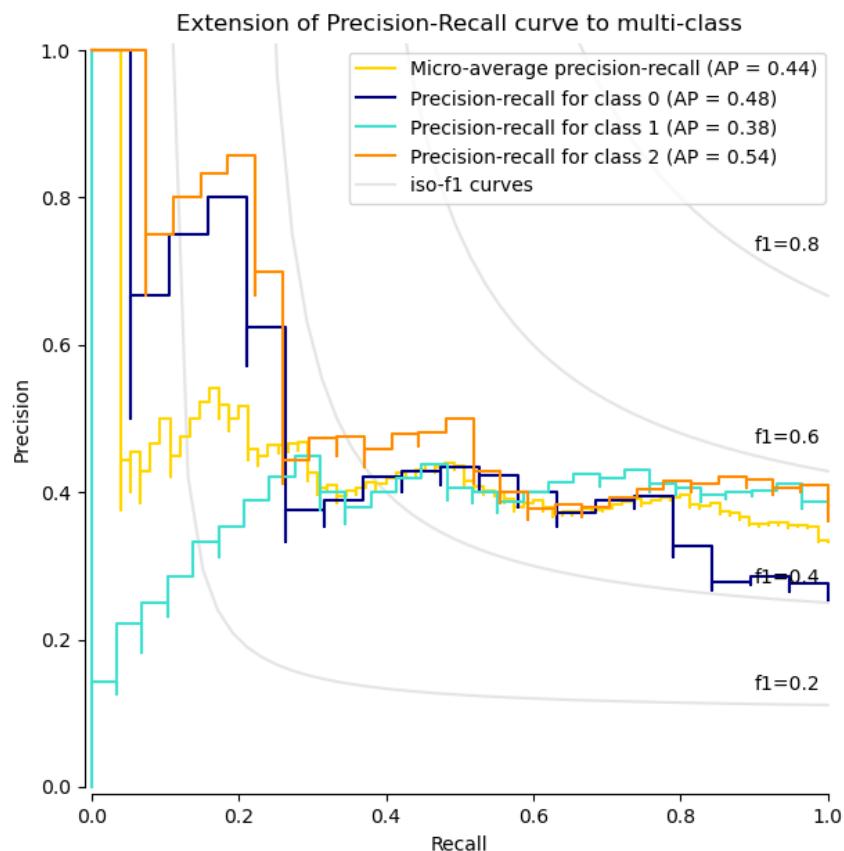


Figura 2.7: Curva *Precision-Recall*. Adaptado de “Plot Precision–Recall Curve”, s.f.

Es especialmente adecuada en **escenarios desbalanceados** y cuando importa el **valor predictivo** de las alertas positivas (minimizar falsos positivos). En este trabajo, la clase positiva (TDAH) es minoritaria y las derivaciones innecesarias tienen coste, por lo que **PR** resulta más informativa que ROC para seleccionar el **umbral operativo**.

Interpretación

- *Base-line dependiente de la prevalencia*: a diferencia de ROC, la **Línea base** de PR no es fija; depende de la prevalencia $\pi = P(Y=1)$ (ver Ecuación 2.3). De hecho, lo que explica por qué, con π pequeña, mantener alta Precisión exige FPR extremadamente baja.

$$\text{Precision} = \frac{\pi \cdot \text{TPR}}{\pi \cdot \text{TPR} + (1 - \pi) \cdot \text{FPR}} \quad (2.3)$$

- *Selección de umbral:* escoger el punto de la curva con mejor **F1** (media armónica de Precisión y Recall) equilibra detección y pureza de positivos. Alternativamente, puede fijarse una **Precisión mínima** y elegir el máximo Recall que la cumpla (criterio de negocio/clínico).
- *Lectura operativa:* moverse a la derecha (más Recall) suele bajar la Precisión; moverse arriba (más Precisión) suele bajar el Recall. La forma de la curva ilustra ese compromiso.

AUC-PR (Average Precision). Resume la curva ponderando más las regiones con mayor densidad de *recalls* observados. Útil para comparar modelos cuando la **clase positiva es rara**. No es directamente comparable entre datasets con distintas prevalencias.

Cuándo reportarla.

- Como **métrica principal** cuando la clase positiva es minoritaria y te importa la **Precisión** de las alarmas (este caso).
- Para **elegir el umbral** (p.ej., maximizando F1 o imponiendo una Precisión mínima clínica).

Relación con este trabajo. En el caso TDAH (clase positiva $\sim 11\%$), se usa la **curva PR** para seleccionar el umbral (maximizando **F1** de TDAH), mientras que **AUC-ROC** se reporta como **capacidad discriminativa global** independiente de umbral. Así se alinea la evaluación con el objetivo de **detectar casos positivos** sin inflar derivaciones por falsos positivos.

2.3. Algoritmos de clasificación aplicados a problemas de comportamiento

En problemas de conducta infantil/adolescente (datos heterogéneos, relaciones no lineales, mezcla de variables clínicas, psicoeducativas y contextuales), es útil emplear clasificadores que:

- Modelen casos de no linealidades e interacciones.
- Sean capaces de tolerar ruido y un desbalance moderado en el *dataset*.
- Ofrezcan, cuando sea posible, *explicabilidad*.

A continuación se resumen tres enfoques clásicos y eficaces.

Árbol de Decisión (*Decision Tree*)

Este tipo de modelo de clasificación particiona recursivamente el espacio de variables mediante reglas del tipo *if-then* para maximizar la pureza (p. ej., Gini, entropía). Este modelo es **clave** en su aportación a la detección de patrones de conducta, debido a los siguientes motivos:

- **Interpretabilidad directa:** las reglas son comprensibles (útiles para comunicación clínica/educativa).

- Maneja **variables mixtas** (categóricas/numéricas) y **valores faltantes** con estrategias simples.
- Detecta **interacciones** entre variables sin ingeniería manual.

En cuanto a las desventajas/riesgos de usar este modelo, hay que tener en cuenta una serie de precauciones, como el alto riesgo de **sobreajuste** si no se poda o la sensibilidad a pequeñas variaciones del dato. Es por ello que algunas **buenas prácticas** son las de limitar la profundidad del árbol, lograr el mínimo de muestras por nodo/hoja, evaluar la calibración y coste de error (p. ej., mediante un umbral de decisión).

Random Forest (RF)

Este modelo deriva del *Decision Tree*, como un conjunto (*ensemble*) de múltiples árboles entrenados sobre *bootstraps* de los datos y subconjuntos aleatorios de variables; promedia sus predicciones.

Este modelo es considerado relevante, según la literatura científica, debido a los siguientes motivos:

- **Robusto** al ruido y a variables irrelevantes; reduce el sobreajuste de un árbol único.
- **Modela no linealidades** e **interacciones** complejas sin suposición paramétrica.
- Proporciona **importancia de variables** (global) útil para priorizar indicadores conductuales/psicoeducativos.

En cuanto a las posibles desventajas/riesgos de usar este modelo como referencia, hay que tener en cuenta una serie de **precauciones**: la más notable es una **menor interpretabilidad** que un árbol único.

Por tanto, es recomendable seguir una serie de **buenas prácticas**: ajustar `n_estimators`, `max_depth`, `max_features`, balancear clases (p. ej., `class_weight`), validar con CV y revisar **calibración** (Platt/Isotónica).

Máquinas de Vectores de Soporte (SVM)

La clave de este tipo de modelo de *ML* se encuentra el hiperplano que maximiza el **margen** entre clases; con *kernels* (lineal, RBF) proyecta a espacios donde la separación es posible.

Las principales ventajas que aporta este modelo al problema de detección de patrones de comportamiento, se detallan a continuación:

- Buen rendimiento en **datasets medianos** con **señal sutil** y **no linealidades**.
- El margen grande tiende a **generalizar** bien cuando hay solapamiento sintomático.

En cuanto a las posibles desventajas/riesgos de usar este modelo como referencia, hay que tener en cuenta una serie de precauciones, como que este modelo es menos interpretable; sensible a **escalado** de variables y a hiperparámetros (C , γ).

Es por ello que se recomienda seguir una serie de **buenas prácticas**: estandarizar, **búsqueda de hiperparámetros** (grid/bayesiana), manejo del desbalance (`class_weight='balanced'`), y evaluar **AUC**, **sensibilidad** y **calibración** si se usarán probabilidades.

2.4. Preparación y calidad del dato

La preparación del dato constituye el eslabón metodológico que conecta la fuente de información con los modelos de aprendizaje automático. Un tratamiento riguroso permite reducir sesgos, evitar fugas de información y mejorar la interpretabilidad, aspectos especialmente relevantes en contextos clínicos y psicoeducativos. A continuación se presentan los principios generales que guiarán, de forma independiente al caso empírico, la selección y transformación de variables, el manejo de ausencias y la garantía de calidad.

2.4.1. Imputación univariada mediante K vecinos

La imputación de valores faltantes es crítica para preservar tamaño muestral y evitar sesgos en la estimación. En este trabajo se adopta una estrategia *univariada* basada en K vecinos (*KNN*) cuyo objetivo es estimar cada valor ausente de una variable X a partir de los K ejemplos válidos más “próximos” dentro de la propia distribución de X .

Idea general.

Para una observación i con X_i ausente, se identifica un conjunto $\mathcal{N}_K(i)$ de K observaciones con X no nulo y máxima similitud con i según un criterio simple y estable (p. ej., proximidad en un orden natural de la variable, reglas de aplicabilidad del cuestionario o un índice de vecindad definido para X). La imputación \hat{X}_i se obtiene a partir de los vecinos:

$$\hat{X}_i = \begin{cases} \text{mean}(X_j : j \in \mathcal{N}_K(i)), & \text{si } X \text{ es numérica,} \\ \text{median}(X_j : j \in \mathcal{N}_K(i)), & \text{si } X \text{ es ordinal,} \\ \text{mode}(X_j : j \in \mathcal{N}_K(i)), & \text{si } X \text{ es categórica/binaria.} \end{cases}$$

Ventajas

Frente a imputaciones constantes (media/moda global), el esquema KNN univariado preserva la estructura local de la variable, reduce la varianza de la estimación para categorías minoritarias y mantiene coherencia con los mapeos originales.

Elección de K

Valores pequeños de K capturan mejor la vecindad (menor sesgo, mayor varianza), mientras que K más grandes suavizan la imputación (mayor sesgo, menor varianza). Se recomienda validar K en función de la densidad de valores no nulos y de la naturaleza de X .

Buenas prácticas

1. Respetar el orden en variables ordinales.
2. No imputar *logical skips* como si fueran ausencias reales.
3. Documentar el diccionario de mapeos y verificar que la imputación no cree categorías inexistentes.
4. Realizar la imputación dentro del bucle de validación para evitar fuga de información cuando proceda.

Limitaciones

En mecanismos MNAR (no al azar) la imputación puede estar sesgada; además, si X carece de un criterio de vecindad significativo, el método se aproxima a una imputación por resumen local y debe interpretarse con cautela.

2.5. Fundamentos de análisis exploratorio y asociaciones

Esta sección resume los criterios estadísticos utilizados para evaluar asociaciones entre la variable objetivo binaria y los distintos tipos de predictores durante el EDA (véase Capítulo 6, Sección 6.3.4).

2.5.1. Correlación biserial puntual (binaria–continua)

La correlación biserial puntual (r_{pb}) cuantifica la asociación lineal entre una variable dicotómica (0/1) y una continua. Es análoga a Pearson cuando una de las variables es binaria. Sus valores oscilan en $[-1, 1]$: magnitud mayor implica asociación más fuerte; el signo indica dirección.

Fórmula.

$$r_{pb} = \frac{\bar{Y}_1 - \bar{Y}_0}{s_Y} \sqrt{pq}$$

donde $p = P(X=1)$, $q = 1 - p$,

\bar{Y}_1 = media de $Y \mid X=1$, \bar{Y}_0 = media de $Y \mid X=0$,

s_Y = desviación estándar de Y .

Contraste de significación.

$$t = r_{pb} \sqrt{\frac{n-2}{1-r_{pb}^2}} \quad (\text{con } n-2 \text{ g.l.})$$

Supuestos e interpretación. Requiere relación aproximadamente lineal y distribución unimodal de Y por grupos. Es útil para valorar si la probabilidad del resultado cambia con el nivel de Y .

2.5.2. Correlación de Spearman (ordinal/binaria–ordinal)

El coeficiente de Spearman (ρ_s) mide la asociación *monótona* entre dos variables a partir de sus rangos, sin exigir linealidad ni normalidad. Es apropiado para variables ordinales y también para binarias cuando se desea un tratamiento uniforme en conjuntos mixtos.

Fórmulas. Si $R(X_i)$ y $R(Y_i)$ son los rangos (con manejo de empates), entonces

$$\rho_s = \frac{\text{cov}(R(X), R(Y))}{s_{R(X)} s_{R(Y)}}.$$

En el caso sin empates, usando $d_i = R(X_i) - R(Y_i)$ y n observaciones,

$$\rho_s = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)}.$$

Significación. Para n moderado/grande puede usarse una aproximación t :

$$t = \rho_s \sqrt{\frac{n-2}{1-\rho_s^2}},$$

o bien pruebas exactas/permuciones. $\rho_s \in [-1, 1]$; mayor magnitud implica relación monótona más fuerte.

2.5.3. Prueba Chi-cuadrado de independencia (categórica–binaria)

La prueba χ^2 contrasta si dos variables categóricas son independientes comparando frecuencias observadas vs. esperadas en una tabla de contingencia.

Esquema visual (tabla de contingencia).

	Clase 0	Clase 1	Total fila
Cat. 1	O_{11}	O_{12}	$O_{1\cdot}$
Cat. 2	O_{21}	O_{22}	$O_{2\cdot}$
⋮	⋮	⋮	⋮
Cat. r	O_{r1}	O_{r2}	$O_{r\cdot}$
Total col.	$O_{\cdot 1}$	$O_{\cdot 2}$	N

Bajo independencia, $E_{ij} = \frac{O_{i\cdot} O_{\cdot j}}{N}$.

Fórmula del estadístico.

$$\chi^2 = \sum_{i=1}^r \sum_{j=1}^c \frac{(O_{ij} - E_{ij})^2}{E_{ij}}, \quad \text{con g.l. } (r-1)(c-1).$$

Tamaño de efecto (recomendado). Para cuantificar fuerza (no solo significación), usa V de Cramér:

$$V = \sqrt{\frac{\chi^2}{N \min(r-1, c-1)}}, \quad 0 \leq V \leq 1.$$

Supuestos e interpretación. Se recomienda que la mayoría de celdas tenga $E_{ij} \geq 5$. Un $p < 0,05$ indica asociación estadísticamente significativa; V aporta la magnitud de dicha asociación.

2.6. Normalización

La *normalización* abarca las transformaciones que hacen que los datos sean compatibles con los supuestos y la geometría de los algoritmos: convertir categorías en representaciones numéricas y homogeneizar escalas entre variables.

Los objetivos de la normalización pasan por evitar introducir orden ficticio en atributos nominales, mejorar la estabilidad numérica y la velocidad de convergencia en modelos basados en distancias u optimización, y por último, reducir efectos espurios debidos a diferencias de magnitud entre características.

En este trabajo consideramos dos familias complementarias: transformaciones de **representación** para variables categóricas, donde empleamos *Codificación One-Hot* (Sección 2.6.1) para preservar la naturaleza nominal sin imponer orden, y transformaciones de **escala** para variables numéricas, donde aplicamos *Estandarización (Z-score)* (Sección 2.6.2) para centrar en media cero y varianza unitaria.

En la práctica, estas operaciones se insertan en el *pipeline* de preprocesado tras la imputación y antes del aprendizaje del modelo. A continuación se describen brevemente ambas técnicas y consideraciones de uso.

2.6.1. Codificación *One-Hot* (variables categóricas)

Las variables categóricas nominales no poseen orden intrínseco; asignarles números enteros induce un orden ficticio. La **codificación One-Hot** (véase Figura 2.8) representa cada categoría c_k de una variable C mediante un vector indicador. Para evitar colinealidad perfecta en modelos lineales, se elimina una columna de referencia (*drop-first*). *Ventajas:* permite a modelos que requieren entrada numérica manejar nominales sin imponer orden. *Precauciones:* con alta cardinalidad, el número de columnas crece y puede introducir dispersión; alternativas incluyen agrupación de categorías raras o técnicas específicas (p.ej., *target/impact encoding*, con validación cuidadosa para evitar fuga de información).

$$\mathbf{e}(C = c_k) = (I[C = c_1], \dots, I[C = c_K])$$

Figura 2.8: Representación *One-Hot* de una categoría c_k de la variable C mediante vector indicador.

2.6.2. Estandarización (normalización *Z-score*)

La **estandarización** transforma una variable X a (véase Figura 2.9) una representación con media 0 y desviación estándar 1. Es especialmente útil para algoritmos sensibles a la escala (SVM, KNN, MLP). *Buenas prácticas:* ajustar (μ, σ) solo en el conjunto de entrenamiento y aplicar la transformación a validación/prueba (evita fuga de información); no es necesaria para árboles y bosques aleatorios. En variables binarias, el centrado/escala desplaza las categorías poco frecuentes a valores más extremos; debe tenerse en cuenta en la interpretación de distancias.

$$Z = \frac{X - \mu}{\sigma}, \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \mu)^2}$$

Figura 2.9: Estandarización *Z-score* para homogeneizar escalas de entrada.

Capítulo 3

Estado del Arte

3.1. Introducción

En este capítulo se presenta el **Estado del Arte** sobre la detección y evaluación del TDAH en población infantil-escolar desde dos tradiciones complementarias: la **clínico-psicométrica** y la **computacional basada en ML**. Se partirá de un **marco contextual** que sintetiza la prevalencia, los desafíos diagnósticos y las implicaciones socioeducativas del trastorno, para situar el problema y las necesidades de objetivación y apoyo a la decisión.

A continuación, se explicitan los objetivos de la revisión:

- Familiarizar al lector con los enfoques predominantes aplicados al TDAH.
- Sintetizar críticamente trabajos representativos resaltando su valor metodológico y tecnológico.
- Identificar lagunas donde una propuesta de ML con foco conductual/psicoeducativo pueda aportar evidencia y transferibilidad.

3.1.1. Contexto TDAH

El TDAH presenta una elevada prevalencia en población infantil y continúa generando debate sobre sus causas, criterios diagnósticos y repercusiones socioeducativas. En paralelo, el auge de la ciencia de datos y el *ML* han impulsado soluciones de cribado y apoyo a la decisión clínica con potencial de objetivar síntomas y agilizar la evaluación.

3.1.2. Objetivos de la revisión

Entre los objetivos de este capítulo se encuentran:

1. Familiarizar al lector con los enfoques predominantes (clínico-psicométricos y ML) aplicados al TDAH.
2. Sintetizar trabajos representativos y su valor metodológico/tecnológico.
3. Identificar *lagunas* donde una propuesta basada en ML, con foco conductual/psicoeducativo, pueda aportar valor científico.

3.1.3. Metodología Revisión del Estado del Arte

Se realizó una búsqueda exploratoria en Google Scholar de estudios (2016–2024) sobre TDAH infantil y ML, priorizando artículos con:

1. Muestras y/o datasets claramente descritos.
2. Puesta en valor de métricas de rendimiento (Accuracy, Sensibilidad, Especificidad, AUC, F1-Score).
3. Transparencia de variables y pipeline de preprocesamiento de datos.

Se incluyeron además referencias sobre instrumentos psicométricos y discusión de prevalencia/-diagnóstico.

3.2. Revisión y síntesis de la literatura

Esta sección revisa la literatura sobre los métodos de diagnóstico del TDAH, contrastando los enfoques clásicos con los computacionales.

Se comienza analizando los métodos tradicionales (entrevistas, escalas) y sus limitaciones de subjetividad. A continuación, se justifica la evolución hacia el ML como una herramienta capaz de modelar patrones complejos que la estadística clásica no captura.

El núcleo del capítulo lo componen tres trabajos relacionados que se analizan en detalle, los cuales aplican ML a diferentes fuentes de datos: una encuesta poblacional (NSCH), una prueba CPT informatizada y un videojuego con sensores.

Finalmente, la sección concluye identificando las limitaciones transversales y las lagunas detectadas en la investigación actual, como la falta de generalización y explicabilidad en la práctica clínica.

3.2.1. Métodos tradicionales de análisis de datos en psicología infantil

Antes del despliegue masivo de ML, la evaluación del TDAH se apoyaba en **entrevistas clínicas, tests neuropsicológicos y escalas estandarizadas** (por ejemplo, Conners (2000), Gioia et al. (2000), Bordin et al. (2013)), con baremos poblacionales y práctica de observación estructurada. Aunque estos métodos preservan **valididad clínica e interpretabilidad**, su dependencia de informes subjetivos y la heterogeneidad en criterios (p.ej., DSM) pueden introducir sesgos y variabilidad; además, su capacidad para capturar relaciones no lineales complejas es limitada. La literatura advierte, asimismo, diferencias culturales y evolutivas en la expresión sintomática y en la aplicación de instrumentos que condicionan la estimación de prevalencia y el diagnóstico diferencial en infancia.

3.2.2. Evolución hacia Machine Learning

Los métodos estadísticos clásicos (regresión logística, análisis factorial, contraste de hipótesis) han permitido identificar asociaciones robustas con variables clínicas, familiares o socio-económicas, con *alta explicabilidad* y requisitos modestos de datos. Sin embargo, ante *datasets* multivariantes con interacciones y no linealidades, los algoritmos de ML (Decision Tree, Random Forest, SVM, MLP, XGBoost, 1D-CNN, entre otros) **mejoran la modelización de patrones**

complejos, a costa de menor transparencia. Para paliar esto, se recurre a *explicabilidad post-hoc* (p.ej., *feature importance* o importancia de las características) y a modelos intrínsecamente interpretables cuando sea viable.

3.2.3. Trabajos relacionados basados en Machine Learning

Tras un proceso de búsqueda y lectura, se identifican varios estudios representativos que aplican ML a la detección o predicción del TDAH en población infantil. Estos trabajos, si bien comparten un objetivo común, difieren significativamente en la naturaleza de los datos que emplean como punto de partida. La comparativa de sus metodologías se pueden consultar en la Tabla 3.1.

Un pilar fundamental, que sirve como base conceptual y metodológica para el presente proyecto, es el trabajo de (Maniruzzaman et al., 2022). Este estudio abordó el desafío utilizando una encuesta poblacional a gran escala, la *National Survey of Children's Health* (NSCH) de 2018-2019, sobre una muestra masiva de 45,779 menores (de los cuales un 11.4 % presentaba TDAH). Su metodología combinó la identificación de factores de riesgo mediante regresión logística con la evaluación de un amplio espectro de clasificadores (RF, NB, DT, XGBoost, KNN, SVM, MLP y 1D-CNN), gestionando el severo desbalance de clases mediante una combinación de **sobre y submuestreo**. El resultado más notable fue el del modelo **Random Forest**, que obtuvo el mejor rendimiento con una **precisión del 85.5 %**, sensibilidad del 84.4 %, especificidad del 86.4 % y un AUC de 0.94. La gran relevancia de este trabajo radica en que demuestra la utilidad de los modelos de ML sobre grandes encuestas poblacionales, logrando encontrar patrones complejos que asocian el TDAH no solo con síntomas evidentes, sino también con variables socioeconómicas y de contexto familiar (como *sexo masculino*, *ansiedad*, *depresión*, *asma*, *pobreza* o la estructura biparental).

En una línea de investigación distinta, orientada a la captura de datos más objetivos y menos dependientes de la percepción del informante, se sitúa el trabajo de Slobodin et al. (2020). En lugar de una encuesta, su enfoque se basó en los resultados de una prueba informatizada (CPT) aplicada a 458 niños (213 con TDAH). El modelo de ML se entrenó utilizando los **índices CPT** objetivos (Atención, Impulsividad, etc.) generados bajo *distractores* visuales y auditivos. Sus resultados superaron notablemente a los modelos basados únicamente en datos clínicos, alcanzando una **accuracy del 87 %**, **sensibilidad del 89 %** y **especificidad del 84 %**. Este estudio aporta evidencia clave sobre la ganancia incremental que suponen los datos de ejecución objetivos en contextos ecológicamente más realistas.

Finalmente, un enfoque aún más innovador es la captura multimodal de datos a través de *serious games*. Zakani et al. (2023) exploró esta vía validando un **videojuego** ('FishFinder') en un grupo de 52 niños. La clave de su método fue la extracción de 114 características que combinaban el **rendimiento in-game** (reflejando inatención) con los datos de los **sensores del smartphone** (acelerómetro y giroscopio) para medir objetivamente la hiperactividad motora. Mediante un modelo **SVM**, alcanzaron una notable **accuracy del 92.3 %** (con 90 % de sensibilidad y 93.7 % de especificidad). Este trabajo evidencia el gran potencial de las herramientas escalables y de bajo estrés que permiten capturar, de forma simultánea, los diferentes síntomas nucleares del TDAH (cognitivos y motores).

Tabla 3.1: Resumen comparativo de trabajos de ML en TDAH infantil.

Estudio	Muestra	Modalidad	Modelo	Rendimiento
Maniruzzaman	45.779 (3 a 17 años)	Encuesta: clínico-sociofam. /prenatal	RF, SVM, XGB, MLP, 1D-CNN	Precisión AUC
Slobodin	458 (6 a 12 años)	Índices CPT + edad/género/tiempo	ML con <i>bootstrap</i>	Precisión Sensibilidad Especificidad
Zakani	52 (5 a 12 años)	<i>Gameplay</i> + IMU (acel./gyro)	SVM	Precisión Sensibilidad Especificidad

3.2.4. Limitaciones transversales y lagunas detectadas

De la revisión crítica se observan patrones recurrentes:

- **Generalización limitada:** tamaños muestrales modestos en estudios instrumentales (CPT/-juegos) y/o sesgos de selección; en encuestas masivas, falta de *granularidad clínica*.
- **Desbalance y validación:** gestión del *imbalance* no siempre estandarizada; escasez de validaciones externas y de *reporting* completo (intervalos de confianza, calibración, costos de error).
- **Variables poco integradas:** predominio de *single-modality* (sólo CPT o sólo cuestionarios) y ausencia de integración conductual-psicoeducativa-contextual (ámbito escolar/familiar).
- **Explicabilidad:** pocos trabajos incorporan sistemáticamente análisis de *feature importance*/XAI orientados al uso clínico-educativo.
- **Transferencia a práctica:** soluciones prometedoras (p.ej., videojuegos) con *desarrollo* limitado (hardware, validación intercultural, *drift*, mantenimiento).

3.3. Conclusiones de la revisión y encaje de la propuesta

En conjunto, la evidencia apoya que ML *complementa* los métodos tradicionales: capta relaciones complejas, mejora el cribado objetivo y, con un pipeline sólido, ofrece rendimientos competitivos. Persisten, no obstante, lagunas en **integración de variables conductuales y psicoeducativas interpretables**, en **validación y calibración** con especial atención al **coste de errores** y en **explicabilidad** alineada con práctica clínica y escolar.

El trabajo de Maniruzzaman ha servido de *base conceptual y empírica* para el presente proyecto por el tipo de **dataset** utilizado: la NSCH integra un abanico amplio de *variables de cuestionario* que cubren dimensiones **conductuales, psicoeducativas y contextuales** (salud física y mental, antecedentes perinatales, comorbilidades, estructura y nivel socioeconómico del hogar, entre otras). Esa diversidad permitió extraer **patrones interpretables** con técnicas de ML (p.ej., Random Forest) y obtener **rendimientos altos** a la vez que se identificaban factores de riesgo y protectores de forma coherente con la literatura.

La presente propuesta se sitúa en dichas brechas:

- Construir un **pipeline reproducible** (imputación, EDA, codificación por tipo de variable, selección de variables finales) para **clasificación supervisada**.
- Priorizar **variables conductuales/psicoeducativas** con **explicabilidad** (árboles, importancia de variables, umbrales).
- Entregar una **aplicación web** que favorezca **transferencia y transparencia** (visualización del modelo y entorno interactivo con el usuario final).
- Evaluar **calibración y umbral** para adecuar la decisión al contexto sensible de TDAH en infancia/adolescencia.

Capítulo 4

Planificación y presupuesto

En este capítulo se fijan los pilares operativos del proyecto, estableciendo cómo se organizaría el trabajo, con qué enfoque metodológico se abordarían las tareas y qué recursos serían necesarios para su correcta ejecución.

4.1. Planificación del proyecto

En esta sección se presenta la planificación temporal inicial del proyecto, describiendo las fases clave y los recursos que se preveían para llevar a cabo la implementación de la solución propuesta, junto con un estudio presupuestario.

Asimismo, se contempló una planificación temporal mediante un diagrama de Gantt que proporcionaría una representación visual de las fases y tareas, así como su cronograma de ejecución. Este diagrama facilitaría la gestión del tiempo y el seguimiento del progreso para, posteriormente, comparar la planificación inicial con la ejecución real. Además, se incluiría un desglose de costes y el cálculo del presupuesto total necesario para la realización del proyecto.

La planificación resulta esencial en cualquier estudio científico, ya que permite una gestión eficiente de los recursos y un seguimiento riguroso del avance. Mediante el análisis y la acción planificada se buscaba garantizar un progreso fluido y efectivo hacia la meta planteada.

4.1.1. Fases

A continuación, se detallan las fases previstas del proyecto. Estas fases se diseñaron para garantizar un enfoque estructurado y eficiente en el desarrollo, permitiendo una gestión adecuada de los recursos y una entrega exitosa del producto final.

Durante estas fases se abordarían aspectos clave como la definición de objetivos, la comprensión de requisitos del sistema, el análisis y diseño de la solución y la realización de pruebas para asegurar la calidad y la funcionalidad del sistema.

Fase 1: Investigación y Definición

El objetivo de esta fase consistía en delimitar con claridad el alcance del proyecto y recopilar la base teórica necesaria. Se preveían los siguientes hitos:

- **Investigación y definición.** Se revisarían la literatura científica y proyectos previos relacionados con la aplicación de *ML* en TDAH, analizando artículos y documentación sobre algoritmos aplicados a la detección de patrones de comportamiento en niños y adolescentes y su potencial uso en apoyo al diagnóstico.

- **Definición de la idea.** Se identificarían los objetivos principales, las preguntas de investigación y las hipótesis a contrastar.
- **Reuniones iniciales.** Se programarían encuentros previos a la presentación oficial del TFG para validar la viabilidad del enfoque.
- **Lectura bibliográfica.** Se profundizaría en la teoría del TDAH y en los modelos de *ML* relevantes.

Fase 2: Análisis de Datos

En esta fase se llevaría a cabo un análisis exhaustivo de los datos destinados al posterior entrenamiento de modelos de *ML* para detectar patrones de comportamiento en niños con TDAH. El objetivo principal sería garantizar la calidad, coherencia y relevancia de los datos, de modo que los modelos pudieran aprender de forma efectiva. Se trataba de una fase crítica y, previsiblemente, la de mayor consumo de tiempo.

1. **Exploración de datasets.** Se identificarían fuentes públicas (estudios clínicos, encuestas a familias y docentes, bases educativas). Se analizarían cantidad y diversidad, balance de clases, atípicos y posibles sesgos.
2. **Estado del arte.** Se revisarían trabajos recientes que aplican *ML* al TDAH y se analizarían modelos como *Random Forest*, *SVM*, redes neuronales, etc. Se registrarían desafíos recurrentes (sesgos, escasez de etiquetas, baja reutilización en futuros *datasets*, etc.).
3. **EDA (Análisis Exploratorio de Datos).** Véase la Figura 4.1.
 - **Visualización y descriptivos.** Se elaborarían gráficos y estadísticas para comprender distribuciones y detectar anomalías.
 - **Limpieza.** Se tratarían valores nulos, duplicados e inconsistencias.
 - **Transformaciones.** Se normalizarían variables numéricas, se codificarían categóricas y se aplicarían técnicas frente al desbalance.
 - **Correlaciones.** Se estudiarían relaciones entre variables para la selección de las más relevantes.
4. **Especificación de requisitos.** Se establecerían umbrales mínimos de calidad (completitud, balance, representatividad), se definiría la estructura final de entrada y se implementaría un *pipeline* de preprocesamiento coherente.

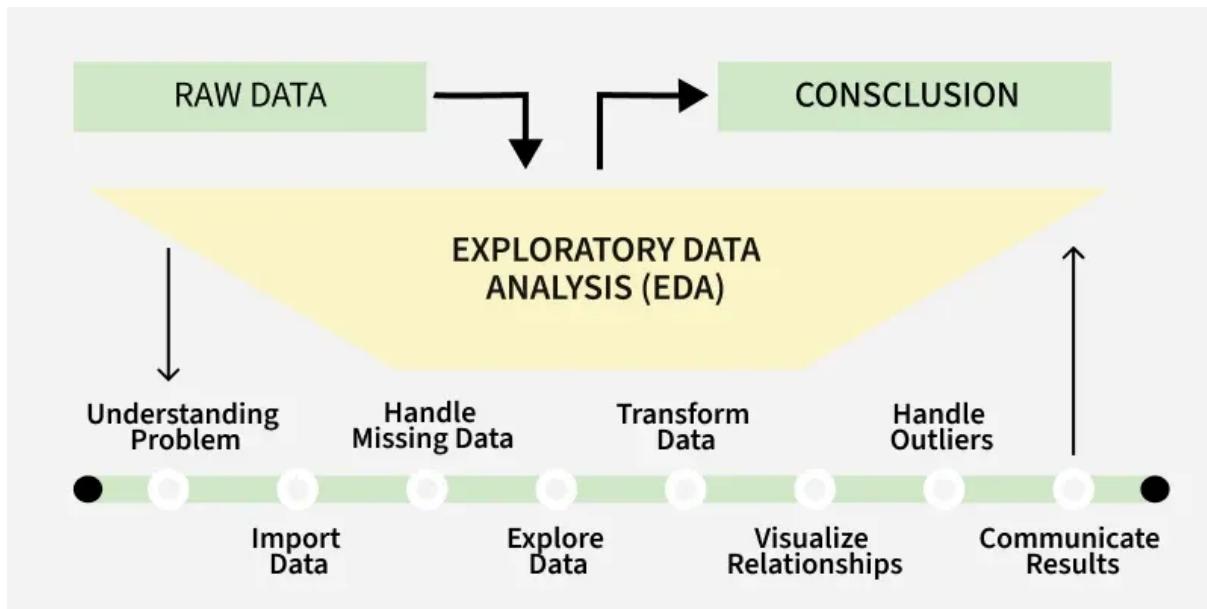


Figura 4.1: Esquema de EDA en *Machine Learning*.

Nota. Adaptado de “What is Exploratory Data Analysis?” (2.^a ed., p. 4), de GeeksforGeeks (2025), 2025, GeeksforGeeks. Copyright 2025.

Fase 3: Desarrollo del modelo

Para el desarrollo se planteó un trabajo secuenciado en **notebooks** de Google Colab, donde se prepararían los datos, se entrenaría y evaluaría el modelo y se generaría los artefactos finales para su uso en la aplicación. El paradigma previsto se basaría en un **pipeline supervisado de scikit-learn** (véase la Figura 4.2).

0_data_loading_and_split : Se cargarían los datos, se revisarían dimensiones y variables y se realizaría la partición estratificada en *train/test* (guardando ambos para usos posteriores).

1_data_clearing_and_preprocessing : Se limpiaría el *dataset* (tipos, nulos, textos especiales, duplicados) y se dejarían las columnas listas para modelado.

2_eda_exploratory_analysis : Se explorarían variables (distribuciones y relaciones) y se prepararían transformaciones/codificaciones (binarias, categóricas, ordinales y numéricas), generando un conjunto “*model-ready*”.

3_model_training_and_validation : Se entrenaría y ajustaría el modelo (búsqueda de hiperparámetros y validación), se evaluarían métricas y se exportaría el *pipeline* serializado (.pk1) para su integración.

Pipeline - Desarrollo del Modelo

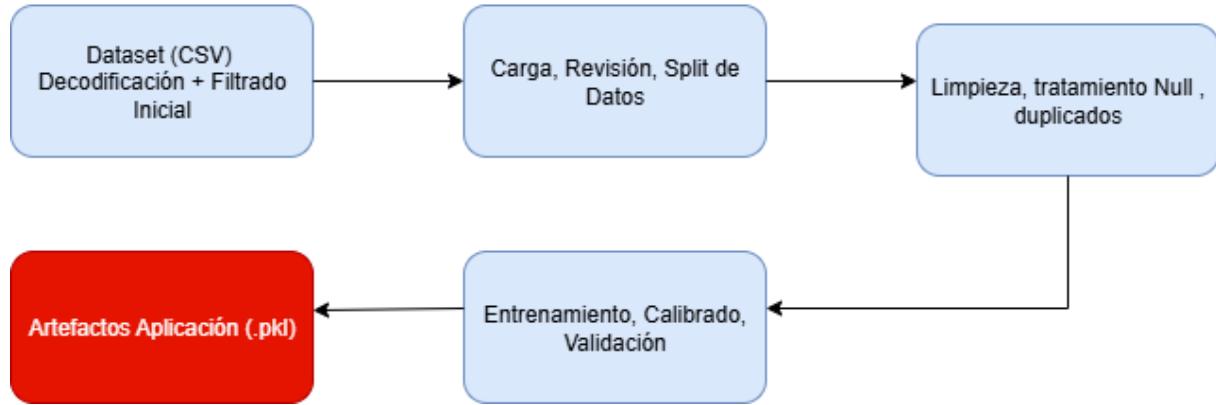


Figura 4.2: Pipeline de *ML* supervisado (diagrama propio elaborado con diagrams.net (2025)).

Fase 4: Desarrollo del sistema

El desarrollo de la aplicación seguiría un enfoque **DevOps** (véase la Figura 4.3), centrado en portabilidad, trazabilidad y calidad: repositorio en **GitHub**, *workflows* de CI para ejecutar *tests* y construir imágenes **Docker**, y despliegue local reproducible con **Docker Compose** (servicios de **FastAPI** y **Streamlit** separados). En el *backend* se definiría un **contrato de datos** con **Pydantic**, preprocesado robusto y un **pipeline de inferencia** que cargaría el modelo serializado (**.pk1**) y las métricas (**metrics.json**); el *frontend* ofrecería tres vistas (Formulario, Explorar y Modelo) para predicción individual, experimentación y transparencia del árbol.

Respecto a **MLOps**, se aplicarían prácticas ligeras: entrenamiento en *notebooks* con semillas y registro de parámetros, exportación del *pipeline* de SCIKIT-LEARN para inferencia y estandarización del *input* (12 variables clave). No se implementaría un MLOps *end-to-end*; por ello, el enfoque quedaría descrito como **DevOps** para la aplicación con **MLOps ligero** para asegurar reproducibilidad.

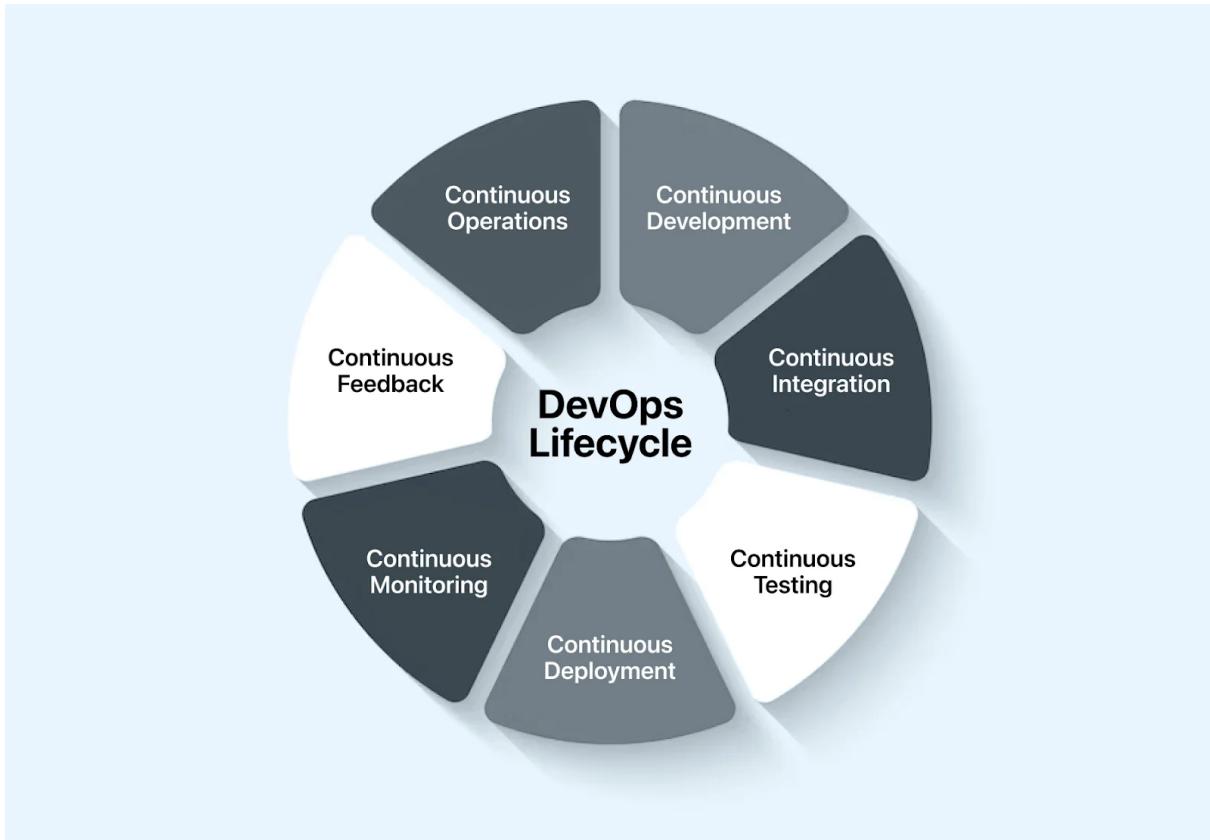


Figura 4.3: Esquema DEVOPS.

Nota. Adaptado de The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations (2.^a ed., p. 4), de Kim et al. (2021), 2021, IT Revolution Press. Copyright 2021.

Fase 5: Documentación

La documentación consistiría en recopilar, organizar y sintetizar la información relativa al alcance, decisiones técnicas y criterios de calidad. Se emplearía **LaTeX** en **Overleaf (2025)** para edición colaborativa, control de versiones del texto y formato académico consistente.

Fase 6: Tutorías del proyecto

Esta fase comprendería reuniones periódicas con la tutoría para el seguimiento del trabajo y revisión de hitos.

4.1.2. Diagrama de Gantt

Con el objetivo de ofrecer una visión detallada de la planificación, se elaboró un diagrama de Gantt que mostraba la estimación temporal por fases y su secuencia. Este diagrama permitía estimar el esfuerzo global y servía como base de comparación entre la previsión y la ejecución final.

Diagrama de Gantt provisional

En la previsión inicial, el proyecto se planificó para 181 días (del 17/12/2024 al 16/06/2025), con una duración total aproximada de **seis meses**. La Figura 4.4 recoge el diagrama Gantt provisional.

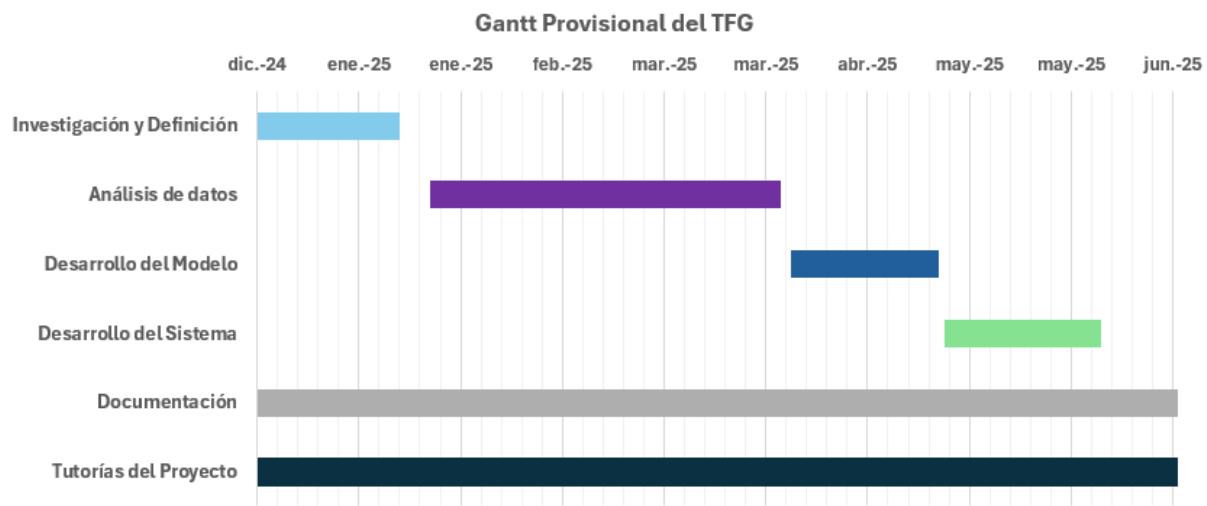


Figura 4.4: Diagrama de Gantt — estimación inicial.

Diagrama de Gantt real

Durante la ejecución, el proyecto se desarrolló desde el **17/12/2024** hasta el **07/11/2025**, con una duración total de aproximadamente **once meses** (casi 10 meses y 3 semanas). La Figura 4.5 muestra el diagrama Gantt real.

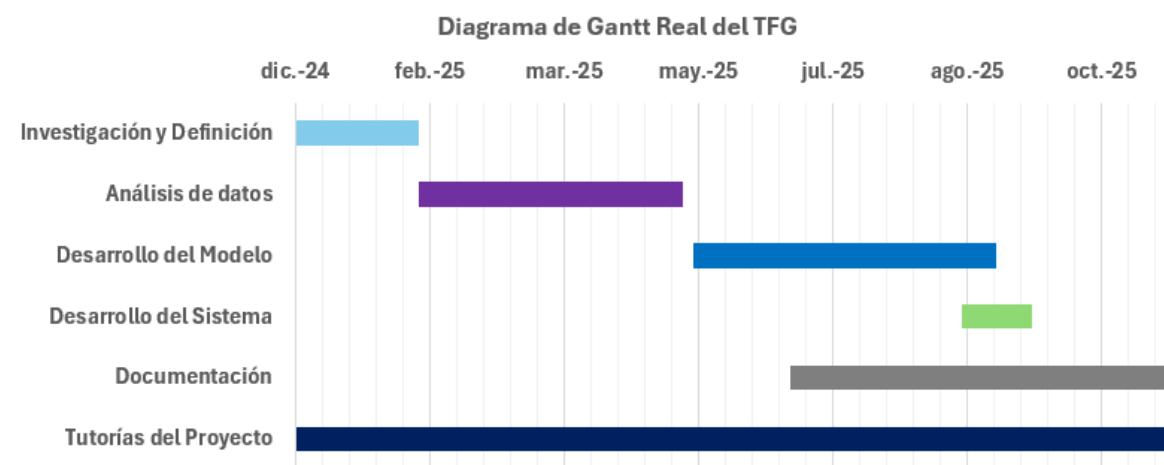


Figura 4.5: Diagrama de Gantt — duración real del proyecto.

Distribución real de horas por fase del proyecto

La distribución efectiva por fases quedó registrada como se recoge en la Tabla 4.1 y en la Figura 4.6.

Tabla 4.1: Planificación de horas por fase del proyecto (Total 300h)

Fase	Duración (HH:MM)	Porcentaje total
Análisis de Datos	72:54	24,3 %
Desarrollo del Modelo	66:00	22,0 %
Desarrollo del Sistema	35:06	11,7 %
Documentación	64:30	21,5 %
Investigación y Definición	40:48	13,6 %
Tutorías del Proyecto	20:42	6,9 %
Total	300:00	100,0 %

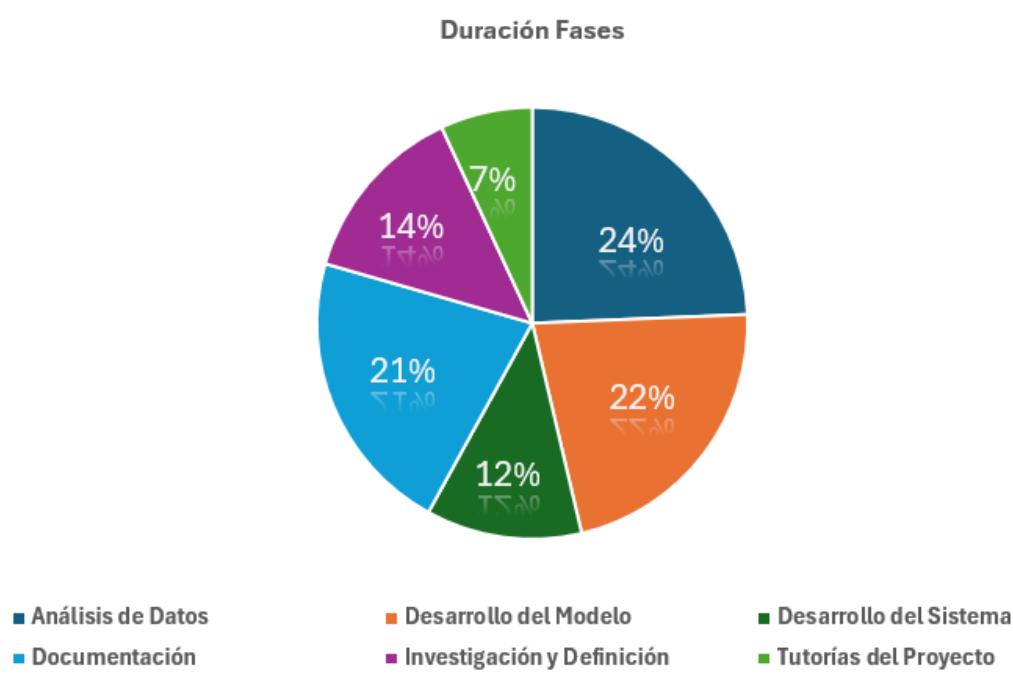


Figura 4.6: Porcentaje de tiempo por fases (gráfico elaborado en Excel).

4.2. Estudio presupuestario

En esta sección se detallan las partidas que generarían gasto y serían necesarias para la ejecución del proyecto.

4.2.1. Licencias de uso

Para un presupuesto realista se incluirían las licencias de software y servicios utilizados durante el desarrollo. En este proyecto se previó el uso de herramientas de suscripción, fundamentales para la redacción de la memoria y el desarrollo. Los costes asociados se recogen en la Tabla 4.2.

Tabla 4.2: Costes asociados a licencias de uso

Concepto	Coste mensual	Coste total (7 meses)
Paquete Microsoft Office	7,00 €	49,00 €
Overleaf Premium	8,00 €	56,00 €
Google One	2,99 €	20,93 €
Stata (Basic Edition)	8,00 €	50,00 €
TOTAL	25,99 €	175,93 €

4.2.2. Recursos materiales

En este apartado se considera exclusivamente software con licencia de pago empleado en el proyecto. Concretamente, se contempla la amortización de **Windows 11 Profesional**, como se puede observar en la Figura 4.7, con coste inicial de **259 €** y vida útil estimada de **5 años (60 meses)** (Tabla 4.3).

$$\text{Amortización anual del software} = \frac{259 \text{ €}}{5} = 51,80 \text{ €}$$

Figura 4.7: Amortización anual de Windows 11 Profesional.

Para el periodo del TFG (7 meses) **se aplica** el prorratoeo indicado en la Figura 4.8:

$$51,80 \text{ €} \times \frac{7}{12} = 30,18 \text{ €}$$

Figura 4.8: Prorratoeo de la amortización anual del software (7/12).

Tabla 4.3: Resumen de costes materiales del proyecto

Concepto	Coste anual (€)	Coste proyecto (7 meses) (€)
Amortización Windows 11 Profesional	51,80	30,18
TOTAL	51,80	30,18

4.2.3. Costes de personal

Para la estimación se considera un único **Data Scientist junior** con un contrato a **tiempo parcial**, coherente con la dedicación total del proyecto (300 horas) distribuida en 11 meses (aprox. 6-7 horas/semana).

Como referencia normativa (España, 2025), la **base mínima de cotización** para un contrato a jornada completa es **1.550,00 €/mes** y la **base máxima 4.720,50 €/mes** (Tabla 4.4). Dado que este proyecto se plantea con una dedicación parcial, se estiman unas bases de cotización adaptadas a dicha parcialidad para el cálculo.

Con estas bases, se calcularon los costes mínimo y máximo para 11 meses, usando unas bases de cotización estimadas para un contrato parcial (Tabla 4.5).

A partir de ambos extremos, se obtuvo el coste medio (Tabla 4.6), que se aproxima al objetivo de 1.500 € mensuales que cobra de media un **Data Scientist junior** en España.

Tabla 4.4: Tipos de cotización en 2025

Tipo de cotización	Empresa	Trabajador
Común	23,60 %	4,70 %

Tabla 4.5: Coste mínimo y máximo (Estimación TFG parcial)

Concepto	Coste mensual (€)	Coste total (11 meses) (€)
Base mínima (TFG Parcial)	900,00	9.900,00
Cotización empresa (23,60 %)	212,40	2.336,40
Cotización trabajador (4,70 %)	42,30	465,30
TOTAL (Base mínima)	1.154,70	12.701,70
Base máxima (TFG Parcial)	1.438,26	15.820,86
Cotización empresa (23,60 %)	339,43	3.733,73
Cotización trabajador (4,70 %)	67,60	743,60
TOTAL (Base máxima)	1.845,29	20.298,19

Tabla 4.6: Coste medio asociado al personal (Estimación TFG)

Concepto	Coste mensual (€)	Coste total (11 meses) (€)
Base media de cotización	1.169,13	12.860,43
Cotización empresa (23,60 %)	275,92	3.035,12
Cotización trabajador (4,70 %)	54,95	604,45
TOTAL (Base media)	1.500,00	16.500,00

4.2.4. Costes de despliegue y mantenimiento

Para el despliegue de la aplicación web se optó por contenedores Docker en un entorno *cloud*. Se asumió el uso de una cuenta de estudiante de AWS (acceso gratuito a ciertos servicios de *free tier*). No obstante, se estimaron costes de almacenamiento, tráfico adicional y mantenimiento técnico (Tabla 4.7).

Tabla 4.7: Coste estimado de despliegue y mantenimiento de la API

Concepto	Coste mensual (€)	Coste total (7 meses) (€)
Alojamiento en AWS EC2 (free tier)	0,00	0,00
Uso adicional estimado (tráfico, <i>logs</i>)	1,50	10,50
Mantenimiento técnico de la API	15,00	105,00
TOTAL	16,50	115,50

4.2.5. Otros costes asociados

Además de los costes directos, se consideraron costes indirectos (electricidad, Internet, impresión del documento). Para su estimación se tomó el 10 % del coste medio de personal (28.154,21 €), resultando 2.815,42 € (Tabla 4.8).

Tabla 4.8: Otros costes asociados al desarrollo del proyecto

Concepto	Coste estimado (€)
Electricidad e Internet (10 % del coste personal medio)	2.815,42
Impresión y encuadernación del TFG	20,00
TOTAL	2.835,42

4.2.6. Presupuesto total

La Tabla 4.9 resume el presupuesto global, agregando las partidas anteriores y mostrando tanto el coste mensual estimado como el total para los 7 meses de duración del TFG.

Tabla 4.9: Presupuesto total del proyecto

Categoría	Coste mensual (€)	Coste total (7 meses) (€)
Personal (coste medio estimado)	4.022,03	28.154,21
Licencias de uso	17,99	125,93
Amortización técnica (equipo y software)	24,31	170,18
Despliegue y mantenimiento	16,50	115,50
Otros costes asociados	405,06	2.835,42
TOTAL GENERAL	4.485,89	31.401,24

Capítulo 5

Metodología

Este capítulo presenta el pilar metodológico del proyecto, describiendo el proceso sistemático y reproducible seguido para construir, entrenar y desplegar el modelo de *ML*. La metodología se ha diseñado para ser un proceso riguroso de extremo a extremo, aplicado en este trabajo al análisis de patrones en datos de encuestas de salud infantil.

Como fuente principal de datos se emplea la Encuesta Nacional de Salud Infantil (**NSCH**) en su edición de **2023**. Si bien la descripción detallada de este *dataset*, sus variables y el caso de uso específico se abordarán en profundidad en el Capítulo 6 (ver Sección 6.1), es fundamental establecer en este capítulo la metodología de ingesta, preparación y procesamiento que se aplica sobre dichos datos.

El capítulo se estructura para seguir el ciclo de vida completo del proyecto. En primer lugar, se presentará un mapa visual del proceso (Sección 5.1) que ilustra los dos grandes carriles de trabajo: la Ciencia de Datos y el desarrollo de la aplicación (*App/DevOps*). A continuación, se detallará el desarrollo metodológico paso a paso (Sección 5.2), abarcando desde la definición del problema y las métricas, la ingesta y partición de los datos (NSCH 2023), el preprocesado y la ingeniería de características, hasta el entrenamiento, calibración y empaquetado del modelo final. Finalmente, se justificará el enfoque *MLOps* ligero adoptado (Sección 5.3), un componente clave que garantiza la trazabilidad y reproducibilidad de todo el sistema.

5.1. Mapa del proceso

La metodología del proyecto, ilustrada en la Figura 5.1, se fundamenta en un **diseño de "doble carril"**. Esta es una decisión arquitectónica clave motivada por la necesidad de **separar el ciclo de experimentación analítica** (la Ciencia de Datos) **del ciclo de entrega operativa** (la aplicación web o App/DevOps). Esta separación garantiza que el desarrollo del modelo y el desarrollo del *software* puedan avanzar en paralelo de forma robusta y mantenible.

El primer carril, **Data Science** (bloques [1] al [5]), describe el *pipeline* completo para la creación y evaluación del modelo. Este proceso es iterativo y se centra en construir un artefacto fiable y trazable. Comienza con la definición del problema (clasificación binaria de TDAH), las métricas y las reglas *anti-leakage* (Bloque 1). Continúa con la ingesta y partición estratificada de la fuente de datos (NSCH 2023) (Bloque 2). Posteriormente, se realiza una higiene de datos y preprocesado (Bloque 3) y un análisis exploratorio (EDA) junto a la ingeniería y selección de características (Bloque 4). Este carril culmina con el entrenamiento, validación y calibración del modelo (Bloque 5).

El **punto de interconexión** entre ambos carriles es el resultado del Bloque 5: un **artefacto de modelo serializado** (ej. un fichero `.pkl`) que encapsula toda la lógica de predicción y preprocesado.

El segundo carril, **App / DevOps** (bloques [6] al [8]), describe la arquitectura del sistema *software* que consume dicho artefacto. El diseño de este sistema está **desacoplado en dos piezas** (o servicios) para mejorar la mantenibilidad y escalabilidad:

- Un **Backend** (API): Se empaqueta el *pipeline* de inferencia (Bloque 6) en una API REST (construida con **FastAPI**). Esta pieza es el "cerebro" del sistema: recibe peticiones, procesa los datos de entrada en tiempo real y devuelve la predicción del modelo.
- Un **Frontend** (UI): Se desarrolla una interfaz de usuario (construida con **Streamlit**) (Bloque 7). Esta es la aplicación web con la que interactúa el usuario final (clínico, docente), enviando datos a la API y visualizando los resultados y la explicabilidad.

Finalmente, el Bloque [8] representa el despliegue de ambos servicios (API y UI) mediante **contenedores Docker** y Docker Compose, lo que garantiza un entorno de ejecución coherente, aislado y reproducible en cualquier máquina.

Las siguientes secciones de este capítulo profundizarán en las decisiones técnicas de cada uno de estos ocho bloques.

Esquema de la Metodología del proyecto.

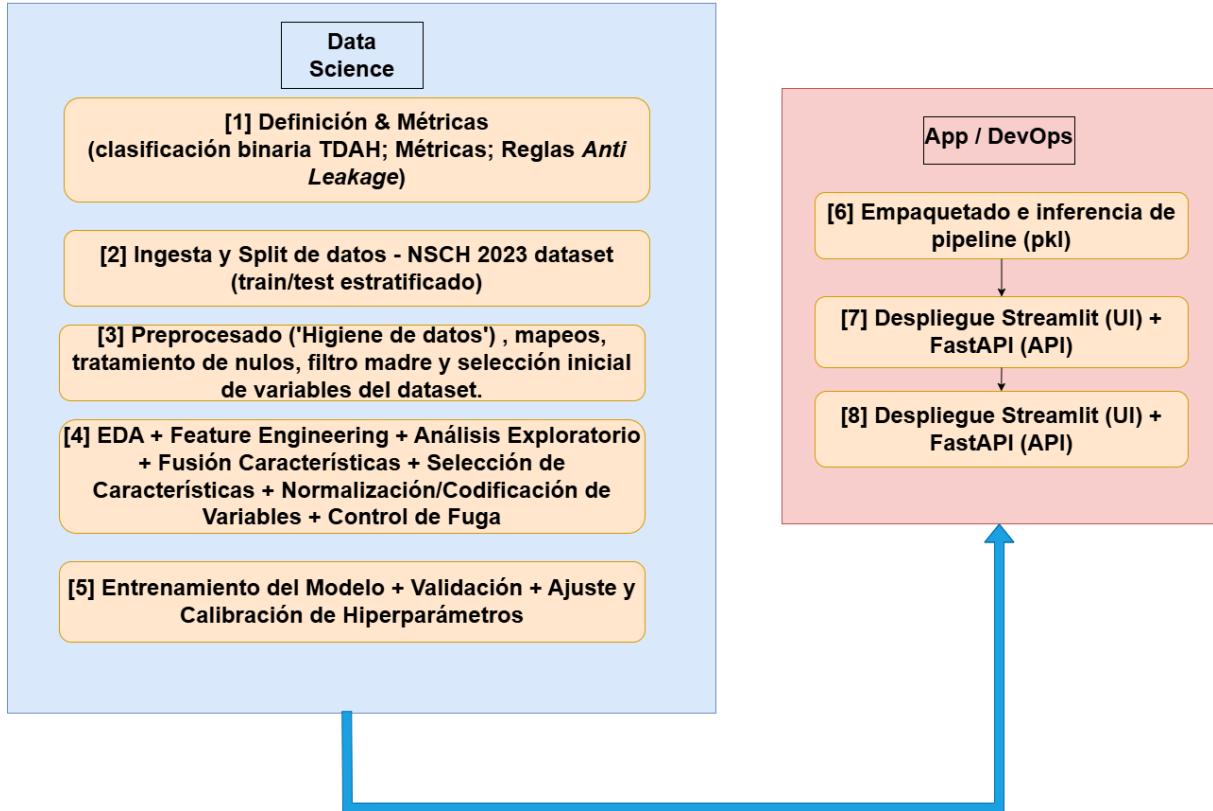


Figura 5.1: Diagrama de flujo de la metodología del proyecto

5.2. Desarrollo metodológico

Esta sección detalla el desarrollo del modelo predictivo, comenzando por la definición del problema (clasificación binaria) y las métricas. Se describe el pipeline de datos, que incluye la ingestión, un riguroso preprocesado y la ingeniería de características para evitar fugas de información.

A continuación, se explica el entrenamiento y la calibración de un modelo priorizando la explicabilidad. Finalmente, se aborda la operacionalización del proyecto: el empaquetado del modelo, la creación de una API (FastAPI) y una interfaz (Streamlit), y su despliegue en contenedores (Docker).

5.2.1. Definición y métricas

El problema se formula como **clasificación binaria** (riesgo de TDAH vs. no TDAH), priorizando *sensibilidad/recall* y la **F1** de la clase positiva porque, en un contexto de cribado clínico, es más crítico no pasar por alto casos de riesgo (minimizar falsos negativos) que reducir al mínimo los falsos positivos. Además la F1-score **equilibra precisión y sensibilidad** en escenarios desbalanceados. Como métrica complementaria, se reporta *ROC-AUC* para evaluar la capacidad discriminativa independiente del umbral.

Este encuadre condiciona el resto: se favorecen modelos interpretables, se establecen **políticas anti-leakage** (excluir predictores que revelan directa o indirectamente el diagnóstico; ajustar imputación, codificación y normalización solo con datos de entrenamiento (dentro de la validación cruzada) y mantener un **conjunto de prueba retenido** que no se utiliza en ninguna fase de entrenamiento ni ajuste) y se realiza la evaluación final exclusivamente sobre dicho conjunto retenido.

Esto se justifica de esta forma, ya que en contextos clínicos interesa **no perder casos de riesgo** y comprender por qué el modelo decide lo que decide.

5.2.2. Ingesta y *split* de datos

La metodología está diseñada para operar sobre datos con un esquema **tabular**, como los provenientes de cuestionarios escolares o psicológicos, encuestas o ficheros estructurados (ej. .csv o .dta).

El primer paso consiste en la **ingesta y preparación** de estos datos. Este proceso incluye la decodificación de variables mediante la ejecución de *scripts* de preparación para transformar los datos brutos en un formato analítico estructurado y limpio.

En la implementación de este trabajo, por ejemplo, se aplicó esta metodología a los microdatos de la NSCH 2023. En este caso particular, la fase de ingestión y preparación requirió la ejecución de los *scripts* de Stata proporcionados por la fuente original para realizar la correcta decodificación de las variables.

Una vez obtenidos los datos tabulares limpios, se realiza una **partición estratificada**, utilizando semillas fijas para garantizar la reproducibilidad de los experimentos. El conjunto de *test* resultante queda sellado y no se utiliza bajo ninguna circunstancia hasta la evaluación final del modelo.

Todo el trabajo analítico y las decisiones de procesamiento posteriores (ej. imputación, escalado de características, selección de variables) se aprenden y ajustan usando **exclusivamente el conjunto de train**. Las transformaciones, parámetros o estadísticos resultantes (ej. la media para el escalado) se guardan y se aplican después, sin recalcularlos, a los conjuntos de validación y *test*.

El hecho de separar rigurosamente los ciclos de aprendizaje/validación de la evaluación final evita el optimismo en los resultados y asegura una estimación realista del rendimiento del modelo en datos no vistos.

5.2.3. Preprocesado (“higiene de datos”)

Se normaliza el formato: tipos coherentes, codificación de valores especiales a NaN, eliminación de duplicados y redundancias. Se definen mapeos por tipo (binaria, ordinal, categórica, numérica) y reglas de imputación y transformación que se guardan como parte del pipeline. Las decisiones sensibles (p. ej., *logical skip*) se tratan de forma consistente —ya sea como nulos informados, fusiones de variables o categorías específicas— según la documentación del cuestionario.

Una higiene estricta constituye la base de cualquier conclusión fiable y facilita reutilizar el pipeline con futuras ediciones de la encuesta.

5.2.4. EDA + Feature Engineering + Selección

Se auditán distribuciones, cardinalidades y coherencia. Se aplican fusiones semánticas (p. ej., consolidar diagnóstico/estado/severidad cuando procede, evitando acercarse de forma trivial a la etiqueta). El filtrado inicial combina criterios estadísticos por tipo de variable (Spearman, Chi-cuadrado, etc.) con criterio clínico. Se codifican variables (ordinal/categóricas) y se normalizan cuando es necesario para etapas posteriores.

Se dedica un esfuerzo especial a detectar y evitar la fuga de información (variables que “contienen” el diagnóstico o consecuencias directas del mismo) para no sobreestimar el modelo.

El objetivo es mantener un conjunto de variables informativo, que capture señales indirectas y útiles en entornos reales.

5.2.5. Entrenamiento, validación y calibración

Se entrena con validación cruzada estratificada, tratando el desbalance de clases con `class_weight='balanced'`. Se prioriza un Árbol de Decisión por su explicabilidad; se exploran alternativas como bosque aleatorio cuando aportan contraste. Se emplean búsqueda de hiperparámetros, poda/regularización, calibración de probabilidades y ajuste del umbral operativo sobre la curva *precision-recall* según criterio de uso.

El equilibrio entre rendimiento y comprensión resulta clave para la adopción en contextos educativos/clínicos.

5.2.6. Empaquetado e inferencia del pipeline

Se serializa un artefacto único (model.pkl) que incluye *preprocesado + modelo + calibración*, acompañado de:

- Contrato de entrada (schema.json) y mapeos.
- Metadatos de versión y métricas mínimas (metrics.json).
- Dependencias (requirements.txt).

La inferencia sigue un flujo determinista: validar → cargar → predecir → devolver probabilidades y clase.

Se adopta esta línea, ya que encapsular todo el flujo en un solo artefacto reduce errores, simplifica mantenimiento y habilita despliegues consistentes.

5.2.7. API (FastAPI) + UI (Streamlit)

Fase de desarrollo del *frontend* y *backend* de la aplicación. Se expone el modelo mediante **FastAPI** y una UI en **Streamlit** para permitir al cliente final exploración y transparencia. Se aplica este diseño en la aplicación, ya que un modelo útil necesita de una **interfaz** y un **contrato**. La combinación API + UI acelera test unitarios, demostraciones y la potencial mejora y adopción del aplicativo.

5.2.8. Despliegue con contenedores

Se envuelve todo en **Docker** y se orquesta con **Docker Compose**. La imagen incluye el artefacto del modelo y las dependencias. Esto permite ejecución local, en servidores o nube sin fricciones. El uso de los contenedores garantiza reproducibilidad del entorno y portabilidad del sistema.

5.3. Propuesta de Enfoque

La metodología seguida se inspira principalmente en **CRISP-DM** (*Cross-Industry Standard Process for Data Mining*) (véase Figura 5.2).

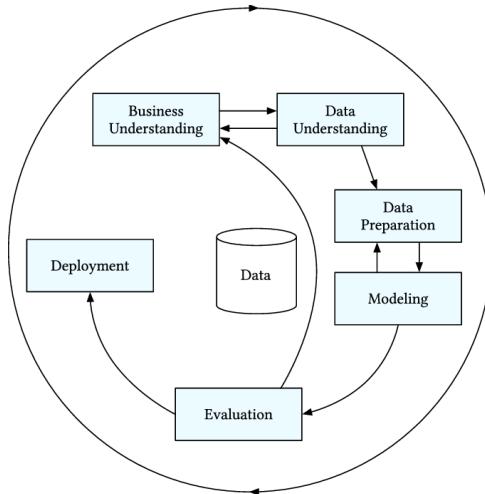


Figura 5.2: Esquema *CRISP-DM*.

Nota. Adaptado de “CRISP-DM” (Venkat (2017), 2017).

Además, se adapta un circuito **MLOps “ligero” o nivel 1**, que consiste en entrenar el modelo de forma continua automatizando el flujo de procesamiento del aprendizaje automático (véase Figura 5.3).

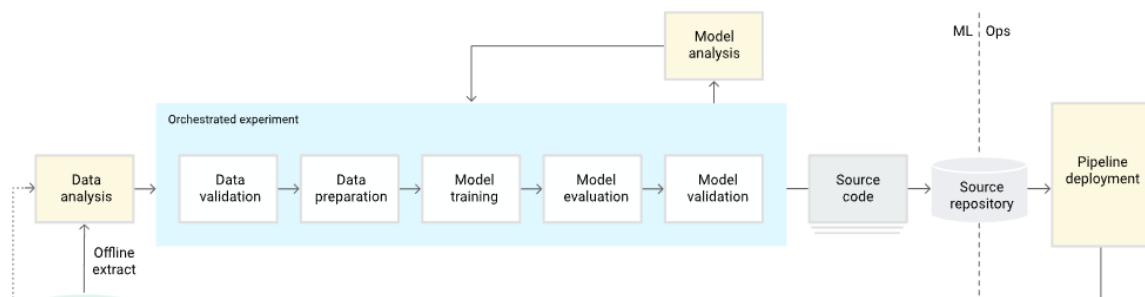


Figura 5.3: Automatización del flujo de procesamiento del aprendizaje automático para el entrenamiento continuo.

Nota. Adaptado de “MLOps: flujos de procesamiento de entrega continua y automatización en el aprendizaje automático” (Kazmierczak et al. (2024), 2024).

La metodología del proyecto no se limita a entrenar un modelo: se diseña como un circuito MLOps ligero que garantiza reproducibilidad, trazabilidad y entrega. En un problema sensible como la detección de riesgo de TDAH, no basta con obtener buenas métricas en un cuaderno; resulta imprescindible poder repetir el proceso extremo a extremo, auditar las decisiones y poner el resultado en manos del usuario de forma estable. Ese es, precisamente, el valor de aplicar MLOps aquí.

Se parte del principio de que cada fase del ciclo (extracción/ingesta, preprocesado, EDA/selección y entrenamiento/validación) debe estar separada, enlazada y parametrizada. Para ello, se construyen **notebooks independientes en Google Colab**, uno por etapa, con entradas

y salidas claras y un conjunto pequeño de parámetros (ruta/versión del NSCH, semillas, lista de variables). Esta organización permite re-ejecutar solo el tramo necesario cuando cambian los datos o se ajusta alguna decisión de ingeniería, evitando rehacer todo el pipeline.

Además, el trabajo se apoya en **control de versiones con GitHub**.

El repositorio¹ unifica el código de la aplicación (API con FastAPI y UI con Streamlit) y los recursos del pipeline (notebooks, scripts auxiliares, artefactos y documentación).

Mantener ramas de funcionalidad cortas y etiquetar los hitos (por ejemplo, cierre de EDA, calibración del modelo o *release* de la app) permite trazar exactamente qué versión de los datos y del código genera cada artefacto. Esta trazabilidad hace posible auditar una predicción o rehacer un experimento tiempo después con el mismo entorno y las mismas dependencias.

¹El código fuente del proyecto se encuentra disponible en la rama `main` del siguiente repositorio de GitHub:
<https://github.com/dcarmor99/TDAHTool/tree/develop>

Capítulo 6

Análisis experimental: construcción del modelo mediante el caso de estudio NSCH-2023

En el presente capítulo se documenta de forma integral la fase de **ciencia de datos y aprendizaje automático**, desde la caracterización del conjunto de datos inicial hasta la evaluación y validación del modelo final, así como, los hitos y resultados alcanzados en cada punto del proceso analítico de datos.

En primer lugar, se describe el dataset empleado, para enmarcar el problema y sus restricciones. A partir de ahí, se detallan las etapas de preprocesamiento, ingeniería y selección de características, y el entrenamiento con ajuste de hiperparámetros y control de reproducibilidad. Finalmente, se presentan los resultados mediante métricas apropiadas, curvas de desempeño y análisis de errores, incluyendo observaciones sobre calibración, robustez y limitaciones del enfoque.

Este recorrido ofrece una visión trazable y reproducible del pipeline seguido, justificando cada decisión técnica y su impacto en el rendimiento del sistema.

6.1. Descripción del dataset

Para el presente trabajo se utiliza el conjunto de datos proporcionado por la *National Survey of Children's Health* (NSCH) (U.S. Census Bureau, 2023), un recurso ampliamente reconocido y empleado en estudios sobre salud infantil en Estados Unidos. Esta encuesta, llevada a cabo por la *U.S. Census Bureau* y patrocinada por la *HRSA's Maternal and Child Health Bureau*, recopila anualmente información detallada sobre el bienestar físico y emocional de los menores, sus condiciones familiares y su entorno social.

La elección de NSCH 2023 responde a su amplitud temática y actualización reciente, que permiten abordar el problema con amplia evidencia representativa. El cuestionario incluye indicadores de salud mental y desarrollo conductual, condiciones médicas diagnosticadas, contexto familiar y socioeconómico, rendimiento escolar y hábitos de vida, todos ellos especialmente relevantes para el análisis del TDAH en población infantil. Además, el tamaño muestral —55.162 niños de 0 a 17 años— aporta potencia estadística y un volumen adecuado para el entrenamiento de modelos de aprendizaje automático.

Con el **objetivo** de construir un conjunto de variables **informativas y clínicamente plausibles**, se parte de una selección inicial de 82 características elaborada en colaboración con el Departamento de Psicología de la Universidad de Granada, y en especial a la psicóloga Purificación Checa Fernández (Checa Fernández, s.f.), procedente de un estudio previo relacionado con TDAH y autismo. Esta curación se complementa con la evidencia sintetizada por Maniruzzaman (Maniruzzaman et al., 2022), cuya revisión exhaustiva de la literatura destaca como variables predictoras factores como la edad del menor, el sexo, la edad materna, alergias, artritis, asma, daño cerebral, cefaleas (migrañas), ansiedad, depresión, consumo de alcohol, nacimiento prematuro o el índice de pobreza. De este modo, el análisis se apoya en un conjunto diverso de indicadores que habilita una lectura más profunda de las relaciones entre TDAH, salud mental y condiciones socioeconómicas, evitando un enfoque restringido a lo puramente clínico.

El archivo original se obtiene en formato **.dta**. Para interpretar correctamente la codificación y el significado de las variables, se utiliza *Stata* (StataCorp LLC, s.f.) y se aplica el script de decodificación distribuido junto con el dataset, asociando cada etiqueta a su descripción y categorías. Tras este proceso de decodificación, los datos se exportan a **.csv** para su tratamiento posterior en Python.

En cuanto a la **variable objetivo**, se emplea la respuesta de progenitores o tutores a la pregunta: “¿Alguna vez un médico o profesional de la salud les ha dicho que el niño seleccionado (S.C.) padece trastorno por déficit de atención o trastorno por déficit de atención con hiperactividad, es decir, TDA o TDAH?”. Esta variable se binariza asignando el valor 1 cuando la respuesta es *Sí* y 0 cuando es *No*.

6.2. Preprocesamiento de datos

El proceso de selección y preprocesamiento de datos es un pilar fundamental para cualquier proyecto de aprendizaje automático, especialmente en este del ámbito de la salud mental infantil, donde la calidad y relevancia de los datos determinan, en gran medida, la capacidad del modelo para detectar patrones clínicamente significativos.

Se parte del NSCH 2023 en formato **.dta**. Se realiza la decodificación completa en Stata con el script oficial de etiquetas y la exportación a **.csv**, preservando la correspondencia entre códigos y descripciones para garantizar trazabilidad clínica. A partir de las **457** variables iniciales, y combinando el criterio experto del Departamento de Psicología con la evidencia de la literatura, se construye una primera versión curada de **82** predictores clínicamente plausibles.

6.2.1. Armonización y controles de calidad

Se homogenizan tipos y formatos, corrigiendo inconsistencias (p. ej., números almacenados como texto) y depurando duplicidades. Se eliminan columnas redundantes como *ecerpals_desc.1*, *k2q31c.1* y *k2q40a.1*. Adicionalmente, se verifica la coherencia de rangos y categorías con el manual de variables del NSCH para evitar errores de codificación.

6.2.2. Manejo de ausencias y saltos lógicos

En una primera fase, las respuestas especiales del cuestionario (“*Not in universe*”, “*No valid response*”) y los campos vacíos se mapean a *NaN*, es decir, como valores nulos.

Inicialmente, los casos etiquetados como “*Logical Skip*” también se consideran nulos; sin embargo, tras revisar la lógica del cuestionario del NSCH, se constata que no representan ausencias en sentido estricto, sino condiciones de *no aplicabilidad* derivadas de respuestas previas. En consecuencia, y tal como se detalla posteriormente, se lleva a cabo *ingeniería de características* específica para manejar estos valores de forma adecuada.

6.2.3. Codificación numérica para el pipeline

Se genera una versión numérica consistente para el entrenamiento:

- Las variables binarias se mapean a los valores numéricos 0,1.
- Las variables ordinales siguen una escala numérica creciente preservando el orden y la gravedad/intensidad del valor original de la variable del cuestionario.
- Las variables categóricas se codifican como identificadores enteros, sin suponer orden alguno.

En paralelo, se mantiene una versión *desmapeada* (con etiquetas originales) para interpretación y reporte.

6.2.4. Imputación de valores faltantes

Se aplica una **imputación univariada mediante K vecinos (KNN)** adaptada al tipo de variable, preservando la semántica de los mapeos originales.

En concreto: para variables numéricas se imputa la media de los K vecinos válidos; para ordinales, la mediana respetando el orden; y para categóricas/binarias, la moda local. El procedimiento se integra en el flujo de preprocesamiento y, tras imputar, se realizan verificaciones de consistencia (frecuencias, rangos y coherencia con los diccionarios de categorías) para asegurar que la distribución resultante no distorsiona las categorías clínicas.

Los criterios conceptuales y consideraciones metodológicas de esta estrategia se desarrollan en el Capítulo 2, Sección 2.4.1 (*Imputación univariada mediante K vecinos*).

6.2.5. Resultado

Tras el ciclo de limpieza, armonización, tratamiento de ausencias y revisión exploratoria orientada a la calidad del dato, el conjunto final para entrenamiento queda reducido a **52** variables, equilibrando información relevante para el problema (véase Anexo A). Se dispone de:

- **Dataset numérico:** listo para el pipeline de entrenamiento y evaluación.
- **Dataset desmapeado:** con descripciones originales para lectura clínica y documentación.

6.3. EDA y Feature Engineering

El Análisis Exploratorio de Datos (EDA) es la etapa en la que se inspecciona y visualiza el conjunto de datos para comprender su estructura, calidad y particularidades antes del modelado. En paralelo, la *feature engineering* (fusión/creación de variables) permite traducir la lógica del cuestionario y el conocimiento de dominio en predictores más informativos.

6.3.1. Fusión de características

Durante el preprocesamiento se identifican múltiples variables afectadas por valores codificados como *Logical Skip*, que no representan datos ausentes en sentido estricto, sino omisiones debidas a reglas del cuestionario. Para evitar interpretaciones erróneas y pérdida de información potencialmente útil, se opta por **fusionar variables relacionadas** en nuevas variables categóricas/ordinales que reflejan tanto el diagnóstico como el estado actual de determinadas condiciones clínicas del menor.

Concretamente, en el caso de TDAH, ansiedad, depresión o problemas de conducta, se integran variables de diagnóstico, estado actual y severidad en variables ordinarias como `adhd_status`, `ansiedad_status`, `depresion_status` o `conducta_status`. Asimismo, se generan variables adicionales como `adhd_medicated` (tratamiento actual) y `educacion_especial_status` (plan educativo especial), de modo que el estado clínico y educativo del menor quede representado de forma más informativa y operativa para los análisis posteriores.

De igual modo, los **Logical Skip** en variables ordinarias individuales se abordan con una estrategia de limpieza consistente: sustitución por categorías informativas cuando procede (p. ej., “Too young”), imputación a nulos cuando la pregunta no es aplicable, o **fusión de indicadores relacionados** cuando es necesario mantener la lógica condicional (por ejemplo, `addtreat_clean` combinando diagnóstico de TDAH y tratamiento conductual).

6.3.2. Análisis Estadístico Descriptivo y estudio de *outliers*

Como paso previo al análisis de asociaciones, se realiza un control exhaustivo de calidad sobre el conjunto de entrenamiento.

Posteriormente, se examinan las distribuciones por tipo de variable mediante representaciones gráficas, lo que permite obtener una visión global del comportamiento de los datos:

- **Variables numéricas** (p. ej., `sc_age_years` y `a1_age`): se observan dos picos de casos diagnosticados con TDAH, como es en la primera infancia (2–5 años) y adolescencia tardía (15–17 años), coherentes con etapas críticas del desarrollo humano. Por otro lado, se observa que la mayoría de cuidadores adultos se sitúa en torno a los 40 años.
- **Transformación de `fpl_i1`:** esta variable (ingresos familiares respecto al umbral de pobreza) muestra una distribución altamente asimétrica con concentración en el valor 200 y presencia de valores atípicos. Dada su relevancia socioeconómica, se transforma en **variable categórica por rangos** (`fpl_group`), reduciendo el impacto de *outliers* y alineando su uso con prácticas habituales en estudios sociales y clínicos (véanse Figura 6.1 y Figura 6.2).

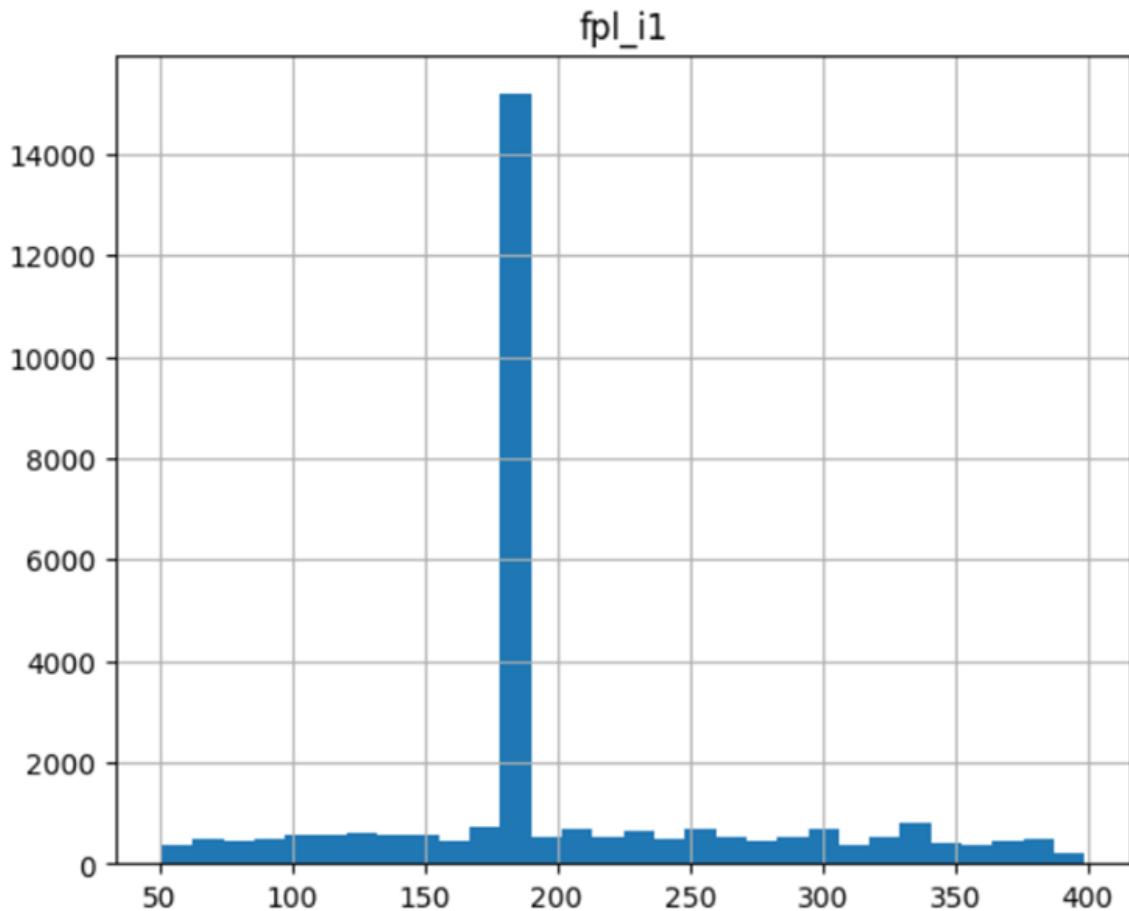


Figura 6.1: Distribución de la variable fpl_i1, outlier en 200

Nota. Adaptado del notebook de `2_eda_exploratory_analysis`.

En las **variables ordinales** se representan gráficamente las distribuciones de todas las seleccionadas, lo que permite validar la coherencia con la población infantil general. En variables de autorregulación emocional y atención predominan categorías intermedias (*Sometimes, Most of the time*); los extremos (*Always, Never*) son menos frecuentes, como cabría esperar. También aparece la categoría *Too young*, aplicable a menores de tres años según la lógica del cuestionario.

Las variables clínicas transformadas muestran una baja prevalencia tanto de diagnósticos confirmados como de intervenciones conductuales; predomina la ausencia de lesión cerebral y la falta de tratamiento del TDAH entre quienes no están diagnosticados.

En habilidades sociales y resolución de conflictos, las frecuencias se concentran en categorías positivas, con una minoría que presenta dificultades marcadas. En el ámbito escolar/funcional, la mayoría supera la educación primaria, no requiere educación especial y rara vez declara condiciones limitantes. En actividad física, el tiempo al aire libre aumenta los fines de semana, un patrón esperable en población infantil.

En cuanto a las **variables categóricas**, tras la depuración las distribuciones resultan **coherentes**: predominio del inglés en el hogar, equilibrio por sexo, diversidad de estructuras familiares con mayor presencia de hogares biparentales casados, alta cobertura sanitaria, rendimiento académico mayoritariamente alto (A/B), baja prevalencia de bajo peso al nacer y niveles socioeconómicos principalmente bajos o moderados.

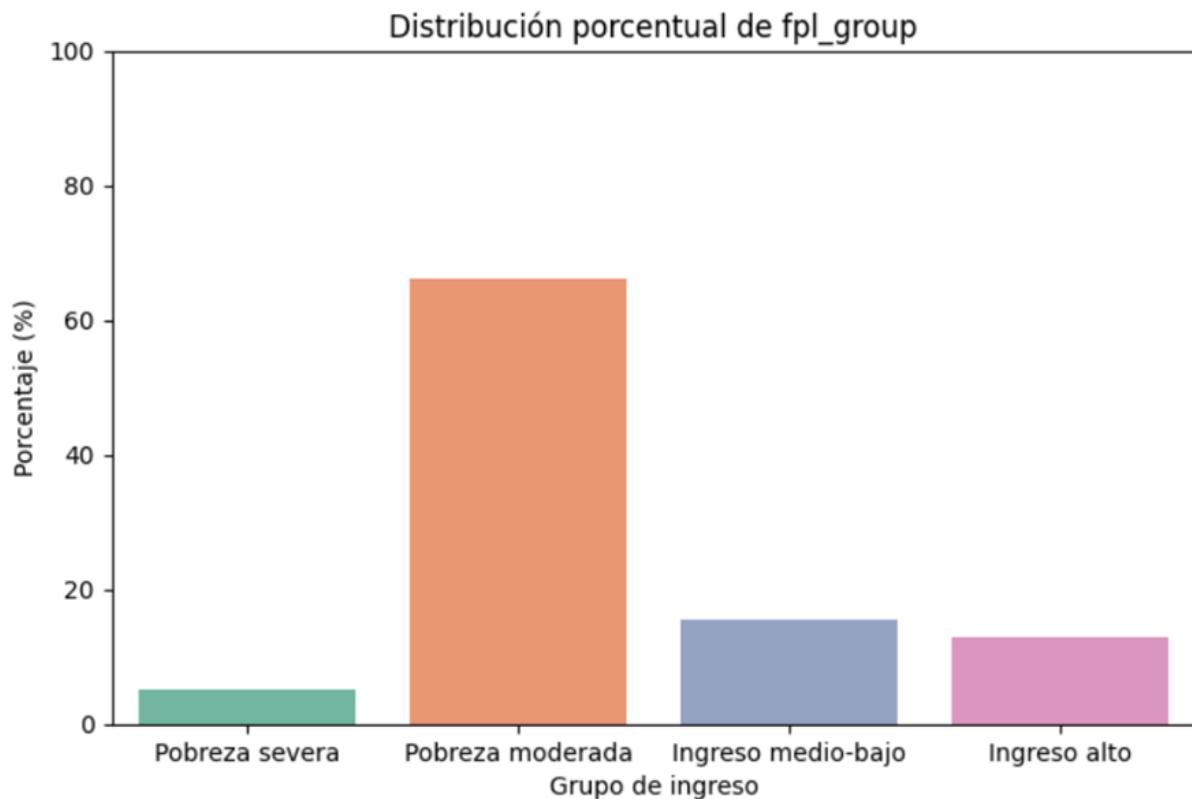


Figura 6.2: Creación de variable categórica fpl_group

Nota. Adaptado del notebook de *2_eda_exploratory_analysis*.

Para comprobar las distribuciones, estas se encuentran dentro del notebook "*2_eda_exploratory_analysis*" (Para Repositorio, véase Figura 6.16, ubicado dentro de la carpeta "notebooks")

6.3.3. Discriminación por sexo

De acuerdo con la literatura de referencia (Maniruzzaman et al., 2022), se efectúa un cribado por sexo del adulto de referencia, considerando **a1_sex = “Femenino”** para explorar posibles patrones ligados a la herencia materna.

6.3.4. Análisis de correlaciones respecto a la variable objetivo

Correlación entre variable objetivo y variables numéricas

Para **k2q31a_bin** (diagnóstico TDAH 0/1) y variables numéricas (**sc_age_years**, **birth_yr**, **a1_age**) se emplea la **correlación biserial puntual**, adecuada para asociaciones binaria–continua (véase Capítulo 2, Sección 2.5.1). Los resultados confirmán lo esperado:

- **sc_age_years** correlaciona positivamente con el diagnóstico (mayor edad, mayor probabilidad de detección).
- **birth_yr** muestra correlación negativa (a mayor año de nacimiento, menor edad y menor probabilidad de diagnóstico).
- **a1_age** presenta relación muy débil con el diagnóstico.

- Todas las correlaciones resultan estadísticamente significativas ($p < 0,05$).

	Variable	Coef.	Pearson	p-valor	Significativo (p < 0.05)
0	sc_age_years	0.235689	0.000000e+00		True
1	a1_age	0.135593	1.350952e-124		True
2	birth_yr	-0.228108	0.000000e+00		True

Figura 6.3: p-valor de Pearson para variables numéricas

Nota. Adaptado del notebook `2_eda_exploratory_analysis`.

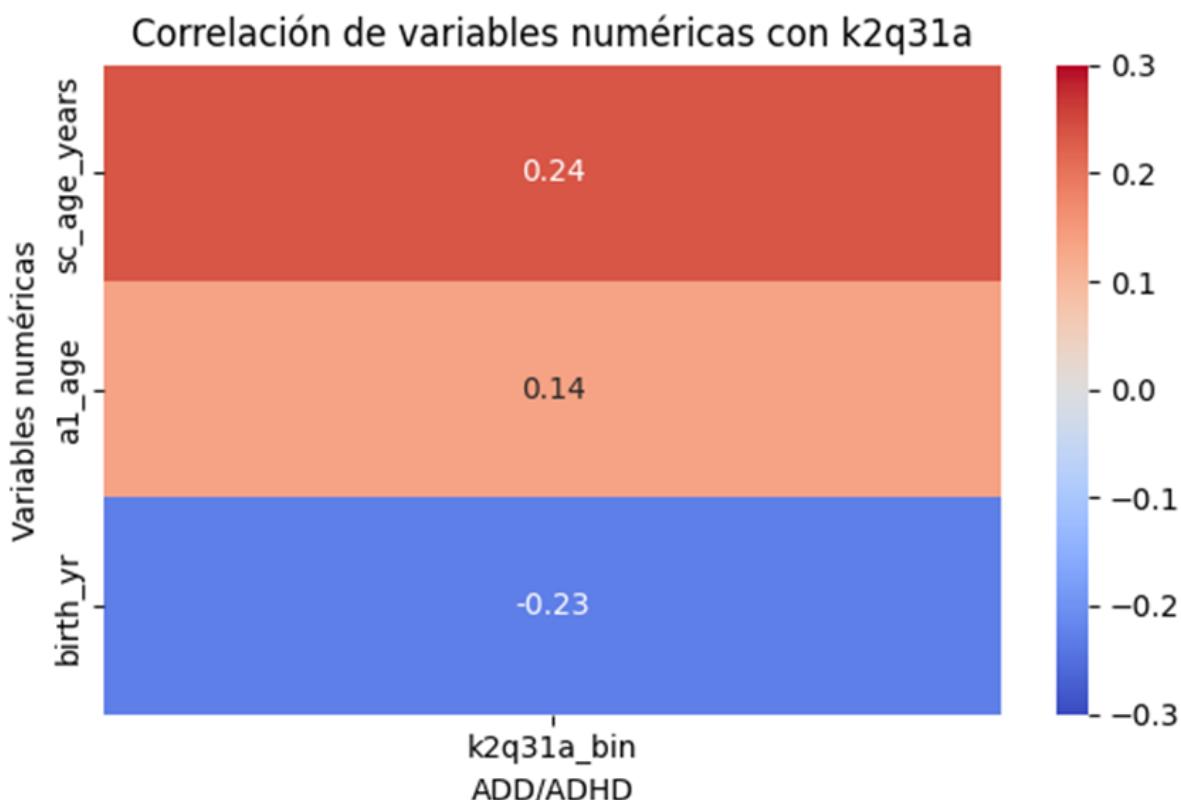


Figura 6.4: Matriz de correlaciones de variables numéricas

Nota. Adaptado del notebook `2_eda_exploratory_analysis`.

Correlación entre variable objetivo y variables ordinales

Para las ordinales se aplica el coeficiente de **Spearman**, que evalúa relaciones monótonas sin requerir linealidad ni normalidad (véase Capítulo 2, Sección 2.5.2). Previamente, todas las variables se codifican numéricamente según su orden lógico (sufijo `_num`). Principales hallazgos:

1. Variables directamente vinculadas al diagnóstico/tratamiento (`adhd_status`, `addtreat`, `adhd_medicated`, `conducta_status`, `educacion_especial_status`) muestran correlaciones altas o casi perfectas con la variable objetivo, señalando **fuga de información**; se excluyen del modelado.

2. Variables de autorregulación/ atención y habilidades sociales muestran correlaciones moderadas y útiles como señal indirecta.
3. Variables con correlación prácticamente nula (p. ej., tiempo al aire libre) se descartan.

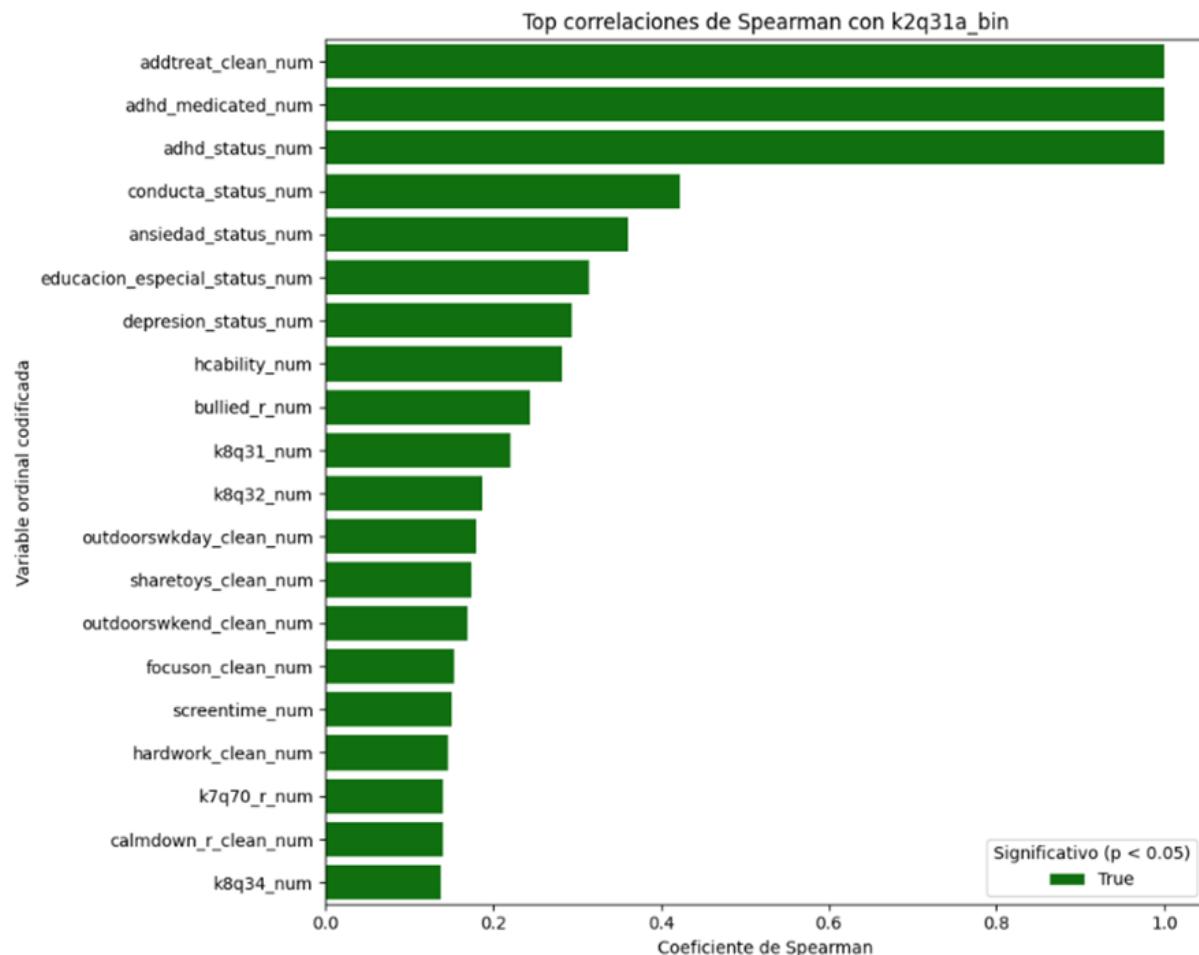


Figura 6.5: Matriz de correlaciones de variables ordinales

Nota. Adaptado del notebook `2_eda_exploratory_analysis`.

Correlación entre variable objetivo y variables binarias

Para las binarias se mantiene **Spearman** con el fin de aplicar un criterio homogéneo (véase Capítulo 2, Sección 2.5.2). Destacan por su relevancia:

- `k4q23` (medicación por problemas emocionales/conducta/concentración).
- `memorycond` (dificultades de memoria/concentración por condición física/mental/emocional).
- `ace9` (convivencia con alcohol/drogas), indicador contextual psicosocial.

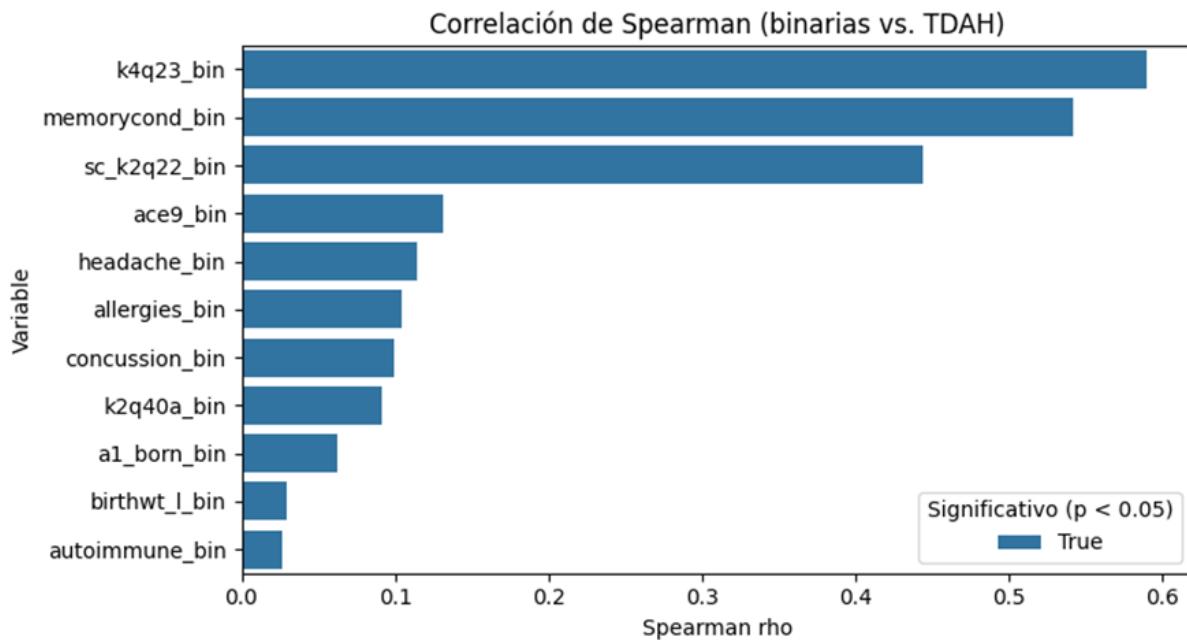


Figura 6.6: Matriz de correlaciones de variables binarias

Nota. Adaptado del notebook *2_eda_exploratory_analysis*.

Correlación entre variable objetivo y variables categóricas

Para las categóricas se aplica la prueba **Chi-cuadrado de independencia** (véase Capítulo 2, Sección 2.5.3) entre k2q31a_bin y cada variable. **Objetivo:** valorar si la distribución del diagnóstico varía según categorías. **Interpretación:** $p < 0,05$ sugiere asociación estadísticamente significativa. **Conclusiones:** grades (rendimiento), family_r (estructura familiar) y a1_marital (estado civil del cuidador) muestran asociación significativa, por lo que se priorizan como candidatas a predictores.

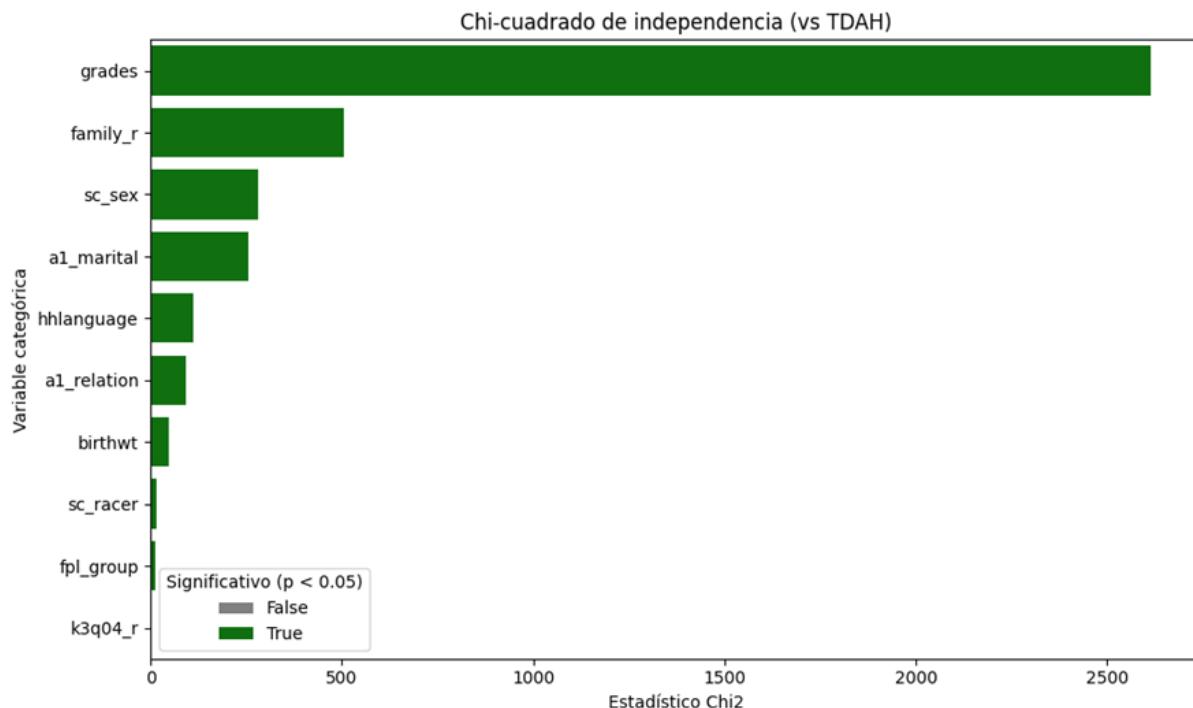


Figura 6.7: Matriz de correlaciones de variables categóricas

Nota. Adaptado del notebook `2_eda_exploratory_analysis`.

6.3.5. Hitos de la fase EDA

1. Fusión de indicadores clínicos/educativos en variables compuestas interpretables.
2. Transformación de `fpl_i1` a rangos para robustez e interpretación.
3. Identificación y exclusión de predictores con *fuga de información*.
4. Priorización de predictores con asociación significativa y plausibilidad clínica/social.

La justificación estadística de los coeficientes y tests empleados se detalla en el Capítulo 2.

6.3.6. Selección de características

Tras el EDA (Sección 6.3), se lleva a cabo una fase de **selección de características** con el objetivo de eliminar variables irrelevantes, redundantes o con riesgo de fuga de información, y así mejorar la eficiencia y el rendimiento del modelo. Esta sección documenta las decisiones ejecutadas y los hitos alcanzados; los conceptos teóricos de codificación y escalado se resumen en el Capítulo 2, Sección 2.6.1 y 2.6.2.

Filtro inicial por relevancia y redundancia.

Se descartan variables que, por su naturaleza o relación directa con la variable objetivo, no aportan valor al modelo o introducen redundancia (para ver el significado de las variables, consultese el Anexo E): `k3q04_r`, `k2q31b`, `K2q34b`, `fpl_i1` y `birth_yr_f`.

Selección preliminar basada en `f_classif`.

Se aplica una criba univariante con `f_classif` (ANOVA F-test) para estimar la asociación individual con la variable objetivo. A continuación, se evalúa un clasificador de referencia (Random Forest) variando el número de variables seleccionadas (K) y se observa un *punto de estabilización* del rendimiento a partir de 40 variables.

Conjunto de variables para modelado.

Con esa referencia, y combinando los resultados de correlación del EDA con variables reportadas en la literatura (p. ej., (Maniruzzaman et al., 2022)), se seleccionan finalmente **52 variables** para construir el conjunto de entrenamiento. Tras la codificación y las verificaciones de coherencia, el **dataset final usado en el modelado queda en 49 variables** (véase Anexo B).

Codificación One-Hot. Para incorporar variables categóricas en los algoritmos de aprendizaje automático, se aplica **One-Hot Encoding** (véase Capítulo 2, Sección 2.6.1). Cada categoría se proyecta a una columna binaria; se utiliza `drop_first=True` para evitar la colinealidad (“trampa de las dummies”). En particular, se codifican:

- **grades**: categorías de rendimiento académico del menor.
- **family_r**: estructura familiar del hogar (combinaciones de cuidadores biológicos/no biológicos).

La elección de esta técnica se apoya en la práctica estándar de representación categórica en ML (Zhou, 2021).

6.3.7. Dataset final para entrenamiento del modelo

Separación del conjunto de entrenamiento

A partir de la selección anterior, se construye `train_df_model` con variables numéricas, ordinales, categóricas y binarias ya preparadas para entrenamiento. Se revisan valores únicos por columna para asegurar interpretabilidad y facilitar validaciones.

Estandarización de variables

Dado que las variables operan en escalas distintas, se aplica **estandarización Z-score** (Capítulo 2, Sección 2.6.2) mediante `StandardScaler` (véase Figura 6.8):

$$Z_i = \frac{X_i - \mu}{\sigma}, \quad \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (X_i - \mu)^2}.$$

Figura 6.8: Estandarización Z-score empleada para homogeneizar escalas de entrada.

Con ello, cada variable queda con media 0 y desviación estándar 1, evitando que las de mayor rango dominen el aprendizaje, especialmente relevante para SVM, redes neuronales o KNN. En variables binarias procedentes de One-Hot, la estandarización sitúa las categorías raras en valores más extremos, efecto esperable que se tiene en cuenta en la interpretación.

Además, se documentan los mapeos de codificación ordinal/categórica y el tratamiento de categorías especiales (p. ej., `Too young (<3 years)`), garantizando trazabilidad.

Balanceo de clases

Dado el desbalance de `k2q31a_bin` (89 % no TDAH vs. 11 % TDAH), se valora SMOTE pero se descarta por introducir datos sintéticos y riesgo de *overfitting*. Se opta por `class_weight='balanced'` en los clasificadores de `scikit-learn`, ajustando la pérdida sin alterar el conjunto original, lo que mejora rigor y reproducibilidad.

Hitos de la fase Selección Características En esta fase se logran los siguientes hitos en el pipeline de ciencia de datos:

- Reducción informada desde el universo inicial hasta **52 variables** útiles.
- Codificación One-Hot controlando colinealidad.
- Estandarización Z-score integrada en el pipeline.
- Manejo del desbalance con `class_weight='balanced'` y sin datos sintéticos.

6.4. Machine Learning y entrenamiento

Durante la fase de entrenamiento se sigue una **estrategia iterativa**:

- Entrenar modelos base sin ajustar hiperparámetros.
- Comparar su rendimiento en un conjunto de métricas objetivo priorizando la clase positiva (TDAH = 1).
- Afinar sólo los modelos con potencial para la clase positiva.

Debido al **desbalance de clases** (minoría TDAH $\approx 11\%$), la métrica principal de evaluación y selección de umbral es la **curva Precisión–Recall** de la clase positiva y, en particular, el **F1-score** de TDAH. La **AUC-ROC** se reporta como *métrica complementaria* para resumir la *capacidad discriminativa global* del modelo, pero no guía la elección de umbral por su menor sensibilidad a la rareza de la clase. La *Accuracy* se incluye solo como referencia global, y se informan además **Sensibilidad/Recall** y **Especificidad** para describir el equilibrio entre detección y falsas alarmas. El detalle de métricas se encuentra en la Subsección 2.2.2, así como el detalle sobre AUC-ROC en la Subsección 2.2.2 y sobre la curva *Precision-Recall* en la Subsección 2.2.2.

Justificación del uso de F1 como métrica principal. El **F1-score** se define como la media armónica entre Precisión (PPV) y Recall (TPR) (Véase detalle de la fórmula en la Sección 2.2.2). Se elige F1 como métrica objetivo debido a que **penaliza fuertemente los desequilibrios** entre Precisión y Recall (la media armónica cae si una de las dos es baja), lo que evita soluciones que “ganan” sólo por una de ellas.

Además es **sensible a la clase positiva rara** y no se ve inflado por la mayoría (a diferencia de la *Accuracy*).

Por otro lado, está **alineado con el objetivo clínico** de *detectar* TDAH sin disparar las derivaciones: maximizar F1 obliga a mantener una Precisión razonablemente alta mientras se conserva un Recall clínicamente útil.

Finalmente es **consistente con la selección de umbral** en la curva *Precision-Recall*, donde se elige el punto operativo que maximiza F1 sobre la clase TDAH.¹

En el contexto clínico de cribado se prioriza reducir derivaciones innecesarias; por ello, el punto operativo se fija en la región de la curva *Precision-Recall* que maximiza el F1 de TDAH y mantiene una **Precisión** elevada, mientras se monitoriza la **Sensibilidad** para no desatender casos positivos.

6.4.1. Objetivo del entrenamiento y función de pérdida

El objetivo es aprender una función $X \rightarrow Y$ que asigne a cada observación la probabilidad de pertenecer a TDAH minimizando la discrepancia entre predicción y realidad (función de pérdida). Se exploran distintas familias de modelos y, por la utilidad en contexto clínico, el foco recae en enfoques *explicables*.

6.4.2. Técnicas utilizadas para el entrenamiento

Para evitar un modelo “optimista”, se separan los datos en **entrenamiento y validación**, reservando un **test** completamente independiente para la evaluación final. La noción general de

¹Cuando el coste clínico favorece más la Precisión o el Recall, puede usarse F_β (por ejemplo, $F_{0.5}$ si se prioriza más la Precisión o F_2 si se prioriza el Recall). En este trabajo se reporta F1 y se monitoriza explícitamente la Precisión mínima requerida para evitar sobre-derivaciones.

validación cruzada se revisa en el Capítulo 2, Sección 2.2.1.

- **Validación cruzada** (véase Sección 2.2.1; k -fold, estratificada en fases avanzadas) para reducir la varianza de una única partición. Todas las transformaciones que aprenden del dato (imputación, escalado, codificación/selección) se ajustan *exclusivamente* dentro de cada pliegue para evitar fuga de información.
- Monitorización de señales tempranas de **overfitting** para justificar ajustes de complejidad y/o de umbral.

6.4.3. Modelado y evolución del proceso

Para la clasificación de TDAH se evalúan varias alternativas; el desarrollo práctico se centra en un modelo explicable cuyo *comportamiento* y *reglas* sean comunicables en entornos clínico-educativos. A continuación, se detalla la cronología de ejecuciones e *hitos* logrados.

Primera ejecución

Se separan X (predictores) e y (objetivo original), y se dividieren X_{train} , y_{train} y X_{val} , y_{val} (`train_test_split`). Para compensar el desbalance, se usa `class_weight='balanced'` en el clasificador. El resultado inicial es **perfecto** en validación (Accuracy, Precision, Recall y F1 = 1.00; AUC-ROC = 1.00), una señal inequívoca de **sobreajuste**. Se detiene el proceso para revisar *EDA*, *feature engineering* y posibles fugas.

Detección y mitigación de sobreajuste. Se siguen dos líneas de trabajo:

- Revisión de correlaciones y redundancias (Spearman/Pearson) para evitar duplicidades que faciliten memorizar.
- Investigación de **data leakage**: variables demasiado próximas al diagnóstico/tratamiento. Las variables trampa identificadas se excluyen del entrenamiento.

Segunda ejecución (tras depuración)

Se detecta fuga en variables derivadas (`addtreat_clean_num`, `adhd_medicated_num`, `adhd_status_num`) con correlación casi perfecta con la etiqueta; se excluyen y se reentrena. El rendimiento pasa a ser **realista** y consistente, tal y como se puede observar en la Tabla 6.1. El análisis de importancias sitúa `k4q23_bin` como más influyente, seguida de `memorycond_bin` y `conducta_status_num`.

El diagrama del árbol correspondiente se encuentra en el Anexo F, Figura F.1.

Tabla 6.1: Comparativa de métricas antes y después del ajuste.

Métrica	Antes del ajuste	Después del ajuste
Accuracy	0.91	0.92
F1 TDAH	0.67	0.69
F1-macro	0.81	0.82

Glosario rápido. *F1-TDAH*: F1 de la clase minoritaria. *F1-macro*: media aritmética de los F1 por clase (con `class_weight='balanced'` se favorece el equilibrio).

Primera revisión

En la reunión de seguimiento se amplía el análisis de correlaciones y se identifica otro grupo de variables con fuga de información, en particular:

- **k4q23_bin**: medicación para emociones, concentración o comportamiento.
- **memorycond_bin**: dificultad seria para concentrarse/recordar/decidir por condición física/-mental/emocional.

Se excluyen para priorizar rasgos *indirectos* y mejorar la validez.

Modelo a entrenar. Se acuerda continuar con un clasificador explicable por su interpretabilidad y su capacidad para jerarquizar variables por importancia.

Nuevas variables incorporadas. **makefriend_num** (dificultad para hacer amigos) y **k7q84_r** (constancia para terminar tareas) se incorporan por coherencia clínica y señal predictiva.

Experimento de sensibilidad. Se evalúa el efecto de eliminar **memorycond_bin**. *Con memorycond_bin*, mejoran Precision y F1 de TDAH; *sin* ella, el modelo usa más diversidad de variables y gana interpretabilidad con sacrificio ligero de rendimiento. Esta decisión se basa en priorizar la interpretabilidad y menor dependencia de una única variable.

Tercera ejecución

Se reentrena con el conjunto depurado. El diagrama actualizado se encuentra en el Anexo F, Figura F.2. El rendimiento alcanzado tras esta depuración se observa en la Tabla 6.2.

Tabla 6.2: Interpretación del rendimiento — 3.º entrenamiento

Métrica	Clase 0 (No TDAH)	Clase 1 (TDAH)
Precisión	0.98	0.43
Recall	0.86	0.83
F1-score	0.91	0.56
Accuracy global		0.86
Especificidad (Clase 0)		0.8592

De la lectura de la Tabla 6.2 se concluye que el modelo acierta con alta pureza la clase negativa (No TDAH) y recupera muchos casos de la clase positiva, lo que supone un avance en el objetivo del entrenamiento orientado a captar el máximo número de casos positivos. Sin embargo, se observa en la *Precisión* de TDAH que el número de falsos positivos resulta moderadamente elevado.

Por tanto, en el siguiente paso, se establece retirar la variable **sc_k2q22_bin** para forzar un uso más equilibrado de variables conductuales y educativas.

Cuarta ejecución

En el cuarto entrenamiento del modelo, se filtra la variable detectada como posible fuga de información indicada anteriormente (**sc_k2q22_bin**) (variable que indica si el niño tiene algún problema conductual, emocional o de desarrollo que requiera tratamiento o terapia).

Adicionalmente, para mejorar la robustez de la evaluación y asegurar que el rendimiento del modelo no dependa de una única partición de datos, se **incorpora una validación cruzada estratificada con nsplits igual a 5 y un random_state de 42**.

El objetivo es simular de una forma más realista el comportamiento del modelo con nuevos datos, mitigando el overfitting.

Las variables más influyentes, catalogadas por el Árbol de Decisión tras el reentrenamiento, en la detección del TDAH se muestran en la Figura 6.9 (consúltese Anexo B para ver significado de las variables)

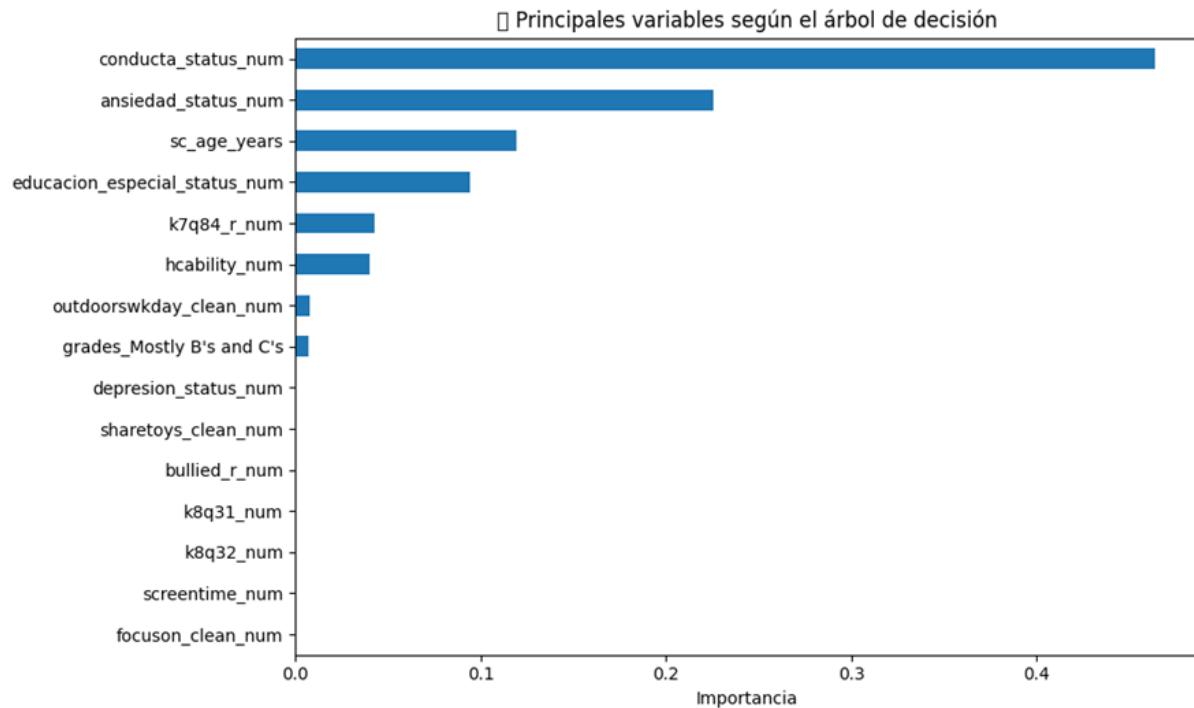


Figura 6.9: Principales variables (criterio Gini).

Los resultados de las métricas objetivo de esta ejecución se observan en la Tabla 6.3:

Tabla 6.3: Métricas cuarta ejecución.

Métrica	Clase 0 (No TDAH)	Clase 1 (TDAH)
Precisión	0.98	0.40
Recall	0.84	0.83
F1-score	0.90	0.54
Accuracy global	0.84	
Especificidad (Clase 0)	0.8412	

El diagrama correspondiente está en el Anexo F, Figura F.3. Tras esta fase, la estructura se presenta más clara y clínicamente coherente, sin “trampas”. La variable del nodo raíz pasa a `conducta_status_num`.

Refinamientos. Para reducir falsos positivos y mejorar la precisión en TDAH, se **eleva el umbral de decisión**. El *AUC-ROC* queda en **0.9068**, confirmando buena capacidad discriminativa. También se prueban variantes (calibración, hiperparámetros, otras familias), pero se mantiene el clasificador explicable por su claridad e interpretabilidad. La evolución de las métricas según el umbral de decisión se observa en la Figura 6.10.

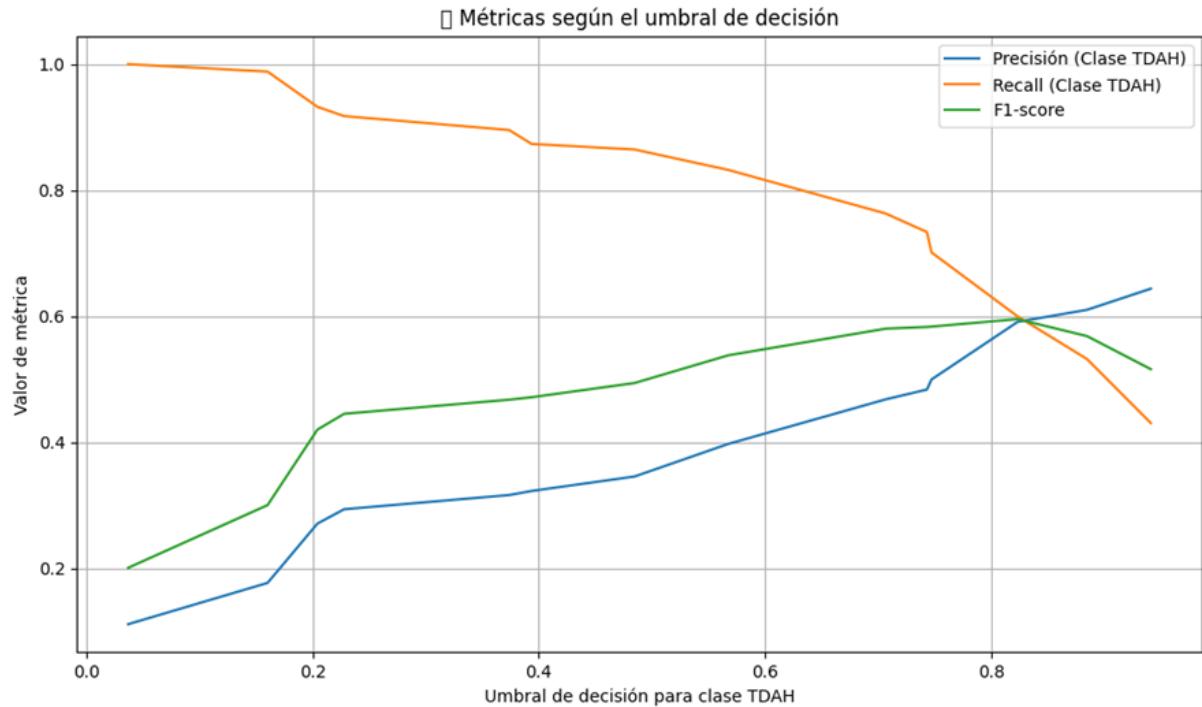


Figura 6.10: Métricas clave según el umbral de decisión.

Tras observar la Figura 6.10, el desplazamiento del umbral de decisión τ modifica el equilibrio entre *precisión* (PPV) y *recall* (TPR): al aumentar τ , el modelo se vuelve más estricto, incrementando la precisión a costa de reducir el recall. La *F1* describe un máximo en torno al entorno operativo seleccionado.

Dado que el objetivo clínico es minimizar derivaciones innecesarias, se prioriza la precisión y se adopta un umbral relativamente alto (véase la Tabla 6.4). Con este punto operativo, el patrón de aciertos y errores queda recogido en la matriz de confusión de la Tabla 6.5. En conjunto, el sistema favorece una especificidad elevada con una sensibilidad moderada, alineándose con un uso como herramienta de apoyo/cribado donde se busca no sobrecargar el circuito asistencial. Umbrales más bajos aumentarían la sensibilidad, pero al precio de más falsos positivos.

Tabla 6.4: Métricas en validación para el umbral seleccionado. Valores redondeados a 2 decimales.

τ	PPV (Precisión)	TPR (Recall)	F1	Especificidad	Accuracy
0.82	0.59	0.60	0.60	0.95	0.91

Conclusión Tras esta ejecución el modelo queda *conservador*: reduce falsos positivos a costa de aceptar una pérdida moderada de positivos reales, decisión coherente con el objetivo clínico declarado.

Tabla 6.5: Matriz de confusión en validación para $\tau = 0.82$.

	Predicho 0	Predicho 1
Real 0	5114	282
Real 1	272	408

El modelo funciona como un detector que suena ante señales fuertes indicadoras de TDAH (en contexto académico, social y conductual), da menos falsas alarmas pero algunas señales reales más débiles no las capta, por lo cual se continúa refinando el modelo.

Dado que la prioridad es la **precisión clínica** para derivar a un especialista para terminar de confirmar el diagnóstico, se determina que es preferible usar un umbral alto; por lo tanto, se adopta el umbral que optimiza el F1-score: 0.82

6.4.4. Hiperparámetros, calibración y umbral de decisión

En esta fase, se amplía el *grid* de hiperparámetros, se compara la optimización por *F1* y por *Recall*, y se añade **calibración** previa al ajuste de umbral. Con el umbral óptimo por *F1* ($\tau \approx 0.35$), los resultados en validación se observan en la Tabla 6.6.

Tabla 6.6: Informe de clasificación (calibrado + umbral). Umbral óptimo calibrado: $\tau = 0.37$.

Clase	Precisión	Recall	F1-score	Support
0 (No TDAH)	0.96	0.94	0.95	5396
1 (TDAH)	0.57	0.65	0.61	680
Accuracy global	0.91			6076
Macro avg	0.76	0.80	0.78	6076
Weighted avg	0.91	0.91	0.91	6076

Selección de variables, calibración y umbral. Para el modelo definitivo se priorizan el rendimiento y la utilidad clínica/interpretabilidad en el cribado de TDAH. Primero, se lleva a cabo una selección de variables a partir de la importancia estimada por un árbol preliminar entrenado sobre datos balanceados mediante *Random UnderSampling*. Después, se aplica un procedimiento iterativo de convergencia: en cada iteración se reentrena el árbol con las variables candidatas y se restringen los conjuntos de entrenamiento y validación a las características efectivamente utilizadas, repitiendo hasta estabilizar un subconjunto consistente y explicativo.

A continuación, se ajustan los hiperparámetros con *GridSearchCV* y validación cruzada estratificada (5 particiones), optimizando el *recall* de la clase positiva para minimizar los falsos negativos. El espacio de búsqueda incluye parámetros clave del árbol (profundidad máxima, mínimos de muestras por división/hoja y poda mínima).

Una vez fijada la estructura, se calibran las probabilidades con *CalibratedClassifierCV* y validación cruzada, entrenando sobre los datos originales (sin *undersampling*) para corregir la tendencia de los árboles a producir probabilidades mal ajustadas y así obtener estimaciones más fiables para la toma de decisiones clínicas.

Por último, el umbral de decisión no se mantiene en 0.5: se selecciona en función de la curva *precision-recall*, calculando precisión, *recall* y F1 para cada τ y eligiendo el que maximiza la F1 de la clase TDAH. Este criterio logra un equilibrio entre detección y falsos positivos, priorizando la sensibilidad del sistema: un falso positivo implica una revisión adicional, mientras que un falso negativo puede dejar a un menor sin diagnóstico ni tratamiento.

6.4.5. Árbol de decisión. Interpretación y versión definitiva

Tras ampliar el *grid* de hiperparámetros, calibrar el *Decision Tree* y optimizar el umbral mediante F1-score, el rendimiento final se resume en la Tabla 6.7. La evolución de las métricas en función del umbral puede consultarse en la Figura 6.11.

Tabla 6.7: Resumen de métricas del entrenamiento final tras calibración y selección de umbral.

Métrica	Valor
Umbral óptimo (calibrado)	0.37
Precisión (TDAH = 1)	0.54
Recall (TDAH = 1)	0.68
F1-score (TDAH = 1)	0.61
Accuracy	0.9057
Sensibilidad (Recall, TDAH)	0.68
Especificidad (No TDAH)	0.9277
AUC-ROC	0.9126

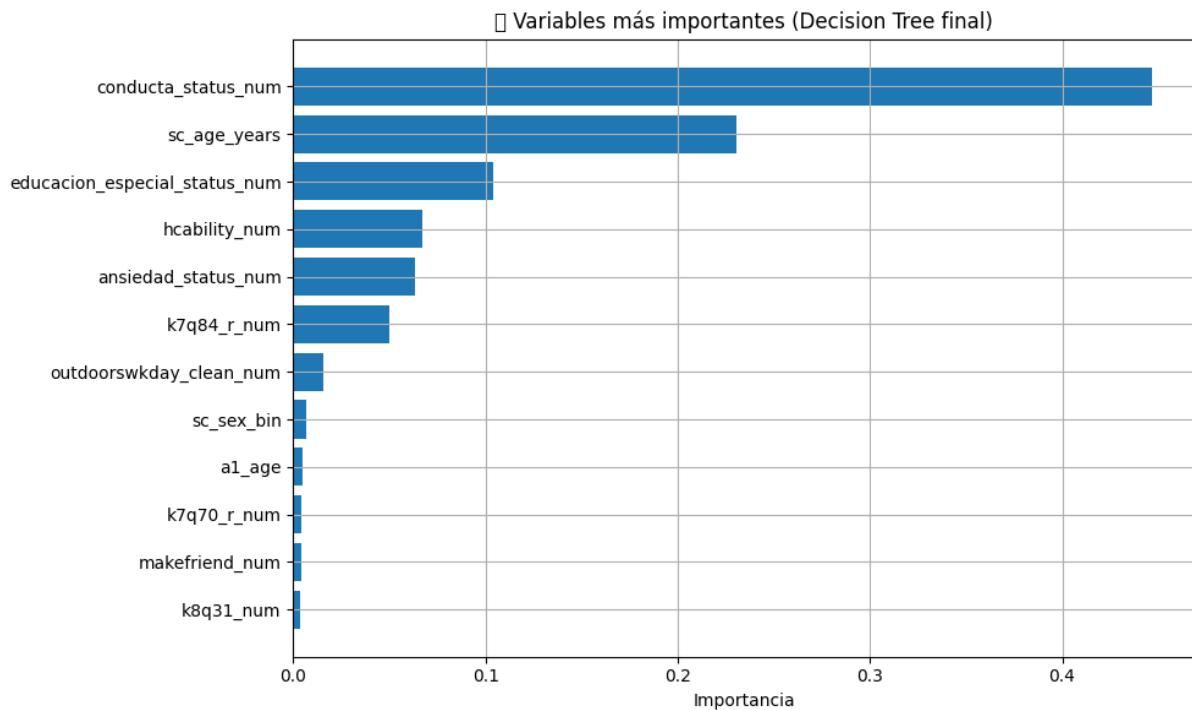


Figura 6.11: Métricas clave en función del umbral de decisión.

El reentrenamiento identifica predictores especialmente informativos y coherentes con la literatura clínica (señales conductuales, funcionales y contextuales):

- **conducta_status_num:** principal contribución; problemas de conducta (impulsividad, desobediencia) estrechamente ligados a TDAH.
- **sc_age_years:** edad del menor; la expresión clínica y el umbral diagnóstico dependen del desarrollo.
- **educacion_especial_status_num:** acceso a apoyos de educación especial; proxy de necesidades asociadas a dificultades atencionales/conductuales.
- **hcability_num:** interferencia de la salud en actividades propias de la edad; impacto funcional cotidiano.

- **ansiedad_status_num**: comorbilidad frecuente con valor discriminativo adicional.
- **k7q84_r**: finalización de tareas; dificultades nucleares de atención y persistencia.
- **outdoorswkday_clean_num**: tiempo al aire libre entre semana; relacionado con autorregulación y socialización.
- **birth_yr, sc_sex_bin**: año de nacimiento y sexo; diferencias por edad y mayor prevalencia en varones.
- **k7q70_r**: frecuencia de discusiones; marcador de impulsividad/baja tolerancia a la frustración.
- **makefriend_num**: facilidad para hacer amigos; dificultades sociales frecuentes en perfiles con TDAH/conducta.
- **k8q31_num**: dificultad percibida de los cuidados; síntesis del impacto conductual en la dinámica familiar.
- **a1_age**: edad del adulto (madre), modulador contextual.

El diagrama final del árbol se encuentra en el Anexo F, Figura F.4.

Interpretación del árbol de decisión.

El clasificador (impureza de Gini) presenta trayectorias transparentes y clínicamente plausibles.

Nodo raíz. El árbol pivota sobre **conducta_status_num** (umbral ≈ 0.5): valores > 0.5 abren rutas con mayor probabilidad de TDAH; valores ≤ 0.5 conducen mayoritariamente a No TDAH.

Rama sin problemas de conducta. Con $\text{conducta_status_num} \leq 0.5$, predomina No TDAH, reforzado por: (i) menor edad ($\leq 6-7$ años), (ii) ausencia de ansiedad y (iii) cuidadores que no reportan dificultad en el cuidado (**k8q31** bajo). Los nodos terminales alcanzan alta pureza hacia No TDAH.

Rama con problemas de conducta. Si $\text{conducta_status_num} > 0.5$, la distribución se inclina hacia TDAH:

- El **tiempo al aire libre entre semana** se vuelve clave: muchas horas fuera, dentro del subgrupo con conducta problemática, es un **marcador notable** de TDAH (en ese nodo, $\sim 94\%$ positivos).
- Con menor tiempo al aire libre (< 3 h/día), la **edad** modula el riesgo: a partir de ~ 3 años aumenta la probabilidad de TDAH, aunque con impureza moderada (Gini ≈ 0.407).
- En los más pequeños, la **edad materna** introduce un matiz adicional; señal útil pero basada en pocos casos (conclusiones poblacionales frágiles).

Zona intermedia (incertidumbre). **Subrama central izquierda:** en menores sin otras condiciones de salud (`hcability = 0`), la triada *persistencia para terminar tareas* → *ansiedad* → *sexo* resuelve la ambigüedad. Una **baja persistencia** es **señal dominante de riesgo**. Con **alta persistencia**, el **diagnóstico de ansiedad** decide: con ansiedad se favorece TDAH; sin ansiedad predomina No TDAH, especialmente en **niñas**. En niños con alta persistencia y sin ansiedad surge un nodo cercano al 50–50.

Subrama central derecha: vuelve a primar la **persistencia en tareas**. Con persistencia inferior a “siempre o normalmente”, el árbol asume TDAH casi siempre; sólo la combinación de *sin ansiedad* y *dificultades para hacer amigos* inclina a No TDAH. En el resto, predomina TDAH.

Lectura global. El árbol reproduce patrones clínicos consistentes: la **conducta** actúa como disparador principal; **ansiedad** y **persistencia en tareas** ajustan el riesgo; y marcadores **funcionales/educativos** (interferencia en la vida diaria, apoyos de educación especial, sociabilidad) resuelven ambigüedades.

La mayor pureza hacia TDAH aparece con la confluencia de conductas problemáticas, baja finalización de tareas y ansiedad. Mientras la tendencia hacia No TDAH se da con ausencia de conducta problemática, baja interferencia funcional y cuidado percibido como fácil.

6.5. Evaluación del modelo

Esta fase se centró en verificar la generalización, comparar escenarios de variables y afinar decisiones de umbral y complejidad del clasificador, manteniendo constante el *pipeline* de preprocesamiento. Los focos fueron:

1. Evaluación en *test* independiente.
2. Control de *logical skips* en la imputación.
3. Comparación experimental de conjuntos de variables.
4. Análisis de *pruning*.

El objetivo fue confirmar un rendimiento estable en datos no vistos y consolidar un modelo útil y sostenible en producción.

6.5.1. Evaluación del modelo sobre el conjunto de *test*

Para estimar el desempeño fuera de muestra, se aplicó el mismo *pipeline* del entrenamiento (mapeos, imputación, tratamiento de *logical skips*, ingeniería de variables y filtrado) sobre un *holdout* independiente `test_df`. Con el modelo calibrado se obtuvieron predicciones y métricas por clase. Los resultados completos se mostraron en la Tabla 6.8 y la comparativa *train/test* en la Tabla 6.9.

En síntesis, el modelo mantuvo consistencia entre validación y *test*, con alta especificidad en la clase No TDAH y sensibilidad moderada en TDAH, sin señales de sobreajuste. Esta estabilidad reforzó la robustez del *pipeline* y la idoneidad del calibrado para la toma de decisiones clínicas.

Revisión de *Logical Skips* imputados. Durante la imputación KNN, el valor -1 (salto lógico) se propagó como vecino válido, introduciendo ruido en agregaciones y codificaciones. Se mitigó con:

- Exclusión de -1 del vecindario
- Validación por dominios de valores permitidos
- *Fallback* controlado (moda en no numéricas y 0 en numéricas).

Este ajuste estabilizó la calidad de las características y, por extensión, las métricas de evaluación (véase Tabla 6.8).

Métrica	Clase 0 (No TDAH)	Clase 1 (TDAH)
Precisión	0.96	0.54
Recall	0.93	0.70
F1-score	0.94	0.61
Support	6836	859
Accuracy global	0.9084	
Especificidad (Clase 0)	0.9393	

Tabla 6.8: Rendimiento del modelo en datos de *test*.

Clase	Partición	Precisión	Recall	F1-Score
0 (No TDAH)	<i>Train</i>	0.9588	0.9277	0.9430
0 (No TDAH)	<i>Test</i>	0.9606	0.9264	0.9432
1 (TDAH)	<i>Train</i>	0.5439	0.6838	0.6059
1 (TDAH)	<i>Test</i>	0.5436	0.6973	0.6109

Tabla 6.9: Comparativa por clase entre entrenamiento (validación) y *test*.

6.5.2. Comparación del rendimiento por conjuntos de variables

Se evaluaron tres escenarios manteniendo fijo el clasificador (árbol calibrado) y el *pipeline*, para aislar el efecto del conjunto de variables:

- **Escenario 1:** `train_df_model` (51 variables)
- **Escenario 2:** `train_df_model_all` (87 variables)
- **Escenario 3:** `expert_train_df_model_coded` (24 variables de literatura/criterio experto)

La Tabla 6.10 resumió las métricas principales por escenario. A continuación, se mostraron las comparativas gráficas (Figuras 6.12, 6.13, 6.14, 6.15, 6.16, 6.17 y 6.18) para F1, *Recall*, *Precision*, AUC-ROC, *Specificity* y *Accuracy*, así como un radar de rendimiento agregado.

Tabla 6.10: Comparativa de métricas por selección de variables.

Conjunto de variables	Num variables	AUC (0)	Accu (0)	Spec (0)	Prec (1)	Rec (1)	F1 (1)
Nuestras variables	51	0.9316	0.9144	0.9375	0.5959	0.7309	0.6565
Todas las variables	87	0.9316	0.9144	0.9375	0.5959	0.7309	0.6565
Variables del experto	24	0.8188	0.8688	0.9157	0.4262	0.4971	0.4589

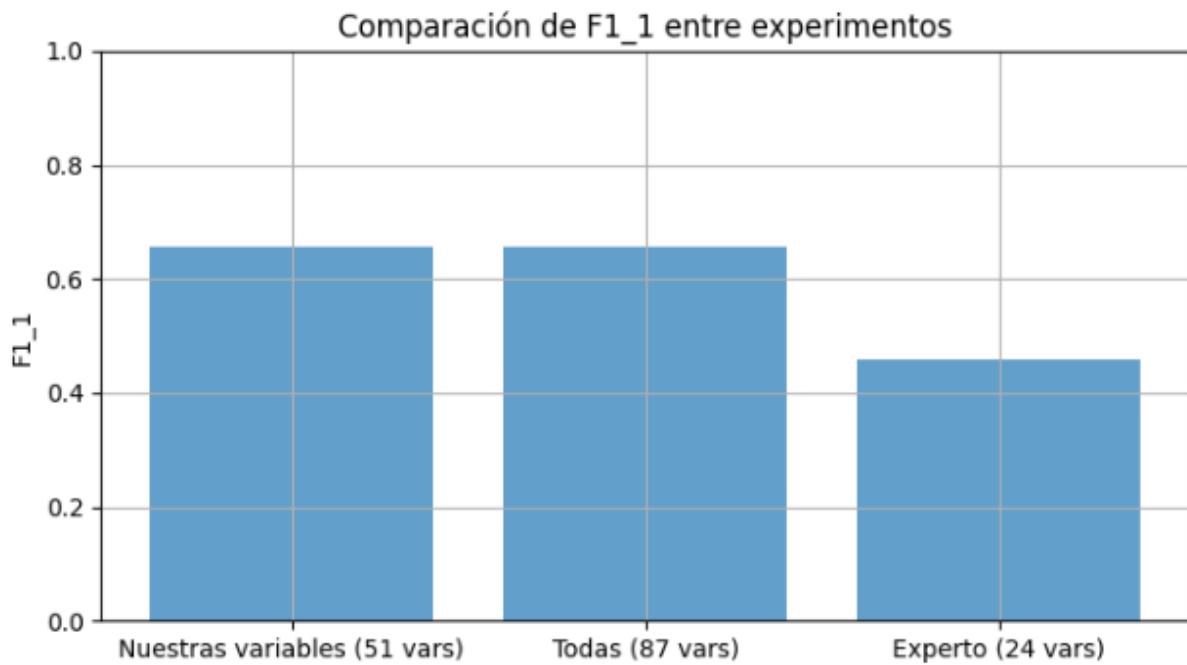


Figura 6.12: Comparativa de F1-score entre los tres conjuntos.

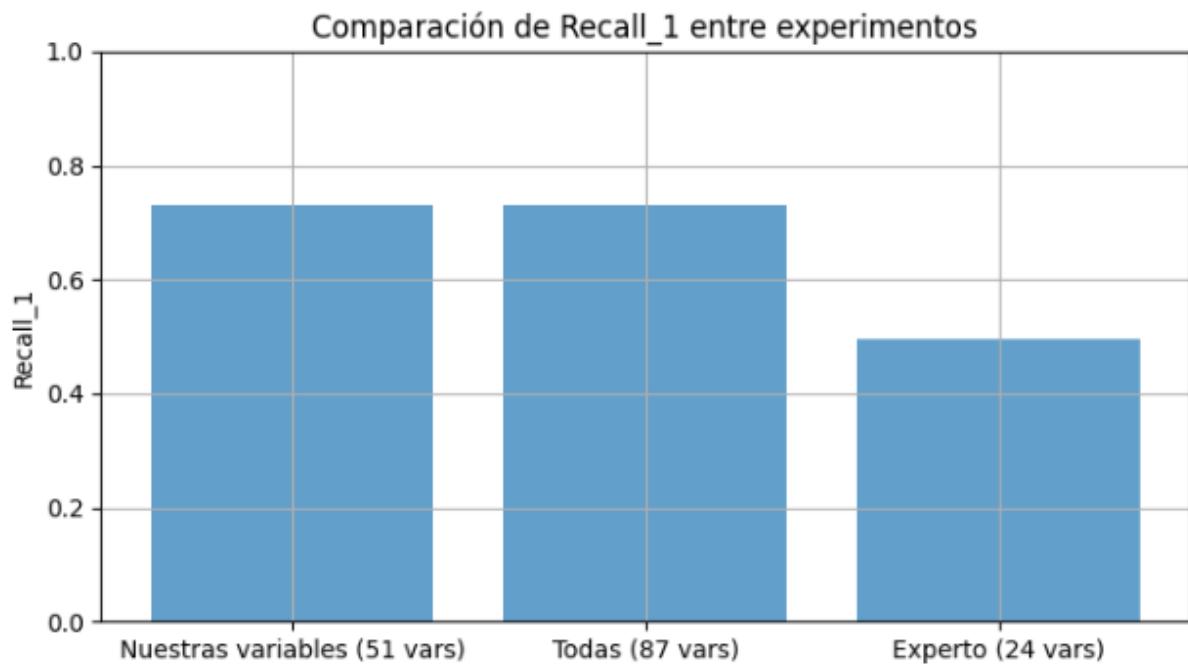


Figura 6.13: Comparativa de *Recall* (TDAH) entre los tres conjuntos.

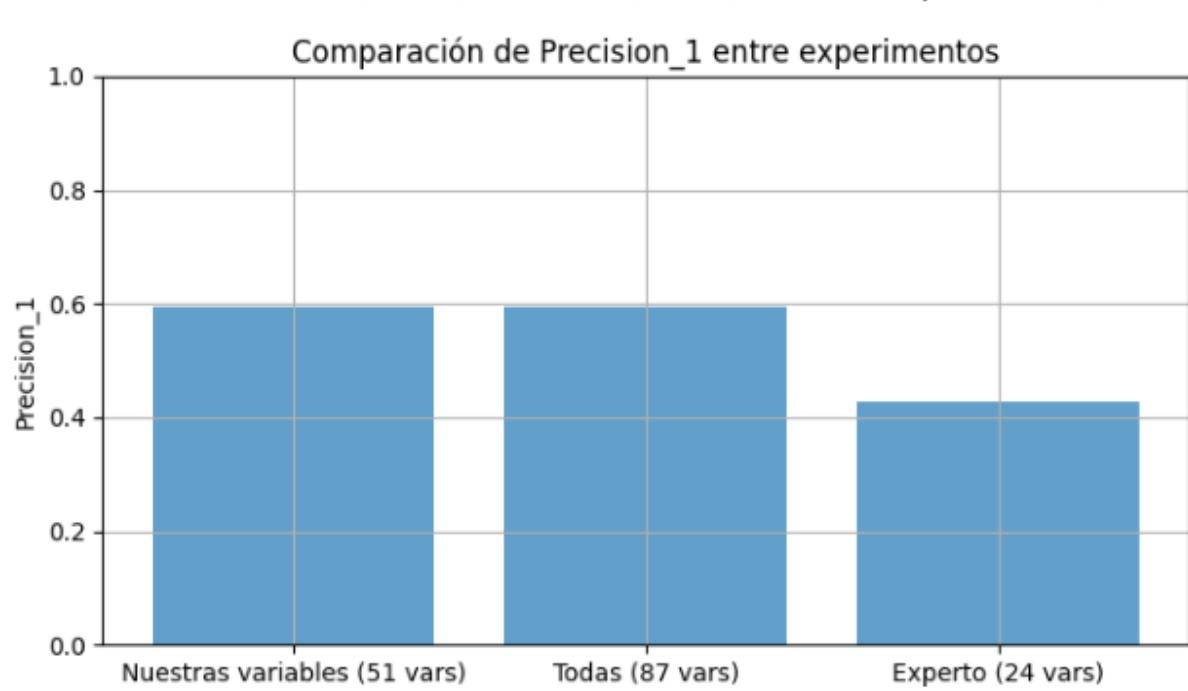


Figura 6.14: Comparativa de *Precision* (TDAH) entre los tres conjuntos.

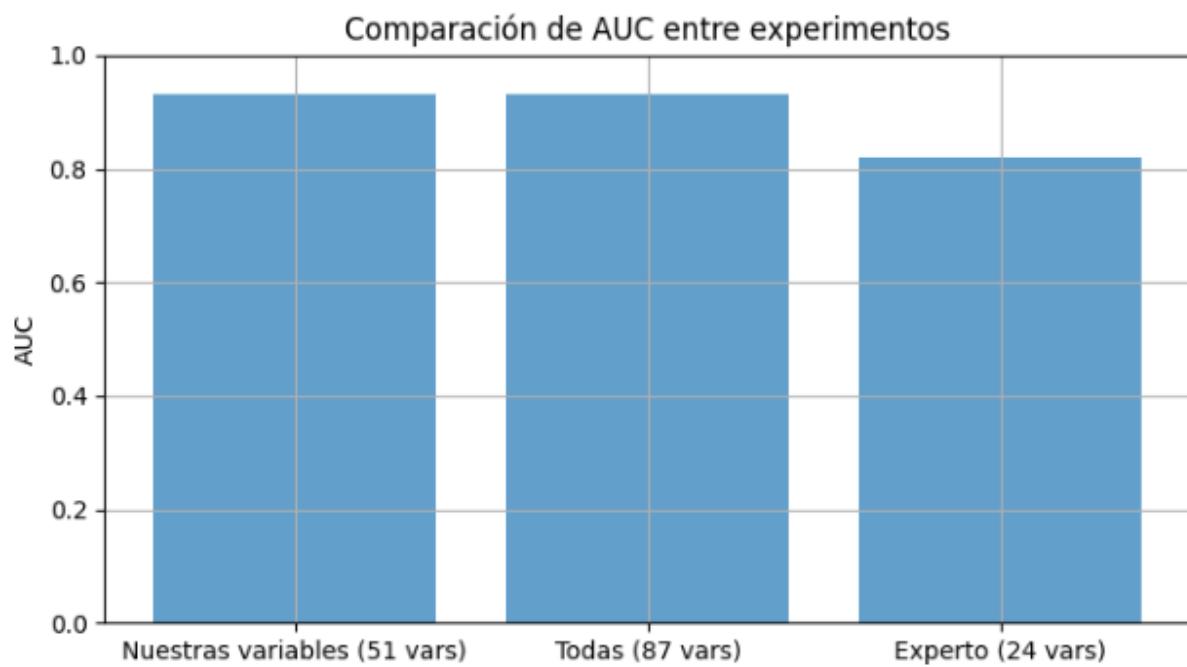


Figura 6.15: Comparativa de AUC-ROC entre los tres conjuntos.

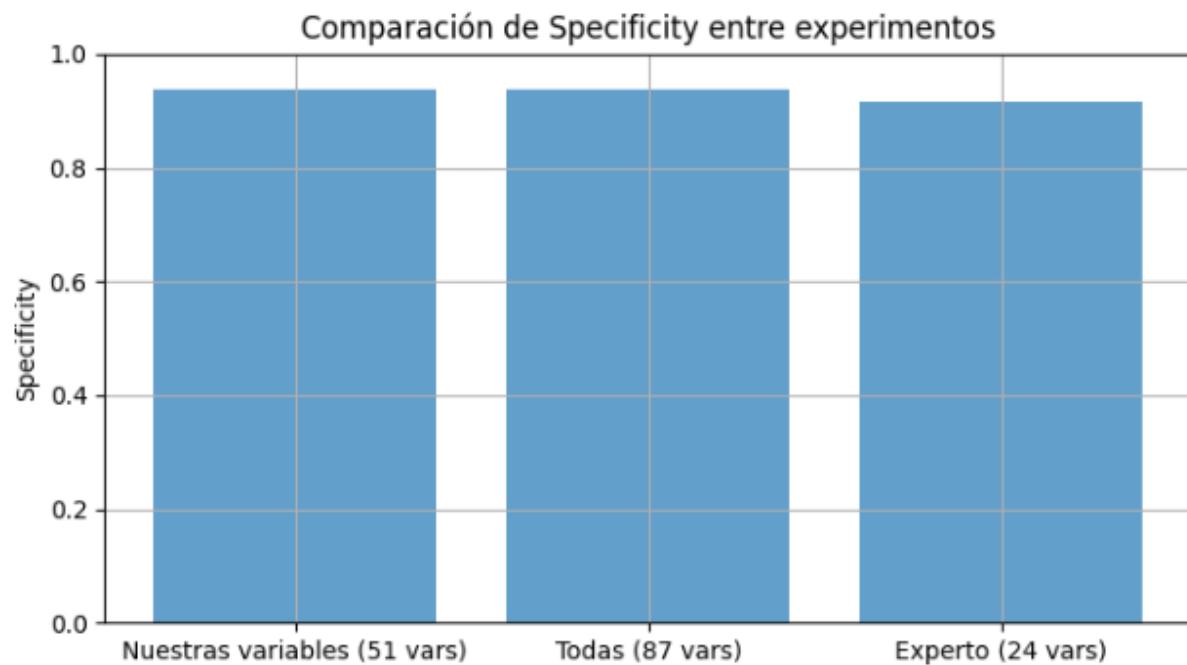


Figura 6.16: Comparativa de *Specificity* entre los tres conjuntos.

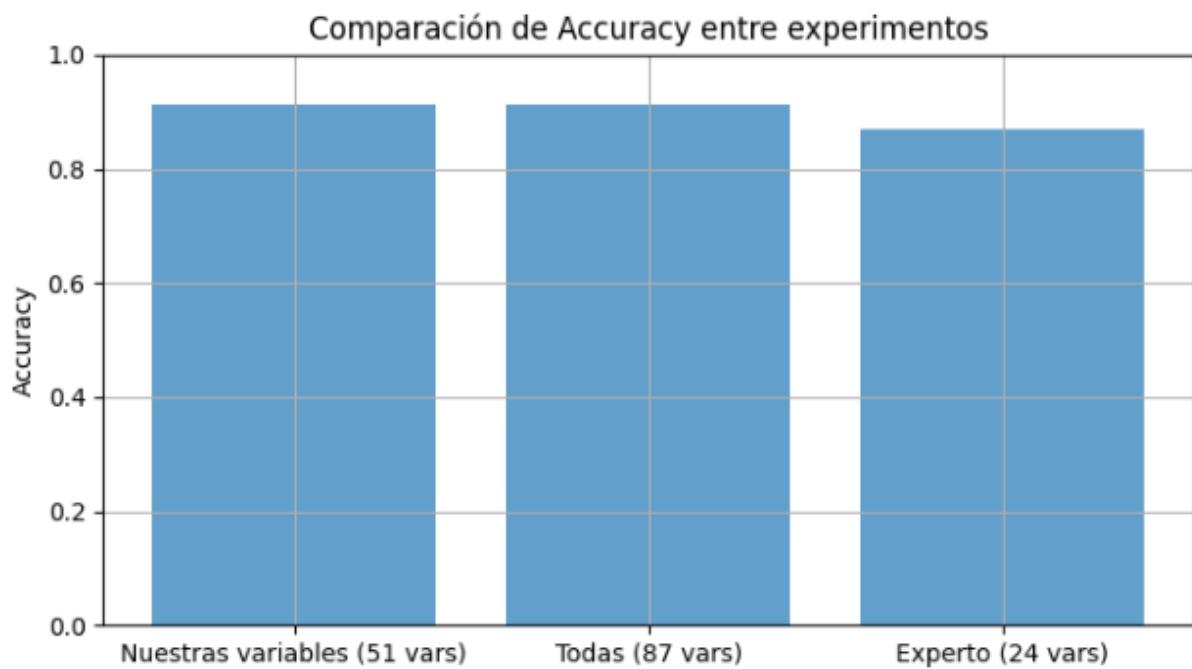


Figura 6.17: Comparativa de *Accuracy* entre los tres conjuntos.

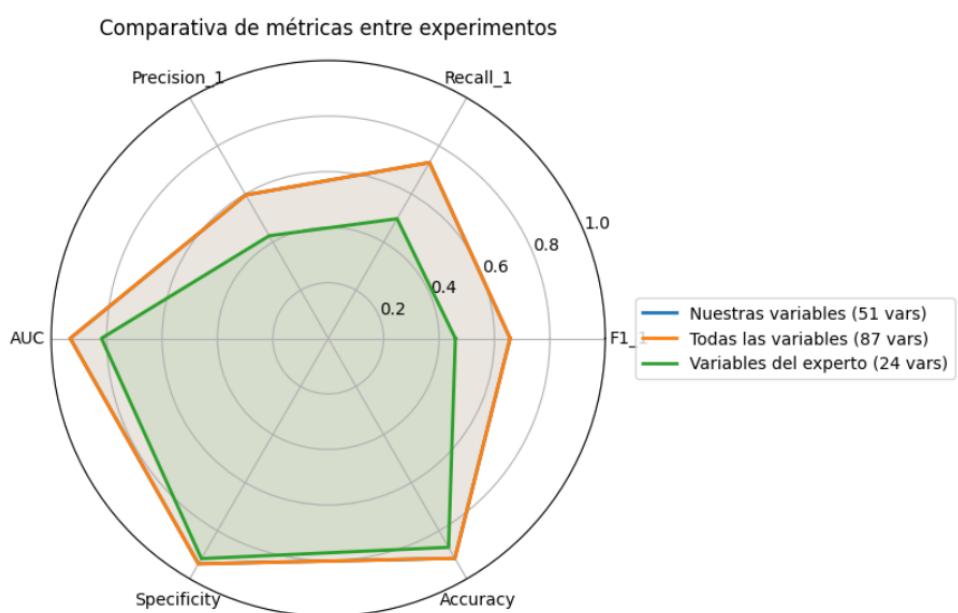


Figura 6.18: Gráfico radar de rendimiento agregado por conjunto.

Conclusión

El conjunto de 51 variables rindió igual que el de 87 (Tabla 6.10, Figuras 6.12–6.17), con la ventaja de haber resultado más compacto e interpretable. El conjunto experto perdió capacidad (menor F1, AUC y *recall* en TDAH), lo que indicó que solo la selección manual no capturó toda la complejidad del problema. La evidencia favoreció una estrategia híbrida: punto de partida experto más depuración empírico–estadística (EDA y validación).

6.5.3. Análisis de *pruning* del *Decision Tree*

Se exploró la poda por complejidad del coste (*ccp_alpha*) para reducir la complejidad del árbol sin sacrificar desempeño. El barrido de *ccp_alpha* mostró una región óptima entre 0.0005 y 0.001, privilegiando la F1 con un número de nodos sensiblemente menor (véase Figura 6.19 y Tabla 6.11). Se seleccionó *ccp_alpha* = 0.001 por ofrecer mejor F1 con menos nodos, respaldando que la poda del modelo final resultó adecuada (y aún optimizable).

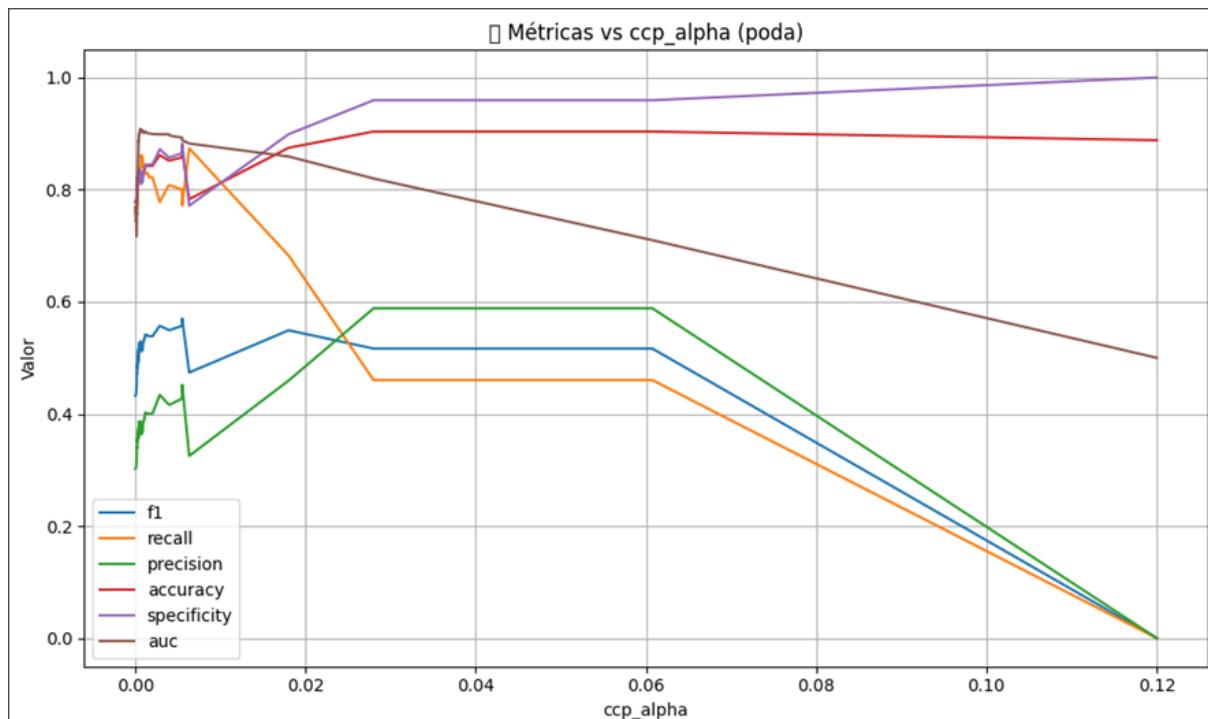


Figura 6.19: Evolución de métricas frente a *ccp_alpha*.

Tabla 6.11: Resultados por valor de *ccp_alpha*.

<i>ccp_alpha</i>	F1	Recall	Precision	Accuracy	Specificity	AUC	Nodos
0.0005	0.5215	0.837	0.379	0.828	0.827	0.899	139
0.001	0.5292	0.838	0.387	0.833	0.832	0.900	39

Conclusión de evaluación. El modelo presentó generalización sólida en *test*, alto control de falsos positivos (especificidad elevada) y sensibilidad razonable en TDAH tras calibración y selección de umbral. El conjunto de 51 variables ofreció el mejor equilibrio entre rendimiento e interpretabilidad sin coste adicional frente al conjunto completo. La poda con *ccp_alpha*

= 0.001 redujo complejidad manteniendo métricas, lo que favoreció despliegue y mantenimiento. En conjunto, la evidencia apoyó el uso del modelo como herramienta de apoyo al cribado, con un sesgo controlado hacia minimizar sobre-derivaciones sin desatender la detección de casos positivos.

6.6. Software, entornos y control de versiones

Con el fin de asegurar trazabilidad y reproducibilidad en la fase de **ciencia de datos**, a continuación se documentan plataformas, programas externos, librerías y el control de versiones utilizados.

6.6.1. Plataformas de trabajo

Tabla 6.12: Plataformas utilizadas en ciencia de datos.

Plataforma	Versión	Propósito
Google Colab	2024-04-15	Ejecución de notebooks, EDA y experimentos reproducibles.

6.6.2. Programas externos

Tabla 6.13: Programas externos empleados en la preparación del dataset.

Programa	Versión	Propósito
Stata	16.1	Decodificación del fichero .dta (etiquetas y categorías) y comprobaciones puntuales.

6.6.3. Entorno base y librerías de ciencia de datos

Tabla 6.14: Entorno y librerías principales empleadas en ciencia de datos.

Software	Versión	Propósito
Python	3.12.12	Lenguaje para EDA, preprocesado y entrenamiento de modelos.
Linux	6.12.12	Sistema base del entorno de ejecución.
Librería	Versión	Propósito
pandas	2.2.2	Manipulación y limpieza de datos; E/S.
numpy	2.0.2	Cálculo numérico y estructuras n-dimensionales.
matplotlib	3.10.0	Visualizaciones básicas y personalización.
seaborn	0.13.2	Visualización estadística de alto nivel.
imblearn	0.14.0	Técnicas para datos desbalanceados.
joblib	1.5.2	Serialización de artefactos (modelos/pipelines).
scikit-learn	1.6.1	Modelado ML, validación y métricas.

6.6.4. Control de versiones

Tabla 6.15: Herramientas de control de versiones en ciencia de datos.

Herramienta	Versión	Propósito
Git (repositorio en GitHub)	2.46.0	Control de versiones distribuido e integración con repositorio remoto (véase Tabla 6.16).

Repositorio del proyecto

En la siguiente Figura 6.16 se presenta el repositorio público utilizado, donde se encuentran los notebooks utilizados en cada fase del proceso de Ciencia de Datos.

Tabla 6.16: Repositorio GitHub del proyecto (URL pública).

Elemento	Contenido
URL	https://github.com/dcarmor99/TDAHTool/tree/develop

Carpeta de notebooks Dentro de la carpeta **notebooks** en la raíz del repositorio se encuentran los notebooks que componen el flujo analítico y pipeline del proceso de Ciencia de Datos (Tabla 6.17).

Tabla 6.17: Resumen funcional de los notebooks del repositorio.

Notebook	Descripción breve
0_data_loading_and_split.ipynb	Carga de datos, exploración inicial y partición en train/test.
1_data_clearing_and_preprocessing.ipynb	Limpieza, tratamiento de nulos y preprocesado de variables.
2_eda_exploratory_analysis.ipynb	Ánalisis exploratorio, visualizaciones y estadísticas descriptivas.
3_model_training_and_validation.ipynb	Ajustes finales, validación y exportación/correción del modelo .pkl.

Capítulo 7

Desarrollo y Despliegue del Sistema

En este capítulo se aborda el proceso integral de desarrollo y la implementación técnica de la aplicación TDAHTool. Se comienza estableciendo el objetivo y el alcance del sistema, centrado en la predicción individual. A continuación, se detalla el diseño de la solución, especificando la arquitectura de software elegida y los requisitos funcionales y no funcionales.

La parte central del capítulo describe la implementación del backend (API con FastAPI) y del frontend (interfaz con Streamlit), incluyendo la lógica de inferencia y la explicación de la ruta del modelo. Posteriormente, se presenta la validación funcional de la interfaz mediante casos de uso concretos. Finalmente, se exponen las metodologías de despliegue (Docker) y las buenas prácticas de ingeniería del software que guían el proyecto.

Objetivo y alcance

El objetivo de la aplicación es apoyar el diagnóstico de TDAH mediante un modelo de ML accesible a través de una interfaz web sencilla. Se plantea una aplicación web simple y orientada al usuario, que permite calcular la probabilidad de TDAH a partir de un conjunto acotado de variables (las más relevantes) y ofrece además una vista para “jugar” con los parámetros y comprender cómo cambia la predicción.

Para reducir tiempos y riesgos, se opta por una **predicción individual** en lugar de en lote: es decir, el usuario rellena un formulario con las 12 variables clave del modelo y la aplicación devuelve los porcentajes de probabilidad de la clase predictora (probabilidad TDAH / probabilidad No TDAH), junto con las métricas globales del modelo usado para contextualizar el rendimiento.

Como futuras actualizaciones de esta aplicación se proponen algunas ideas:

- **Carga de CSV y scoring en batch**, permitiendo realizar predicciones en conjuntos de múltiples pacientes.
- **Data Profiling automatizado del dataset** subido para extraer las estadísticas más relevantes para el estudio de datos de la población estudiada.
- **Reentrenamiento desde la aplicación**.

7.1. Diseño del sistema

En esta sección se establece el diseño técnico de la aplicación. Se comienza describiendo la estructura de carpetas del proyecto, que sienta las bases para la separación de responsabilidades

(RNF2). A continuación, se definen los requisitos funcionales y no funcionales que guían el desarrollo, especificando qué debe hacer el sistema (RF) y cómo debe hacerlo (RNF). Finalmente, se presenta la arquitectura del sistema, ilustrando el flujo de datos y la interacción entre los componentes clave.

7.1.1. Estructura de carpetas

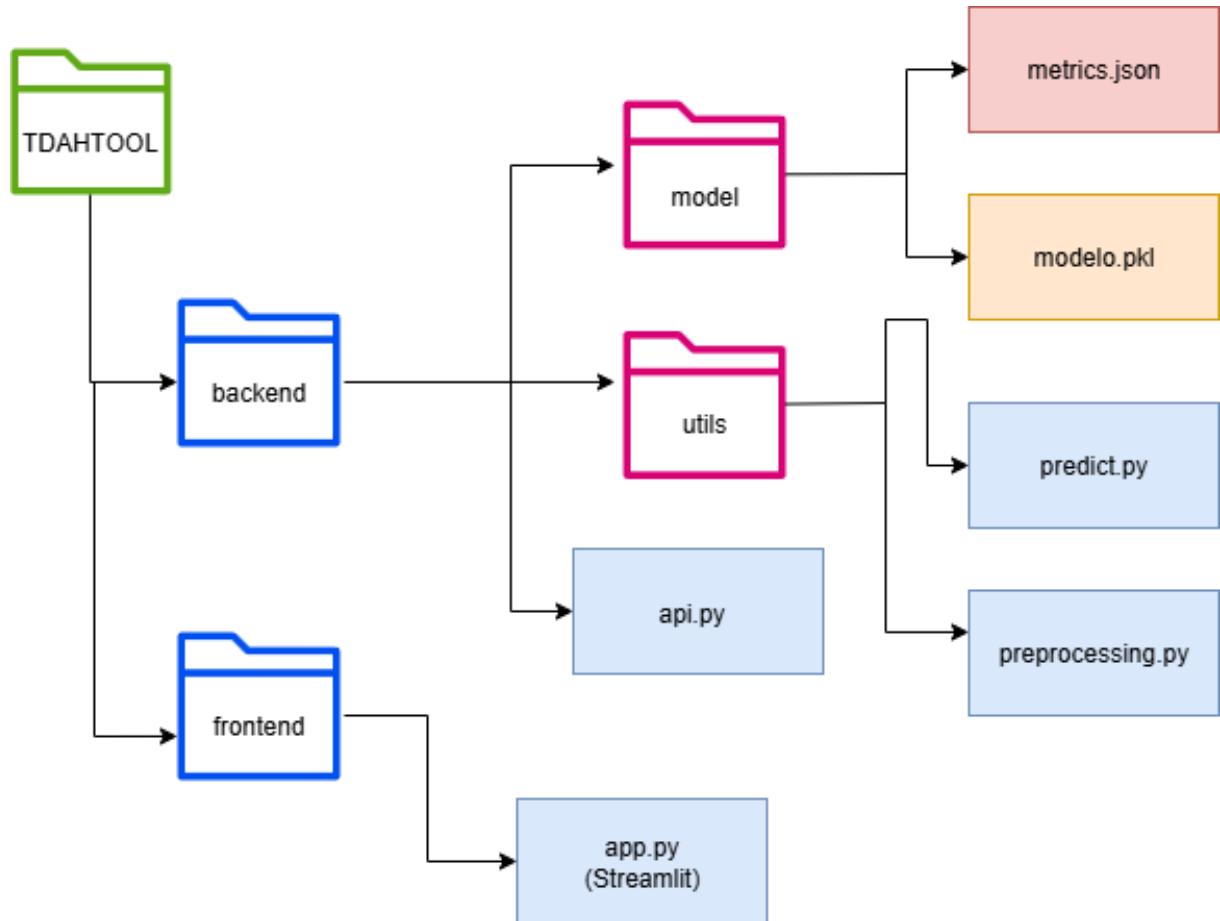


Figura 7.1: Estructura de carpetas de la aplicación

Nota. Elaboración propia con Draw.io

7.1.2. Requisitos funcionales y no funcionales

Requisitos Funcionales (RF)

- **RF1: Recepción de datos del paciente.** El sistema debe proveer una interfaz ('Formulario') que permita al usuario introducir valores para las 12 variables de entrada requeridas por el modelo.
- **RF2: Inferencia y visualización de predicción.** El sistema debe procesar la entrada del usuario, ejecutar la inferencia con el modelo de Árbol de Decisión calibrado y mostrar de forma clara la predicción de clase resultante ('TDAH' o 'No TDAH').
- **RF3: Visualización de probabilidades.** El sistema debe mostrar, junto a la predicción, las probabilidades porcentuales calculadas para cada clase ($P(\text{TDAH})$ y $P(\text{No TDAH})$) en un gráfico de barras.
- **RF4: Visualización de métricas del modelo.** El sistema debe exponer en la interfaz las métricas de rendimiento globales del modelo (calculadas offline), incluyendo *accuracy*, *recall*, *precision*, *F1-score*, *specificity*, AUC-ROC y el umbral óptimo de decisión.
- **RF5: Análisis de sensibilidad ('What-If').** El sistema debe proveer una interfaz ('Explorar') que permita al usuario modificar interactivamente los valores de entrada y recalcular la predicción, para observar la variación en la probabilidad de TDAH en tiempo real.
- **RF6: Trazabilidad del análisis 'What-If'.** La interfaz 'Explorar' debe registrar y visualizar un historial de las predicciones ejecutadas por el usuario, mostrando la evolución de $P(\text{TDAH})$ en un gráfico de líneas.
- **RF7: Transparencia del modelo.** El sistema debe proveer una vista ('Modelo') que muestre una representación visual estática del árbol de decisión original para fines de transparencia.

Requisitos No Funcionales (RNF)

- **RNF1: Usabilidad.** El sistema debe ser intuitivo y de fácil uso para un perfil no técnico.
- **RNF2: Arquitectura (Separación de responsabilidades).** El sistema debe implementar una separación estricta entre el *frontend* (Streamlit) y el *backend* (FastAPI).
- **RNF3: Mantenibilidad (Trazabilidad de versiones).** El sistema debe gestionar las versiones de las bibliotecas (ej. `scikit-learn`) para evitar incompatibilidades al cargar los artefactos del modelo (.pkl).
- **RNF4: Robustez (Preprocesado).** El *pipeline* de preprocesado debe manejar de forma robusta los datos de entrada para la inferencia.

Arquitectura del sistema

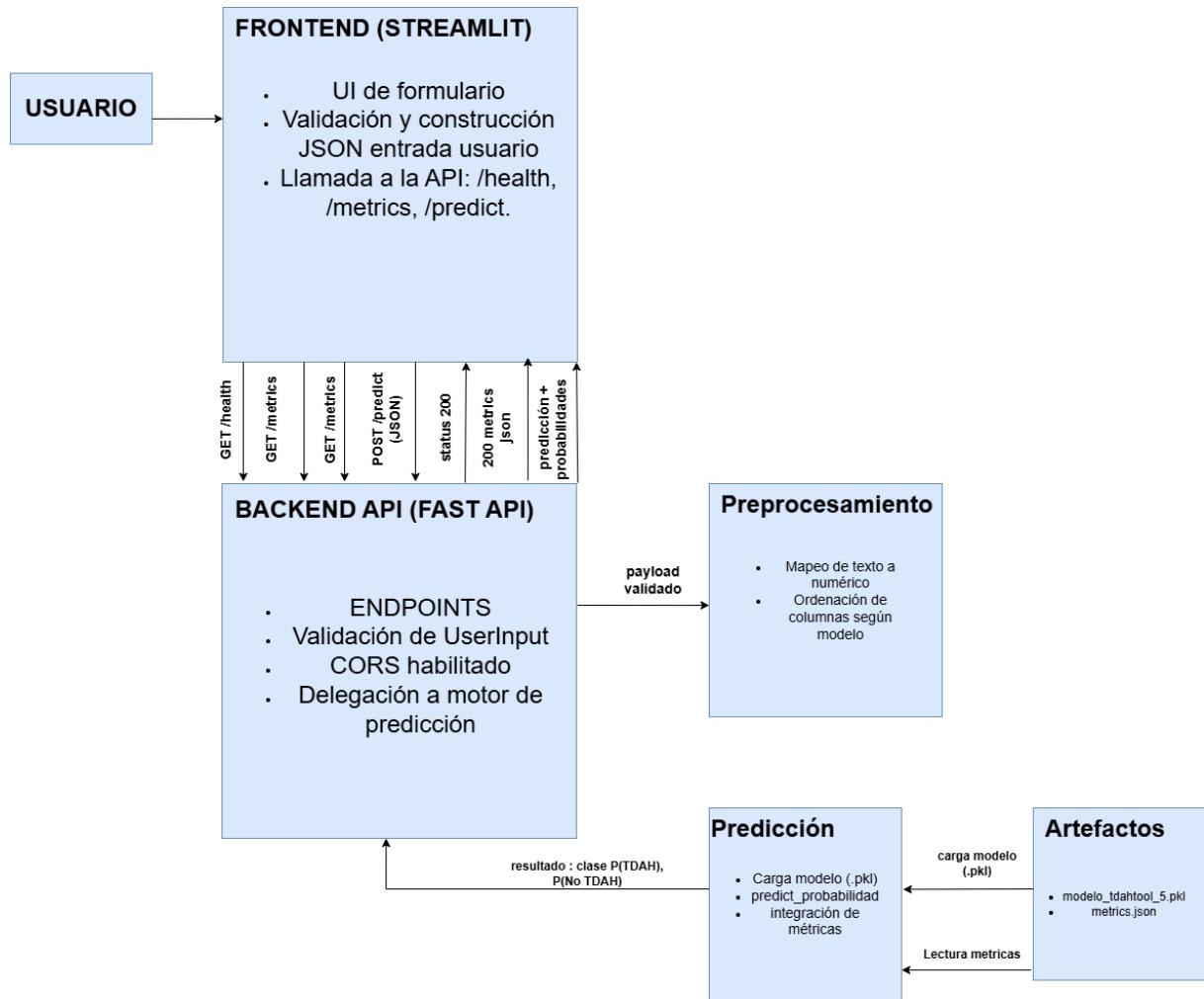


Figura 7.2: Arquitectura de la aplicación TDAH

La arquitectura de la aplicación TDAHTool, mostrada en la Figura 7.2, sigue un diseño desacoplado que separa la interfaz de usuario del motor de predicción.

El flujo se inicia cuando el **Usuario** interactúa con el **Frontend (Streamlit)**. Esta capa es responsable de la UI del formulario, la validación de la entrada del usuario y la construcción de un *payload* JSON.

Este *payload* se envía (vía POST /predict) al **Backend API (FastAPI)**, que actúa como controlador. El *backend* valida la entrada (*UserInput*), gestiona la seguridad (CORS) y delega la tarea al motor de predicción.

El *payload* validado pasa primero por el módulo de **Preprocesamiento**, donde los datos de texto se mapean a numéricos y las columnas se ordenan según el modelo. A continuación, el módulo de **Predicción** carga los **Artefactos** (el modelo .pkl y las metrics.json) para calcular las probabilidades de TDAH y No TDAH.

Finalmente, el resultado (clase y probabilidades) se devuelve al *backend*, y este lo reenvía al *frontend* para su visualización por parte del usuario.

7.2. Implementación

Desde el inicio se adopta una **arquitectura separada**: un frontend ligero y un backend que encapsula el modelo de ML y el preprocesado. Por tiempo y foco, se prioriza la **predicción individual** (no por lotes) y una interacción basada en formularios y *sliders*, con visualización inmediata de probabilidades y métricas.

A nivel tecnológico, el *stack* se define de la siguiente manera: *Streamlit* para el frontend, *FastAPI* para la API, *scikit-learn* para el modelo y dockerización del sistema (contenedores separados y orquestación con Docker Compose).

7.2.1. Entrenamiento del modelo y selección de variables

Para la puesta en producción del clasificador entrenado, se genera un **artefacto serializado** en formato **.pkl**, con el estimador final del Decision Tree y todos sus parámetros aprendidos (y calibrados). Este modelo constituye una versión “más ligera” del original, dado que se entrena el Decision Tree (con el mismo pipeline, calibración, ajuste de hiperparámetros, etc.) sobre las **12 variables más relevantes** identificadas (reducción de 52 a 12).

Esta reducción no penaliza el rendimiento; de hecho, **mejora la sensibilidad**, lo que justifica el uso de un conjunto “más compacto” en producción. El modelo se acompaña de un fichero JSON con las métricas “oficiales” calculadas *offline* para su visualización.

7.2.2. Backend: API y pipeline de inferencia

El backend se construye con FastAPI y cumple tres responsabilidades:

1. Exponer una API estable para el frontend.
2. Validar y preprocesar la entrada siguiendo el contrato de datos del modelo.
3. Ejecutar inferencia con el modelo entrenado y serializado en .pkl. La estructura se organiza en torno a **api.py** (endpoints/validación), **preprocessing.py** (mapeos y orden de columnas) y **predict.py** (carga de artefactos y predicción).

Los **endpoints** expuestos por el servicio backend son:

- **/health**: verificación de vida del servicio (retorno de {status: ok}).
- **/metrics**: método GET cuya función es devolver las métricas oficiales del modelo (las calculadas *offline*) y el umbral óptimo.
 - **Entrada**: sin *body*.
 - **Salida** (200 OK, JSON): incluye los campos *accuracy*, *recall*, *precision*, *F1-score*, *AUC-ROC*.
- **/predict**: método POST cuyo propósito es realizar **inferencia con el modelo**. Se recibe como entrada el JSON con las 12 variables con los literales exactos (strings) y enteros, introducidos por el usuario. Además, por defecto, como parámetros, se activa la inclusión de las métricas del modelo y la predicción con el umbral óptimo.
 - **Salida** (200 OK, JSON): *prediccion* (0/1), *probabilidad_tdh* (en %), *probabilidad_no_tdh* (en %), métricas, umbral óptimo y ruta del árbol de decisión junto con el grado de confianza en cada nodo.

- **Errores típicos:** **422** si el *payload* no valida (literal/tipo fuera de lo definido por **Pydantic**); **400** si algo falla en el preprocesado/predicción.

Validación y contrato de datos

Para garantizar que el frontend y el modelo se comuniquen con un contrato estricto y sin ambigüedades, en el backend se utiliza **Pydantic** (integrado en FastAPI) como capa de validación y *parseo* de datos de entrada. Pydantic permite declarar modelos tipados (clases) que describen la estructura, tipos y dominios válidos del JSON que llega a la API; al instanciar esos modelos, se valida y normaliza la información y, si algo no cuadra, se devuelven errores estructurados que FastAPI expone como HTTP 422 (Unprocessable Entity).

En el módulo **backend/api.py** se define el modelo de entrada **UserInput**. Este modelo se encarga de:

- Restringir variables categóricas con **Literal[...]** para forzar valores exactos (algunos en español y otros en inglés, tal y como se entrena el *pipeline*).
- Acotar numéricos con **Field(...)** (por ejemplo, en el caso de la variable edad, entre 0 y 18 como rango permitido).
- Servir como tipo del parámetro **payload** en el endpoint **POST /predict**, de modo que la validación ocurra **antes** de ejecutar la lógica de negocio.

Además, los parámetros de consulta (**include_metrics**, **use_optimal_threshold**) se incluyen en la firma del endpoint con **Query(...)** de modo que también queden validados y documentados automáticamente.

Preprocesado de texto

El preprocesado se encapsula en **preprocessing.py**, donde se definen:

- **MAPEOS**: diccionarios que transforman los literales de entrada (exactamente los validados por `UserInput`) a códigos numéricos requeridos por el modelo.
- **ORDERED_COLUMNS**: orden canónico de 12 columnas que el estimador espera en inferencia.

La función **preprocess_user_input(data)** aplica el mapeo, inserta directamente los enteros (`sc_age_years`, `birth_yr`) y construye un `DataFrame` con el orden correcto. Con este aislamiento se garantiza que el frontend no conozca detalles de codificación ni del orden esperado por el modelo.

Inferencia y artefactos del modelo

El módulo **predict.py** carga una única vez el modelo entrenado (`modelo.pkl`) con `joblib.load` cuando se importa el módulo (patrón de carga perezosa a nivel de proceso). También se define la ruta a `metrics.json`.

La función central **predecir_probabilidades()** realiza:

1. Llamada a `preprocess_user_input` para obtener el **DataFrame** codificado y ordenado.
 2. Ejecución del modelo para obtener probabilidades y clase.
 3. Devolución de un **diccionario** con `prediccion`, `probabilidad_no_tdah` y `probabilidad_tdah` en porcentaje (0–100).
- En caso de error controlado, se retorna `{'error': "..."}`; el endpoint lo traduce a HTTP 400.

Lógica del endpoint /predict

El módulo **api.py** recibe el *payload* ya validado y delega en `predecir_probabilidades`. Si el resultado contiene `.error`, se responde 400 con el detalle.

Si el usuario solicita `use_optimal_threshold=True`, el endpoint calcula también la predicción con umbral óptimo usando el valor hallado en `metrics.json` (cuando está disponible).

Manejo de errores y robustez

- **422 (validación)**: se produce automáticamente si algún literal no coincide o los tipos/rangos no son válidos (Pydantic).
- **400 (lógica de predicción)**: se emite cuando `predecir_probabilidades` devuelve un error (clave ausente, valor no mapeable, etc.); el endpoint lo eleva vía `HTTPException`.

Desarrollo funcionalidad ruta del árbol

Objetivo

La *funcionalidad de explicación de ruta* muestra, para una instancia concreta (datos de un usuario), qué decisiones toma el árbol hasta alcanzar la hoja que determina la predicción. En

cada nodo se presentan: el valor del usuario, la regla aplicada y un indicador de confianza local (pureza respecto a la clase finalmente predicha). Además, se reporta una confianza global derivada de las probabilidades calibradas.

Visión general del proceso

1. **Preprocesamiento de la entrada.** El diccionario recibido (entrada de usuario desde formulario) se transforma con `preprocess_user_input`, que:
 - a) Mapea etiquetas textuales a valores numéricos.
 - b) Devuelve un `DataFrame` de una única fila con todos los campos codificados a numérico.
2. **Predicción y probabilidades finales.** El modelo cargado es un `Pipeline` (formato `.pk1`) cuyo último paso es un `CalibratedClassifierCV` que envuelve un `DecisionTreeClassifier`. Con `model.predict_proba(X)` y `model.predict(X)` se obtienen, respectivamente:
 - a) Probabilidades *calibradas* de cada clase (promedio de varios calibradores).
 - b) La clase predicha.
3. **Selección del árbol para explicar la ruta.** Con `CalibratedClassifierCV` y `cv=5` se entrena cinco calibradores, cada uno con su árbol base (uno por *fold*). No existe un “único” árbol oficial; por ello:
 - Se evalúa cada calibrador para la instancia.
 - Se elige el *fold* más confiado: el que asigna mayor probabilidad (calibrada) a la clase finalmente predicha.

La ruta se calcula sobre el árbol base de ese *fold*.

4. **Cálculo de la ruta (camino de decisión).** Identificado el árbol base:

- Se alinean las columnas del `DataFrame` al orden exacto que espera el árbol (se evitan errores sutiles si el entrenamiento usa un subconjunto/orden concreto).
- Se utiliza la API de árboles de `scikit-learn`:
 - `tree.decision_path(X)`: nodos visitados por la muestra.
 - `tree.apply(X)`: id de la hoja alcanzada.
- Con los ids se reconstruye el camino nodo a nodo, extrayendo:
 - Variable del *split*.
 - Umbral de la regla.
 - Valor de la variable (del usuario).
 - Decisión tomada (\leq o $>$).
 - Proporciones de clases en el nodo (no calibradas).
- Se construye una *ruta lineal* compacta:

`feat_1` (valor \leq umbral) \rightarrow `feat_2` (valor $>$ umbral) $\rightarrow \dots \rightarrow$ [hoja]

5. **Confianza por nodo y confianza global.**

- **Confianza por nodo:** para cada nodo del camino se toma la proporción de la clase predicha en ese nodo (no calibrada), como indicador de *pureza* local.

- **Confianza global:** $\max(\text{predict_proba})$, es decir, el mayor de los valores devueltos por el modelo calibrado. Es el valor recomendado como confianza global.

Funciones implementadas

1. **_get_final_estimator.** Dado que el modelo se guarda en un Pipeline, se devuelve el último bloque que realmente predice (independiente del nombre o número de pasos previos).
2. **_unwrap_decision_tree.** Se navega por un estimador compuesto para localizar un DecisionTreeClassifier entrenado. Resulta crucial para extraer el camino de decisión del árbol real.
3. **_align_columns.** Se alinea el DataFrame de entrada al orden exacto de columnas que espera el árbol/estimador, evitando rutas incorrectas o errores por desorden de features.
4. **_unique_path_nodes.** Se obtiene la secuencia de nodos visitados por la muestra *sin duplicados* y excluyendo la hoja, para evitar repeticiones en la ruta.
5. **_extract_path_with_node_conf(tree: DecisionTreeClassifier, X_df, pred_label: int).**

Propósito: construir la explicación de ruta para una instancia.

- **Entrada:** tree (árbol entrenado), X_df (1 fila, preprocessada y alineada), pred_label (0/1).
- **Salida:** diccionario con *pasos* (nodos intermedios), *hoja* (una sola vez), *ruta_lineal* y confidencias por nodo.

Resumen de funcionamiento:

- a) Se verifica que el árbol posee `tree_` (si no, se lanza `TypeError`).
- b) Se determinan nombres de *features* (`feature_names_in_` si existen; en su defecto, columnas de `X_df`).
- c) Se alinean columnas con `_align_columns`.
- d) Se calcula `node_indicator = tree.decision_path(X_aligned)` y `leaf_id = tree.apply(X_aligned)`.
- e) Se obtienen nodos del camino con `_unique_path_nodes(...)`.
- f) Para cada nodo: *feature* usada, umbral, valor del usuario, decisión ($\leq >$), proporciones de clases (`t.value[node_id][0]`), y *node_confidence* para la clase final (si la clase final es 0, se toma $p(\text{NoTDAH})$ del nodo; si es 1, $p(\text{TDAH})$).
- g) Se calcula la hoja una única vez (mismas métricas).
- h) Se construye `ruta_lineal`:

$$\text{feat_1} \ (\text{valor } \leq \text{thr}) \rightarrow \text{feat_2} \ (\text{valor } > \text{thr}) \rightarrow \dots \rightarrow [\text{leaf}]$$

Utilidad: constituye el núcleo de la explicabilidad: se indica por dónde pasa la muestra y cuán “puro” es cada punto de decisión respecto a la clase final.

6. **_get_estimator_from_calibrated(cal_clf).**

Propósito: extraer el estimador subyacente de un calibrador individual (`CalibratedClassifierCV`), compatible con distintas versiones de `scikit-learn`.

Cómo: se buscan, en orden, los atributos `estimator`, `base_estimator`, `classifier`, `clf`,

devolviendo el primero existente (o `None` si no hay coincidencia).

Motivación: entre versiones de `sklearn` cambian los nombres de atributo; esta función evita roturas por incompatibilidad.

7. `explicar_ruta_lineal(data_dict: Dict[str, Any]) ->Dict[str, Any]`.

Propósito: función orquestadora: prepara datos, calcula la predicción y construye el JSON final con probabilidades calibradas, confianza global y explicación de ruta.

- a) Se preprocesa la entrada con `preprocess_user_input` (mapeos y orden de columnas).
- b) Se obtienen predicción y probabilidades calibradas del `model` cargado.
- c) Se llama a `_get_final_estimator` para obtener el último bloque (p. ej., `CalibratedClassifierCV`).
- d) Se selecciona el árbol a explicar:
 - Si existen `calibrated_classifiers_` (e.g., `cv=5`), se recorre cada calibrador, se extrae su estimador interno con `_get_estimator_from_calibrated` y se “desenvuelve” con `_unwrap_decision_tree`; se alinean columnas y se calcula la probabilidad del `fold` para la clase final; se elige el `fold` más confiado y se guardan metadatos en `source` (`type`, `fold`, `fold_probabilities`).
 - Si `cv="prefit"` (único calibrador), se toma su `base_estimator/estimator`.
 - Si no hay calibrador, se intenta desenvolver un árbol directo.
 - Si no se encuentra árbol, se devuelve JSON con error.
- e) Se llama a `_extract_path_with_node_conf` para construir `ruta_lineal`, `pasos` y `hoja`.
- f) Se devuelve el JSON final con: `prediccion`, `probabilidades_calibradas`, `confidence` (global) y, dentro de `explicacion`: `source`, `ruta_lineal`, `pasos`, `hoja`.

7.2.3. Frontend: interfaz y comunicación con la API

Objetivo y enfoque

El frontend se diseña con Streamlit para ofrecer una interfaz directa, en tres pestañas, donde el usuario introduce variables, explora cómo varía la probabilidad de TDAH en función de cambios en los valores de las variables predictoras principales y consulta el árbol del experimento. Se prioriza una UX sencilla (selectores y *sliders*), visualizaciones inmediatas y un contrato de datos estable con la API.

Estructura general y configuración

El módulo principal del frontend se define en `app.py`, que establece la **configuración global** del `layout`, define **rutas robustas** para extraer las métricas y la imagen del árbol (pestaña “Modelo”) y declara **constants** con las 12 variables que usa el backend.

Comunicación con la API

Para la comunicación con la API del backend se implementan funciones de ayuda (*helpers*):

- `api_health()`: comprueba el estado de la conexión con la API.
- `get_metrics()`: *cachea* durante 30s la respuesta del método `/metrics` (métricas oficiales del modelo).

- **call_predict()**: envía POST a `/predict` con el JSON del usuario y los parámetros de *query*.
- **extract_pct(resp)**: homogeniza la lectura de probabilidades (soporta varios formatos).

Localización de campos (UX). Contrato frontend-backend

Las etiquetas visibles se presentan en español, pero los **valores enviados** corresponden **exactamente** a los literales que valida el backend (algunos en español y otros en inglés). Para las opciones en inglés se emplea **format_func** con un pequeño traductor que solo cambia el texto mostrado, manteniendo el valor original para la API. Con ello se resuelve el 422 que aparece al intentar mapear a números enteros y opciones en español desde el frontend.

Construcción del formulario y *payload*

La construcción del formulario se realiza con la función **render_controls()**, que genera los selectores y *sliders* en 2 columnas y devuelve un diccionario con las 12 claves técnicas y tipos que exige el backend (cadenas para variables categóricas, enteros para edad y año de nacimiento...). Este diccionario se envía tal cual a `/predict`.

Interfaz de Usuario

La interfaz se organiza en 3 pestañas: **Formulario**, **Explorar** y **Modelo**.

1. Formulario:

La pestaña de formulario (Figura 7.3) muestra el estado de la API, la predicción de TDAH en función del formulario llenado con las principales variables del modelo, y permite incluir métricas, obtener la ruta del árbol con grados de confianza en los nodos y, opcionalmente, visualizar el *payload* que se envía al backend.

Tras **POST /predict**, se muestran:

- Barras TDAH vs No-TDAH y métricas numéricas (*accuracy*, *recall*, *precision*, *F1-score*, *AUC-ROC*).
- Umbral óptimo.
- Un *expander* con la respuesta cruda de `/predict` para depuración.

2. Explorar:

Esta vista (Figura 7.4) habilita un análisis de sensibilidad *what-if*. Se registra en **st.session_state** el historial de probabilidades TDAH/No-TDAH y los *inputs* de usuario usados. Se muestra tanto la **evolución temporal de P(TDAH)** con la **línea de umbral** como un gráfico de barras por ejecución, incluyendo un botón de **reset** para limpiar el historial.

3. Modelo:

Esta pestaña (Figura 7.5) se centra en la transparencia. Se carga y muestra una **imagen estática del árbol completo** (correspondiente al árbol de decisión generado durante la fase de entrenamiento con el dataset de 52 variables), resolviendo la ruta con **pathlib** y permitiendo la subida manual si no se encuentra el archivo. En paralelo, se listan las **12 variables** productivas del backend.



TDAH Tool — Demo

API conectada ✓

Formulario Explorar Modelo

Introducir datos y obtener predicción

F_Estado de conducta del paciente Nunca diagnosticado	F_Dificultad para cuidar al paciente Nunca
F_Edad del paciente (años) 10	F_El paciente discute mucho Nunca
F_Año de nacimiento 2015	F_Estado de ansiedad en el paciente Nunca diagnosticado
F_Estado de servicios de educación especial Nunca ha tenido plan especial de educa...	F_Persistencia del paciente para finalizar las tareas Nunca
F_Condiciones de salud presentes en el paciente El menor no presenta problemas de salud	F_Dificultad para hacer amigos Mucho dificultad
	F_Sexo del paciente Femenino
	F_Tiempo de juego del paciente (entre semana) Muy joven (<3 años)

Figura 7.3: Aplicación TDAHTOOL — Formulario (1)

Captura de la aplicación TDAHTool.

Formulario Explorar Modelo

Mover controles y visualizar cómo cambia la probabilidad

E_Estado de conducta del paciente

Nunca diagnosticado

E_Dificultad para cuidar al paciente

Nunca

E_Edad del paciente (años)

10

E_El paciente discute mucho

Nunca

E_Año de nacimiento

2015

E_Estado de ansiedad en el paciente

Nunca diagnosticado

E_Estado de servicios de educación especial

Nunca ha tenido plan especial de edu...

E_Persistencia del paciente para finalizar las tareas

Nunca

E_Condiciones de salud presentes en el paciente

El menor no presenta problemas de salud

E_Dificultad para hacer amigos

Mucha dificultad

E_Sexo del paciente

Femenino

E_Tiempo de juego del paciente (entre semana)

Muy joven (<3 años)

Calcular y registrar punto

Reset historial

Ver histórico de entradas

Figura 7.4: Aplicación TDAHTOOL — pestaña “Explorar” (formulario).

Captura de la aplicación TDAHTOOL.

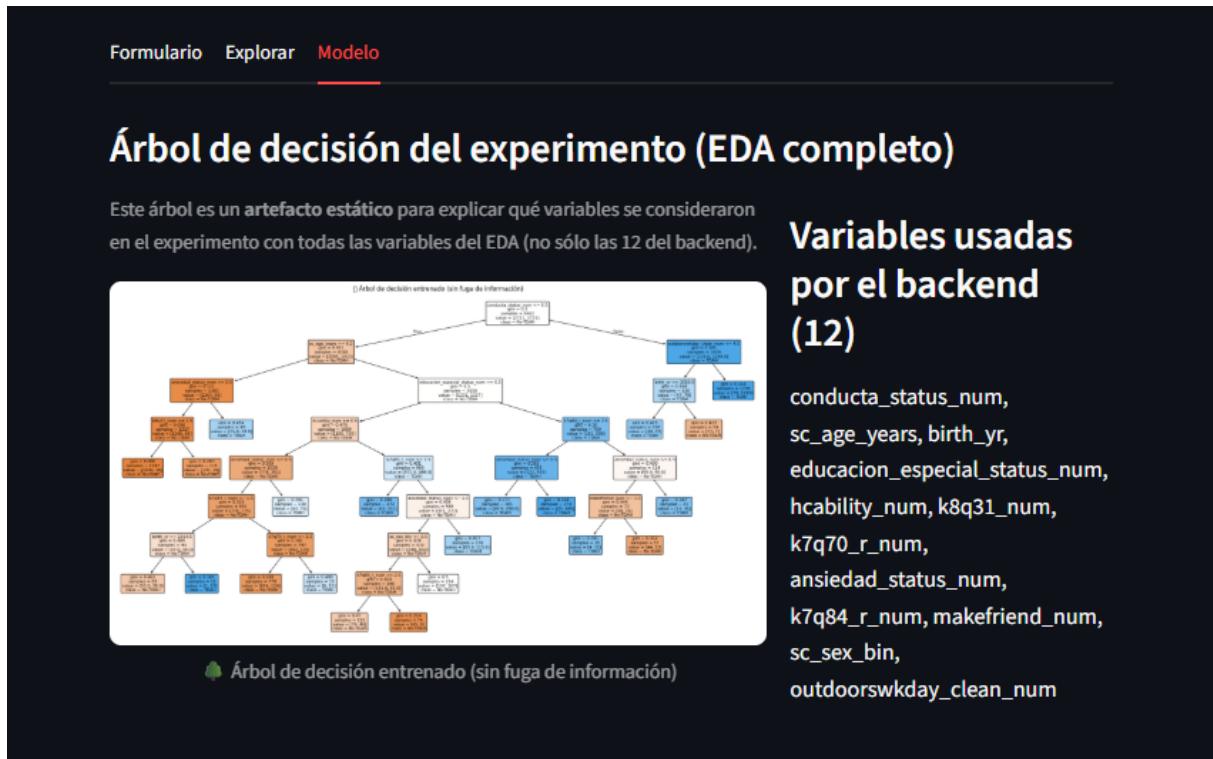


Figura 7.5: Aplicación TDAHTOOL — pestaña “Modelo” (visualización del árbol).

Captura de TDAHTool.

Análisis Detallado de la Explicabilidad (Ruta del Árbol)

En este apartado se describe el proceso de diseño e implementación de la capacidad de explicar *cómo* el modelo llega a una predicción concreta dentro de la pestaña *Formulario* de TDAHTool, mediante la reconstrucción de la ruta de decisión en el árbol y la exposición de probabilidades *calibradas* coherentes con el *pipeline* de producción.

Objetivo funcional. Se persigue que, para cada muestra introducida en el formulario, el sistema:

1. Devuelva la predicción final del modelo con probabilidades **calibradas** (salida operativa).
2. Muestre la **ruta de decisión** seguida en un árbol real del sistema, con sus umbrales, variables y estados nodales.
3. Alinee la explicación (ruta) con la probabilidad calibrada más representativa para la clase predicha.

Contexto del modelo. El modelo en producción es un `DecisionTreeClassifier` envuelto en `CalibratedClassifierCV` con $cv = 5$. Cada *fold* dispone de su árbol base y de un calibrador. El entrenamiento usa *undersampling* y, en el árbol, `class_weight=balanced`. Por ello:

- La **pureza de hoja** (proporciones internas del árbol) no se calibra y puede estar sesgada respecto a la distribución real.
- La **probabilidad final** operativa se obtiene promediando las probabilidades calibradas de los 5 calibradores.

Diseño de la solución. Se adoptan tres decisiones clave:

1. **Separación de responsabilidades.**
 - **Backend:** cálculo de predicción, selección del *fold* de referencia, reconstrucción de ruta y empaquetado de métricas (pureza, probabilidad calibrada del *fold* y probabilidad final).
 - **Frontend:** representación textual y estructurada de la ruta y de las probabilidades.
2. **Selección del *fold* para la explicación.** Dado que la decisión operativa promedia calibradores, para explicar un *camino* concreto se escoge el ***fold* más confiado** para la clase predicha, es decir, el que asigna la mayor $P(\text{clase predicha} | x)$. Esta elección:
 - Garantiza que la ruta mostrada proviene de un *árbol real* del sistema.
 - Alinea la cifra calibrada mostrada junto a la ruta con el calibrador que más respalda la decisión.
3. **Contrato de datos (API).** El endpoint de inferencia devuelve una estructura JSON con:
 - `pred_label` y `final_calibrated_proba` (promedio entre *folds*).
 - `route`: lista ordenada de pasos con `feature`, `threshold`, `operator`, `node_id` y `leaf_raw_proba` (distribución interna no calibrada).
 - `fold_seleccionado`: índice del *fold* más confiado y su `fold_calibrated_proba` para la clase predicha.

Implementación técnica (backend).

1. **Preprocesado coherente.** Se reutiliza el mismo *pipeline* de preparación de variables del entrenamiento (codificaciones ordinales/categóricas, imputación, orden de columnas) para construir el vector x de inferencia.
2. **Predicción y calibración.** Se obtiene \hat{y} y el vector de probabilidades promedio con `CalibratedClassifierCV`. En paralelo, se evalúa cada calibrador/*fold* para extraer $P^{(i)}(\hat{y} | x)$ y se selecciona el *fold argmax*.
3. **Reconstrucción de la ruta.** Con el árbol del *fold* seleccionado:
 - Se recorre `tree_.feature` y `tree_.threshold`, determinando los saltos izquierdo/-derecho; se mapean índices a nombres de variables del dataset ya transformado.
 - En cada nodo se registra `tree_.value` (para `leaf_raw_proba`) y el operador/umbral aplicado.
 - Se marca la hoja alcanzada y se adjuntan métricas de nodo/hoja.
4. **Salida consistente.** Se empaquetan: `pred_label`, `final_calibrated_proba` (promedio), `fold_calibrated_proba` (del *fold* de la ruta), `leaf_raw_proba` y la secuencia de decisiones.

Integración en *Frontend* (pestaña Formulario).

- Se muestra primero la **probabilidad final calibrada** (decisión operativa).
- Bajo ella, la **ruta de decisión** como lista de reglas humanas: `variable ≤ umbral / > umbral`, hasta la hoja.
- En la hoja se indican, de forma breve: `leaf_raw_proba` (no calibrada), `fold_calibrated_proba` (del árbol de la ruta) y `final_calibrated_proba` (promedio entre *folds*).

Notas sobre probabilidades: pureza vs. calibración.

- **Pureza de hoja (no calibrada):** proporción de clases en el nodo/hoja del árbol base (`tree_.value`); puede estar sesgada por *undersampling*/pesos.
- **Probabilidad calibrada (fold de la ruta):** $p_{fold} = \text{calibrated_classifiers}_{[fold]}.predict_proba$ para el árbol mostrado.
- **Probabilidad final (modelo):** $p_{final} = \frac{1}{5} \sum_{i=1}^5 p_{\text{calibrada}}^{(i)}$; es la cifra que se utiliza para decidir.

Ventajas de esta solución

1. **Fidelidad al sistema de producción.** La decisión se toma con *árbol + calibración + promedio de folds*. Mostrar solo la pureza de la hoja no refleja el comportamiento real del *pipeline*; añadir probabilidades calibradas sí lo hace.
2. **Coherencia con la ruta.** Fijar la ruta a un *fold* concreto (el más confiado) y enseñar su probabilidad calibrada conecta reglas con una cifra coherente con ese camino.
3. **Transparencia estadística.** La pureza explica qué observa el árbol durante el entrenamiento; las probabilidades calibradas muestran la cifra que realmente se usa para decidir.
4. **Evita conclusiones erróneas.** Con *undersampling* y `class_weight`, las frecuencias de hoja pueden sesgarse; la calibración corrige ese sesgo.

Limitaciones y trabajo futuro.

- La ruta refleja un único árbol (el del *fold* más confiado); puede diferir de los demás árboles del conjunto.
- En versiones futuras, se valora mostrar un *resumen* de rutas mayoritarias entre *folds* o indicadores de estabilidad de reglas, e incorporar medidas locales adicionales.

Conclusión. La funcionalidad combinada ofrece **interpretabilidad local** (ruta de un árbol real) con **probabilidades fiables** (calibradas y promediadas), alineando la explicación con el comportamiento de producción y evitando lecturas basadas únicamente en purezas de hoja.

Manejo de errores y diagnóstico

Para gestionar el manejo de errores se establece la siguiente política:

- Los errores HTTP se muestran con detalle (incluido el **detail** de Pydantic cuando hay 422).
- *Expanders* de **payload** (antes de llamar) y de **respuesta** (después de llamar) facilitan la detección de problemas con el contrato frontend-backend.

7.2.4. Docker y Docker Compose

Con el objetivo de facilitar el desarrollo, escalado y despliegue de la aplicación TDAHTool en distintos entornos sin conflictos de dependencias, se opta por empaquetar y desplegar la aplicación de manera reproducible y aislada mediante **Docker**, separando interfaz y API en contenedores independientes y orquestándolos mediante **Docker Compose**.

Arquitectura y servicios Docker

Con el objetivo de separar la interfaz y el backend, la aplicación se divide en 2 servicios (cada uno en su contenedor), conectados entre sí por la red interna de Docker, tal y como se observa en la Figura 7.6.

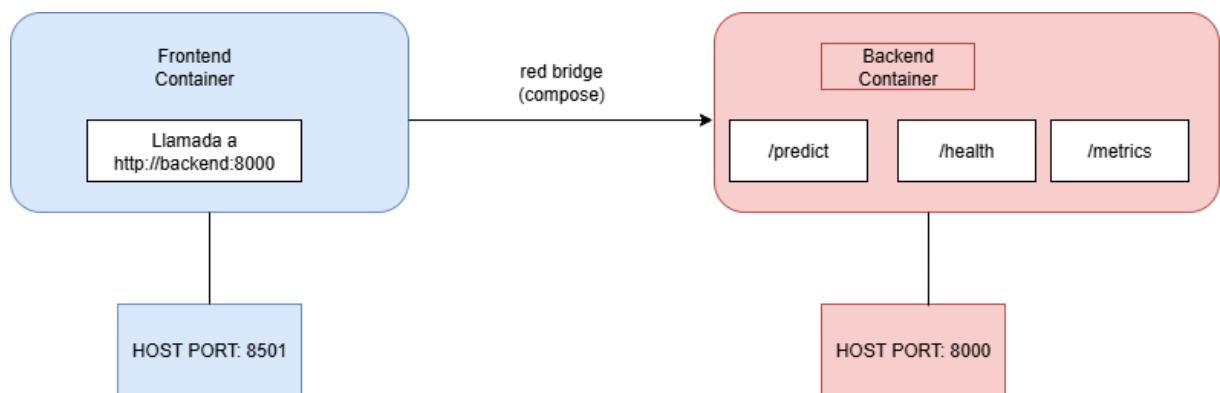


Figura 7.6: Arquitectura Docker.

Captura sacada de diagrama propio hecho en Draw.io

Se identifican 2 elementos clave de la arquitectura:

- **backend:** servicio **FastAPI** que expone los endpoints (`/health`, `/predict`, `/metrics`), carga el modelo y las métricas desde `backend/model`.
- **frontend:** aplicación **Streamlit** que consume la API del backend. La URL del backend se inyecta por variable de entorno (**API_URL**) (por defecto `http://backend:8000` dentro de la red Docker).

Para habilitar la comunicación y orquestación entre contenedores, se definen 2 **Dockerfile** (backend y frontend), orquestados por un fichero **docker-compose.yml**.

Decisiones clave y buenas prácticas

- Creación de **red interna por nombre de servicio**. Cada servicio se registra dentro de una red *bridge* como un nombre DNS. De este modo no se depende de **localhost** ni de IPs, ya que se usa el **nombre del servicio**, ganando portabilidad.
- **Volumen del modelo en solo lectura (RO):** en `docker-compose.yml`, se monta el directorio del host (`./backend/model`) dentro del contenedor en `/app/backend/model`, en modo lectura. Actualizaciones del modelo no requieren *rebuild*; la integridad queda protegida con `:ro`.
- **Parámetro depends_on:** se garantiza el orden de arranque (primero backend, después frontend). Para esperar a que la API esté saludable, se añade `healthcheck` en backend y `condition: service_healthy` en frontend.
- **Versionado de dependencias:** requisitos establecidos en `requirements.txt` para *builds* reproducibles.
- **.dockerignore:** se acelera la construcción de imágenes.

7.2.5. Ingeniería del software y buenas prácticas

En este apartado se comenta cómo la ingeniería del software aporta un conjunto de metodologías, principios y herramientas cuyo propósito es garantizar que el desarrollo de aplicaciones se realiza de manera **sistemática, controlada y eficiente**.

Con estas pautas se contribuye a la calidad y fiabilidad de la aplicación final y se favorece su mantenibilidad, escalabilidad y facilidad de evolución en entornos reales.

Control de versiones

Durante el desarrollo de la aplicación se utiliza **GitHub** como plataforma principal de control de versiones. Se crea un repositorio (véase Tabla[6.16] de la Subsección[6.6.4]) en el que se va subiendo de manera progresiva el código implementado en cada una de las fases de desarrollo y prueba.

El motivo de utilizar control de versiones como GitHub responde a varias razones:

- **Gestión del historial de cambios:** se registra de manera estructurada cada modificación, facilitando la trazabilidad y la posibilidad de recuperar versiones previas en caso de errores o regresiones.
- **Colaboración y escalabilidad:** el uso de integración continua facilita un flujo de trabajo colaborativo ante futuras ampliaciones o mantenimientos por otros desarrolladores.
- **Integración con herramientas externas:** se habilita la integración de flujos de trabajo automatizados (CI/CD), como ejecución de pruebas automáticas o despliegue continuo.

Limpieza y estandarización del código

La **calidad del código fuente** se considera fundamental. De ella dependen la mantenibilidad, escalabilidad y comprensión futura del software. Con este propósito, durante la implementación se siguen criterios de limpieza y estandarización inspirados en buenas prácticas (Sommerville, 2015).

Dado que el lenguaje de desarrollo es Python, se adopta el estilo PEP 8 (van Rossum et al., 2001) para una sintaxis uniforme y legible: uso coherente de sangrías y espacios, nombres descriptivos para funciones, variables y clases, y separación de bloques lógicos mediante comentarios estructurados.

Asimismo, se cuida la **organización modular del código**: en el backend, la lógica de predicción se encapsula en funciones auxiliares dentro de un módulo **utils**, mientras que otro archivo se centra en la definición de endpoints y esquemas de entrada, respetando la separación de responsabilidades. En el frontend, en **app.py** se adopta un diseño análogo, integrando funciones de apoyo con una clara separación entre la construcción de la interfaz y la comunicación con la API.

Se incorporan *docstrings* y comentarios explicativos para documentar el propósito de funciones y endpoints, facilitando la comprensión por parte de otros desarrolladores y garantizando la trazabilidad del diseño.

Finalmente, se aplican prácticas de gestión de constantes y configuraciones mediante variables centralizadas (como **API_URL**, **BACKEND_VARS_12** o los diccionarios de traducción en el frontend), reduciendo duplicación y favoreciendo la mantenibilidad.

Justificación de frameworks

- **Frontend — Streamlit**

Streamlit se elige como *framework* de interfaz de usuario por su orientación al desarrollo rápido de aplicaciones interactivas en Python. Su principal ventaja es la **simplicidad** en la construcción de formularios y visualizaciones, permitiendo que los usuarios interactúen con el modelo sin conocimientos técnicos avanzados. Además, Streamlit facilita la integración con **matplotlib** y **pandas**, posibilitando la visualización clara de métricas y resultados.

- **Backend — FastAPI**

Para la API se selecciona FastAPI, *framework* moderno basado en estándares como OpenAPI y JSON Schema. FastAPI permite definir de manera declarativa los esquemas de entrada y salida mediante *Pydantic*, garantizando validación automática de datos y reduciendo errores en la comunicación con el frontend. Su sintaxis clara y la documentación interactiva generada automáticamente suponen ventajas relevantes respecto a marcos tradicionales.

- **Orquestación — Docker Compose**

Se emplea Docker Compose para orquestar contenedores y desplegar conjuntamente frontend y backend en un entorno controlado y reproducible. La contenerización asegura portabilidad y escalabilidad, gestionando la comunicación entre servicios de forma sencilla y proporcionando un entorno aislado y coherente para pruebas y despliegue.

7.3. Validación y demostración funcional

Objetivo de la validación El objetivo de esta sección es demostrar el comportamiento operativo de la aplicación. Se busca validar que la herramienta, a partir de datos reales introducidos por el usuario, genera predicciones individuales sobre el riesgo de TDAH de forma visualmente comprensible.

Asimismo, se verifica la funcionalidad interactiva (pestaña ‘Explorar’) que permite observar cómo varían la probabilidad de TDAH y el umbral de decisión al modificar los valores de entrada.

Pestaña ‘Formulario’ (Validación de predicción individual)

Esta pestaña implementa el requisito funcional **RF1**. Se presenta un formulario con controles de interfaz (selectores, *sliders*, campos numéricos) adecuados al tipo de cada variable. Al completarlo, el botón *Calcular* activa la inferencia en el *backend*.

Para comprobar la fiabilidad de la aplicación y su correcta integración con el modelo productivo (Sección 7.2.1), se realiza una prueba con dos casos extraídos del conjunto de *test*: un paciente con diagnóstico negativo y otro con diagnóstico positivo en TDAH. El objetivo es verificar que la predicción de la aplicación coincide con el resultado esperado del modelo.

Caso Negativo: Predicción Se introduce un paciente del conjunto de *test* cuyo resultado conocido es **negativo (No TDAH)**. Los valores introducidos en el formulario son los siguientes:

- **Estado de conducta del paciente:** Nunca diagnosticado
- **Edad del paciente (años):** 5
- **Año de nacimiento:** 2020
- **Estado de servicios de educación especial:** Nunca ha tenido plan especial de educación
- **Condiciones de salud presentes en el paciente:** This child does not have any health conditions (El menor no presenta problemas de salud)
- **Dificultad para cuidar al paciente:** Sometimes (A veces)
- **El paciente discute mucho:** Sometimes (A veces)
- **Estado de ansiedad en el paciente:** Nunca diagnosticado
- **Persistencia del paciente para finalizar las tareas:** Usually
- **Dificultad para hacer amigos:** Sin dificultad
- **Sexo del paciente:** Masculino
- **Tiempo de juego del paciente (entre semana):** 4 or more hours per day (4 horas o más por día)

Las Figuras 7.7 y 7.8 muestran la entrada y la salida. Al ejecutar *Calcular (Formulario)*, la aplicación comprueba la conexión con la API y devuelve los gráficos de probabilidad. El resultado coincide con la etiqueta del conjunto de *test*, y el paciente es clasificado como **negativo**.

API conectada ✓

Formulario Explorar Modelo

Introducir datos y obtener predicción

F_Estado de conducta del paciente Nunca diagnosticado	F_Dificultad para cuidar al paciente A veces
F_Edad del paciente (años) 5	F_El paciente discute mucho A veces
F_Año de nacimiento 2020	F_Estado de ansiedad en el paciente Nunca diagnosticado
F_Estado de servicios de educación especial Nunca ha tenido plan especial de educa...	F_Persistencia del paciente para finalizar las tareas Normalmente
F_Condiciones de salud presentes en el paciente El menor no presenta problemas de salud	F_Dificultad para hacer amigos Sin dificultad
	F_Sexo del paciente Masculino
	F_Tiempo de juego del paciente (entre semana) 4 o más horas por día

Figura 7.7: Formulario de entrada (Caso negativo).

Captura sacada de TDAHTool.

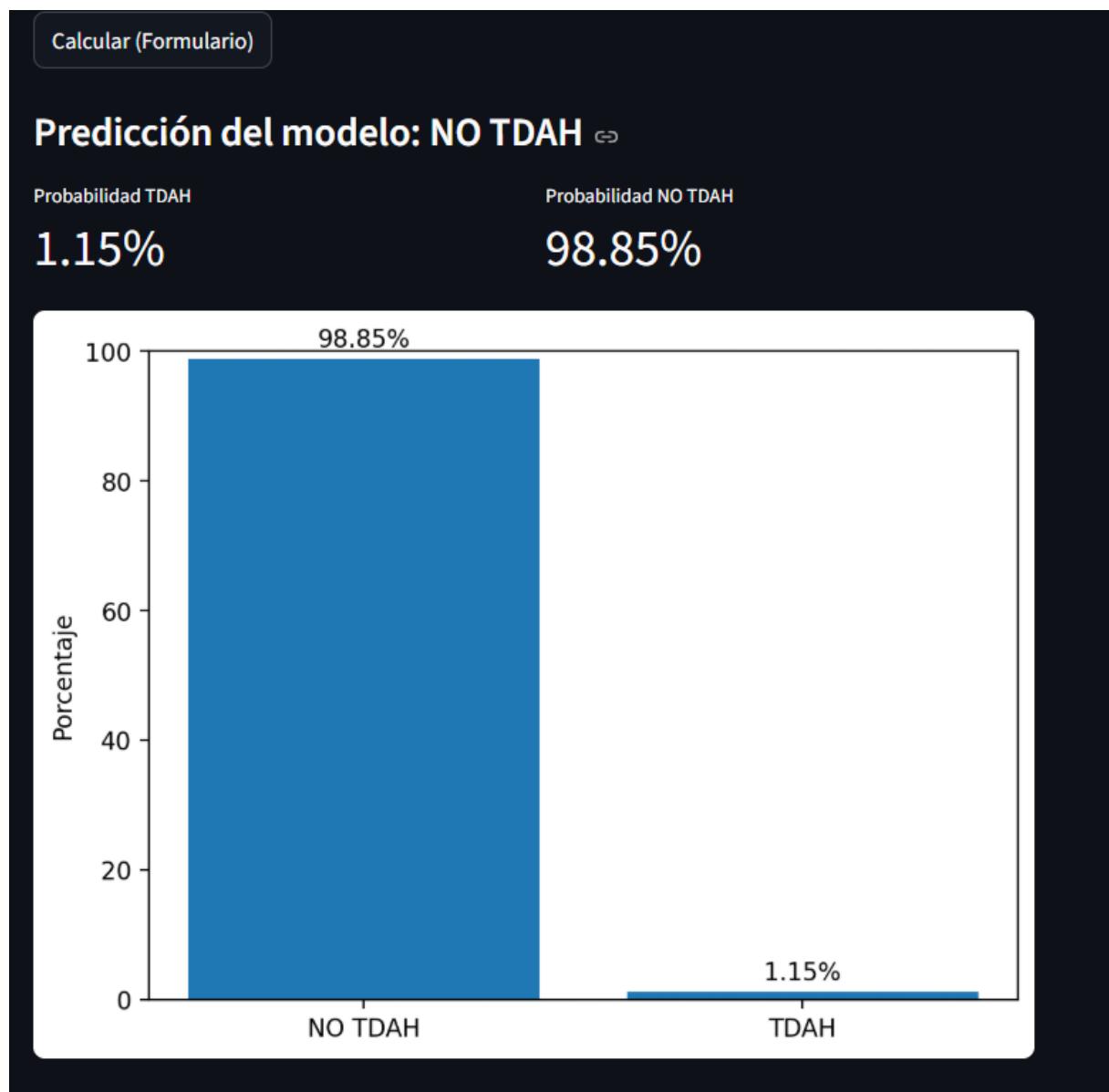


Figura 7.8: Resultado de la predicción (Caso Negativo: No TDAH).

Captura sacada de TDAHTool.

Caso Positivo: Predicción De manera análoga, se introduce un caso del conjunto de *test* etiquetado como **positivo (TDAH)**, con los siguientes valores:

- **Estado de conducta del paciente:** Diagnosticado pero ya no lo tiene
- **Edad del paciente (años):** 16
- **Año de nacimiento:** 2009
- **Estado de servicios de educación especial:** Tiene plan y recibe servicios de educación especial

- **Condiciones de salud presentes en el paciente:** This child does not have any health conditions (El menor no presenta problemas de salud)
- **Dificultad para cuidar al paciente:** Nunca
- **El paciente discute mucho:** A veces
- **Estado de ansiedad en el paciente:** Diagnosticado y lo tiene actualmente
- **Persistencia del paciente para finalizar las tareas:** Normalmente
- **Dificultad para hacer amigos:** Mucha dificultad
- **Sexo del paciente:** Masculino
- **Tiempo de juego del paciente (entre semana):** 4 or more hours per day (4 horas o más por día)

Como se documenta en las Figuras 7.9 y 7.10, tras la ejecución la aplicación devuelve los gráficos de probabilidad. El resultado coincide con el valor esperado del conjunto de *test*, y el paciente es clasificado correctamente como **positivo**. Esta validación confirma que el *pipeline* de preprocesado e inferencia funciona según lo diseñado.

API conectada ✓

Formulario Explorar Modelo

Introducir datos y obtener predicción

F_Estado de conducta del paciente	F_Dificultad para cuidar al paciente
Diagnosticado pero ya no lo tiene	Nunca
F_Edad del paciente (años)	F_El paciente discute mucho
16	A veces
F_Año de nacimiento	F_Estado de ansiedad en el paciente
2009	Diagnosticado y lo tiene actualmente
F_Estado de servicios de educación especial	F_Persistencia del paciente para finalizar las tareas
Tiene plan y recibe servicios de educaci...	Normalmente
F_Condiciones de salud presentes en el paciente	F_Dificultad para hacer amigos
El menor no presenta problemas de salud	Mucha dificultad
F_Sexo del paciente	
Masculino	
F_Tiempo de juego del paciente (entre semana)	
4 o más horas por día	

Figura 7.9: Formulario de entrada (Caso Positivo).

Captura sacada de TDAHTool.

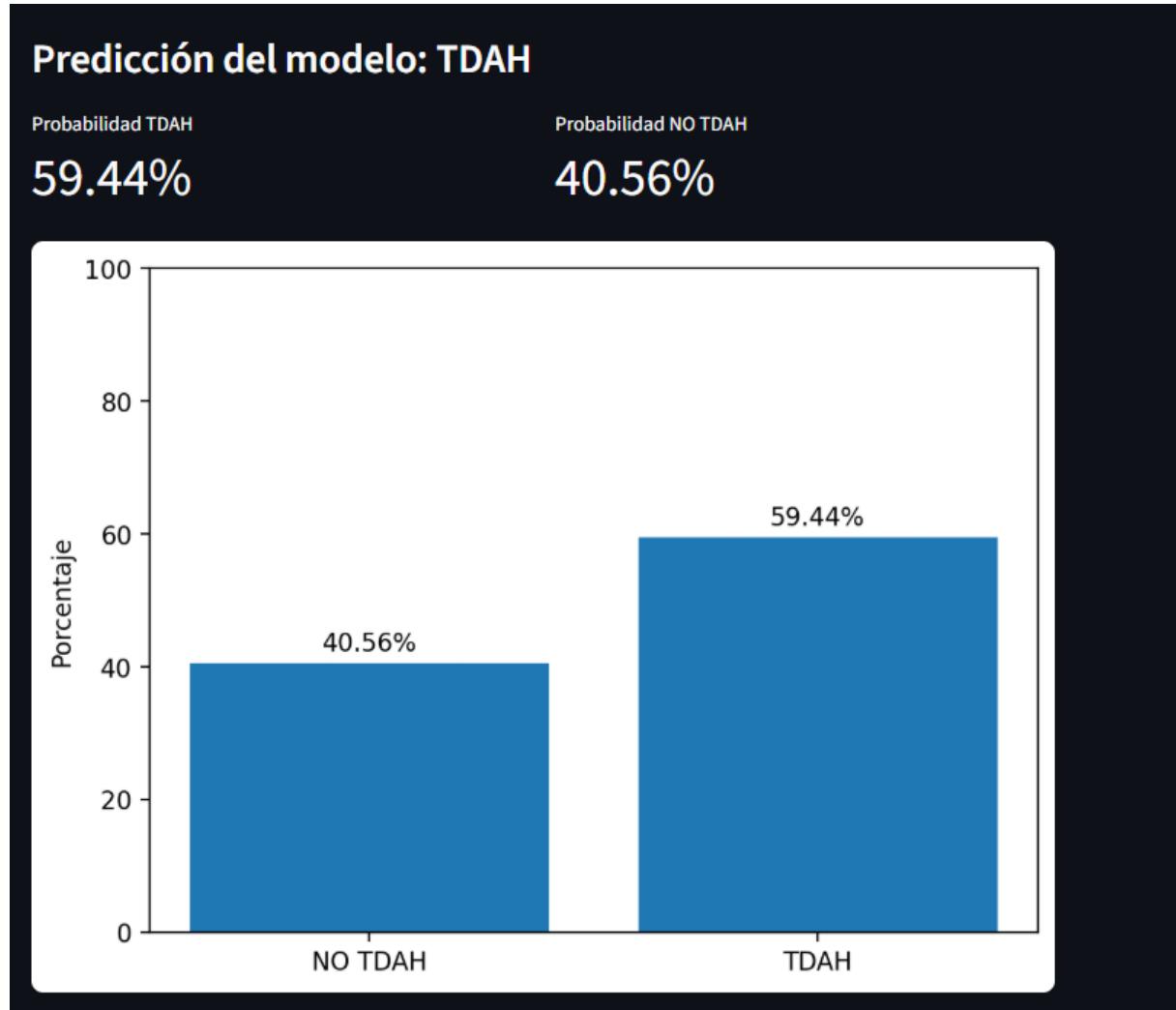


Figura 7.10: Resultado de la predicción (Caso Positivo: TDAH).

Captura sacada de TDAHTool.

Demostración de la ruta explicada del árbol (Explicabilidad local) Además de la predicción, la pestaña *Formulario* muestra cómo se llega a dicha predicción. Se presentan tanto la predicción final con probabilidades *calibradas* como la ruta de decisión concreta que sigue la muestra a través de un árbol real del sistema.

Bloques mostrados.

- **Predicción del modelo** (Figura 7.11): probabilidades *calibradas* del *pipeline* final (promedio de calibradores). Es la salida que se utiliza para decidir. Además, se muestran las métricas globales del modelo utilizado en el backend (*metrics.json*).
- **Ruta del árbol** (Figura 7.12): secuencia de nodos hasta la hoja alcanzada. En cada paso se muestran las proporciones internas (no calibradas) observadas por el árbol durante su entrenamiento.

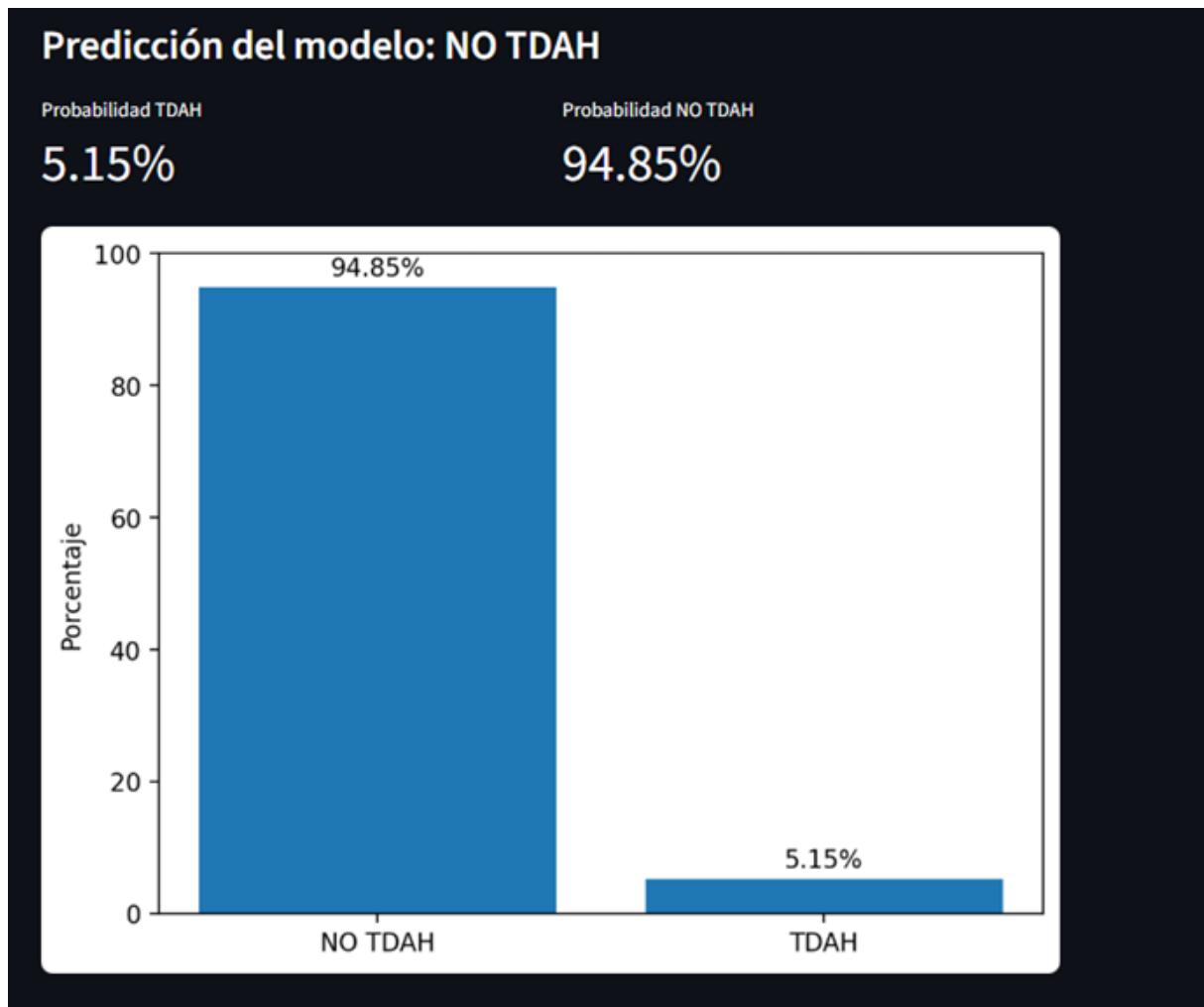


Figura 7.11: Predicción del modelo con probabilidades calibradas (salida operativa).

Captura de TDAHTool.

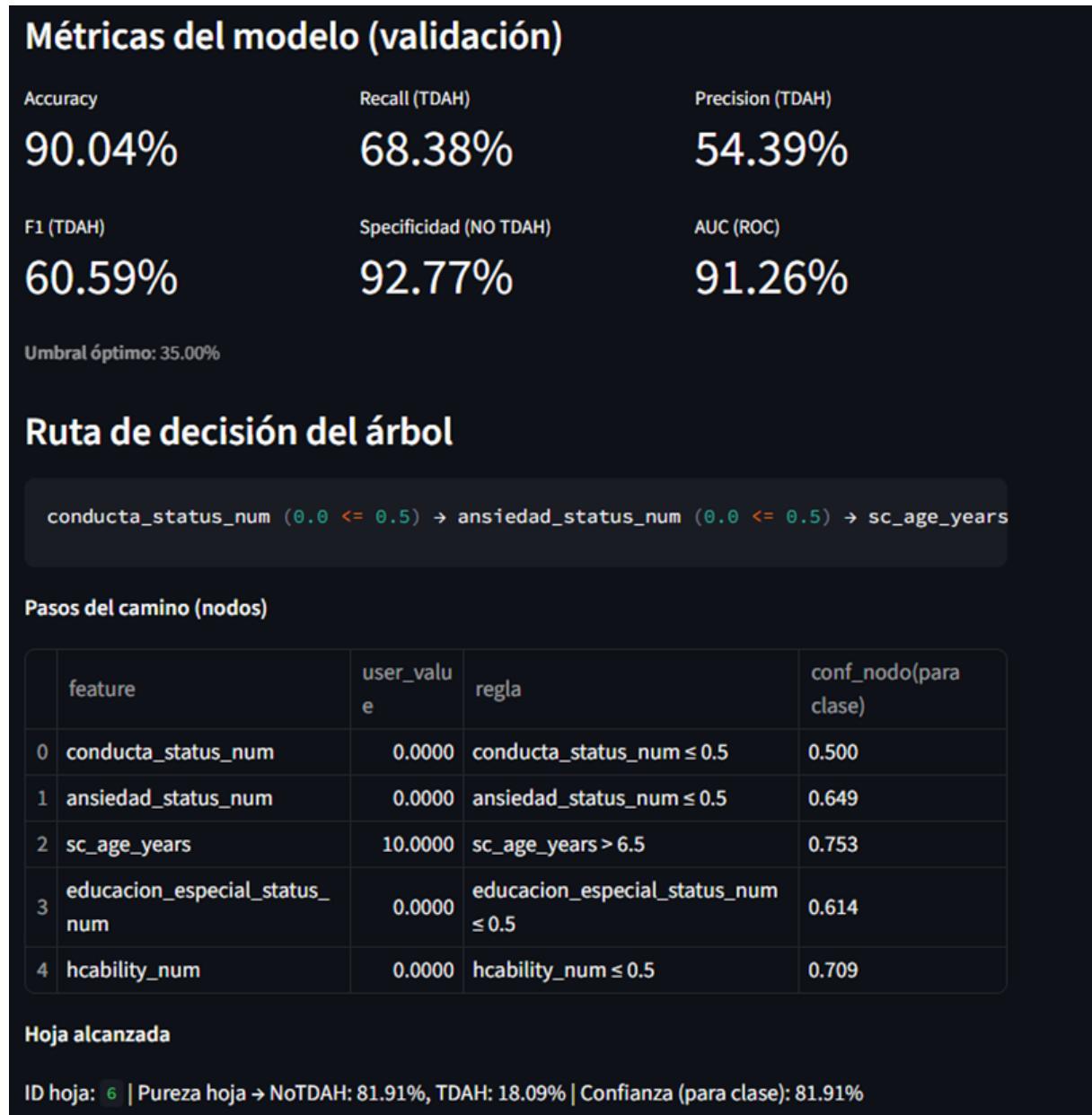


Figura 7.12: Ruta del árbol: métricas del modelo, pasos de decisión y purezas no calibradas.

Captura de TDAHTool.

Pureza vs. calibración (coherencia de cifras). Se observa que la probabilidad final *calibrada* difiere de la *pureza* de la hoja. Esto resulta esperable: la pureza proviene del árbol base (potencialmente sesgada por *undersampling* y *class_weight*), mientras que la cifra operativa se obtiene con `CalibratedClassifierCV` ($cv = 5$), que calibra y promedia:

$$p_{\text{final}} = \frac{1}{5} \sum_{i=1}^5 p_{\text{calibrada}}^{(i)}$$

Adicionalmente, para la explicación de la ruta se muestra la probabilidad calibrada del *fold* cuyo árbol genera dicha ruta, alineando las reglas visualizadas con un calibrador real del sistema. En síntesis, la pestaña *Formulario* no solo valida que la aplicación predice correctamente casos positivos y negativos, sino que también demuestra operativamente la explicabilidad local: se presenta la ruta de decisión y se contextualizan las probabilidades, diferenciando entre *pureza* (no calibrada) y *probabilidades calibradas* usadas en producción.

float placeins subcaption

Pestaña 'Explorar' (Análisis 'what-if')

Esta pestaña (mencionada en el requisito **RF4**) proporciona una herramienta de análisis de sensibilidad o *what-if*. Su objetivo es permitir al usuario explorar interactivamente cómo responde la predicción de riesgo de TDAH a cambios en las variables de entrada.

Para demostrar su funcionamiento, se parte de un caso base identificado como negativo (sin TDAH). A continuación, se modifican iterativamente tres de las variables clave del modelo (dificultad para terminar tareas, problemas de ansiedad y de conducta), ajustando sus valores hacia escenarios más problemáticos, comúnmente asociados al TDAH. Las Figuras 7.13 - 7.17 documentan este proceso.

Formulario Explorar Modelo

Mover controles y visualizar cómo cambia la probabilidad

E_Estado de conducta del paciente: Nunca diagnosticado

E_Dificultad para cuidar al paciente: Nunca

E_Edad del paciente (años): 10

E_El paciente discute mucho: A veces

E_Año de nacimiento: 2015

E_Estado de ansiedad en el paciente: Nunca diagnosticado

E_Persistencia del paciente para finalizar las tareas: Nunca

E_Estado de servicios de educación especial: Nunca ha tenido plan especial de edu...

E_Dificultad para hacer amigos: Mucha dificultad

E_Condiciones de salud presentes en el paciente: El menor no presenta problemas de salud

E_Sexo del paciente: Femenino

E_Tiempo de juego del paciente (entre semana): Muy joven (<3 años)

Calcular y registrar punto

Reset historial

Punto registrado: P(TDAH) = 7.96% | P(No TDAH) = 92.04%

Variable	Opción Selección
E_Estado de conducta del paciente	Nunca diagnosticado
E_Dificultad para cuidar al paciente	Nunca
E_Edad del paciente (años)	10
E_El paciente discute mucho	A veces
E_Año de nacimiento	2015
E_Estado de ansiedad en el paciente	Nunca diagnosticado
E_Persistencia del paciente para finalizar las tareas	Nunca
E_Estado de servicios de educación especial	Nunca ha tenido plan especial de edu...
E_Dificultad para hacer amigos	Mucha dificultad
E_Condiciones de salud presentes en el paciente	El menor no presenta problemas de salud
E_Sexo del paciente	Femenino
E_Tiempo de juego del paciente (entre semana)	Muy joven (<3 años)

Figura 7.13: Pestaña Explorar: Formulario inicial (Caso Base).

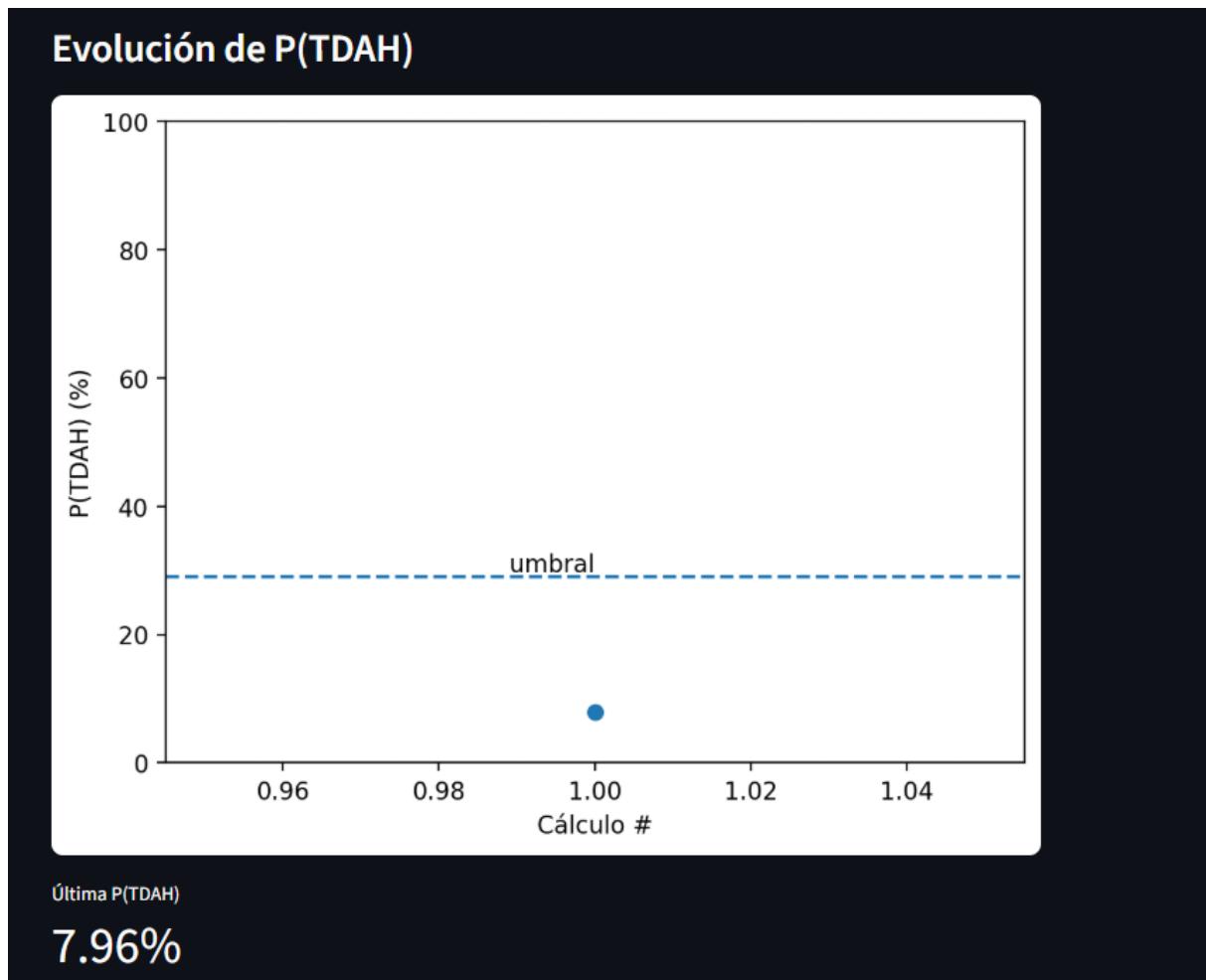


Figura 7.14: Primera ejecución (Caso Base Negativo).

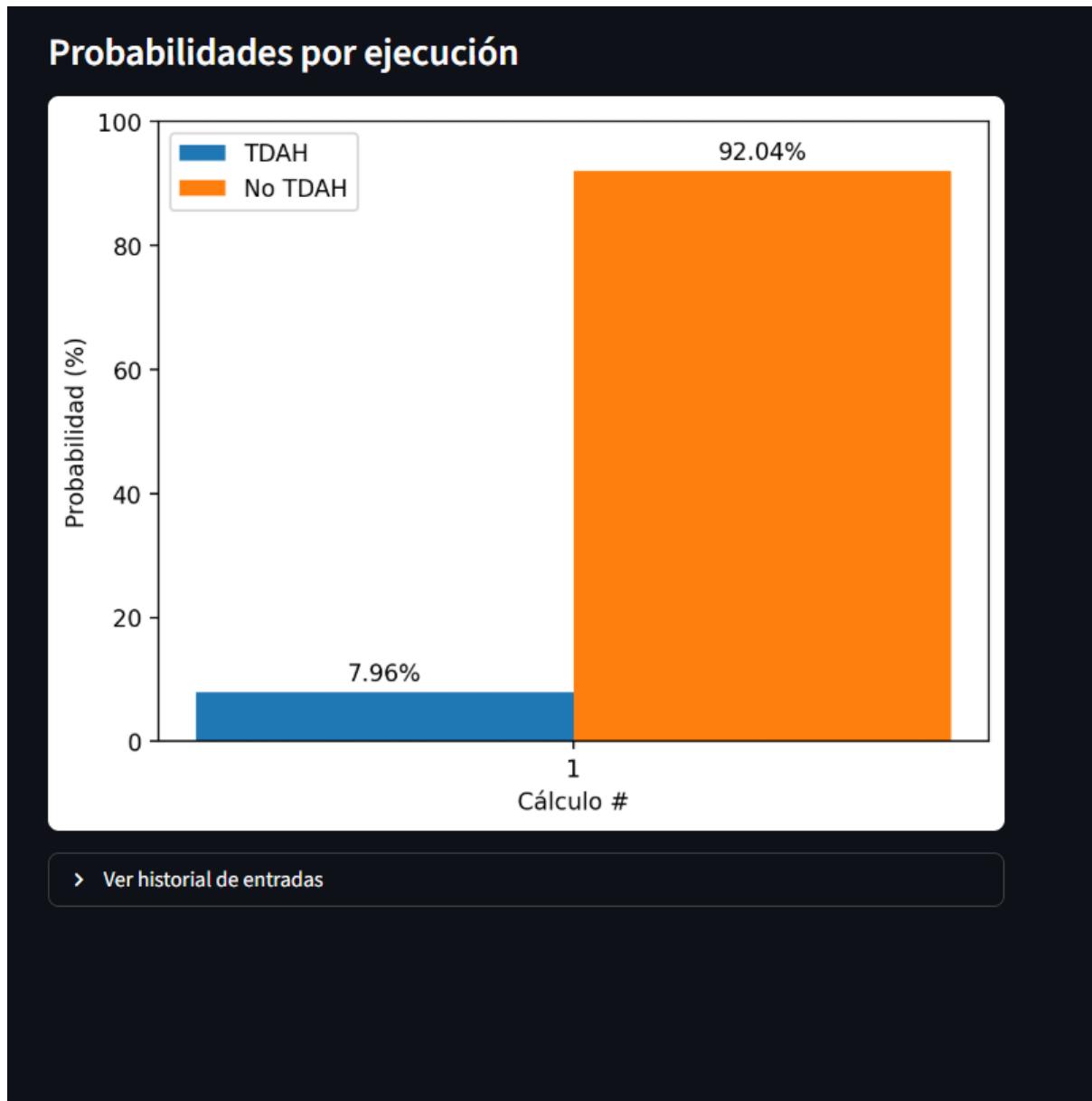


Figura 7.15: Detalle de la primera ejecución (Probabilidades).

Como se aprecia en el gráfico evolutivo (Figura 7.16), a medida que se ajustan las variables clave, la probabilidad de TDAH predicha por el modelo muestra una tendencia al alza y cruza eventualmente el umbral de decisión hacia un resultado positivo. El historial (Figura 7.17) mantiene la trazabilidad de estas simulaciones. Esta funcionalidad no solo valida la reactividad del modelo, sino que ofrece una herramienta didáctica sobre el peso de los distintos factores de riesgo.

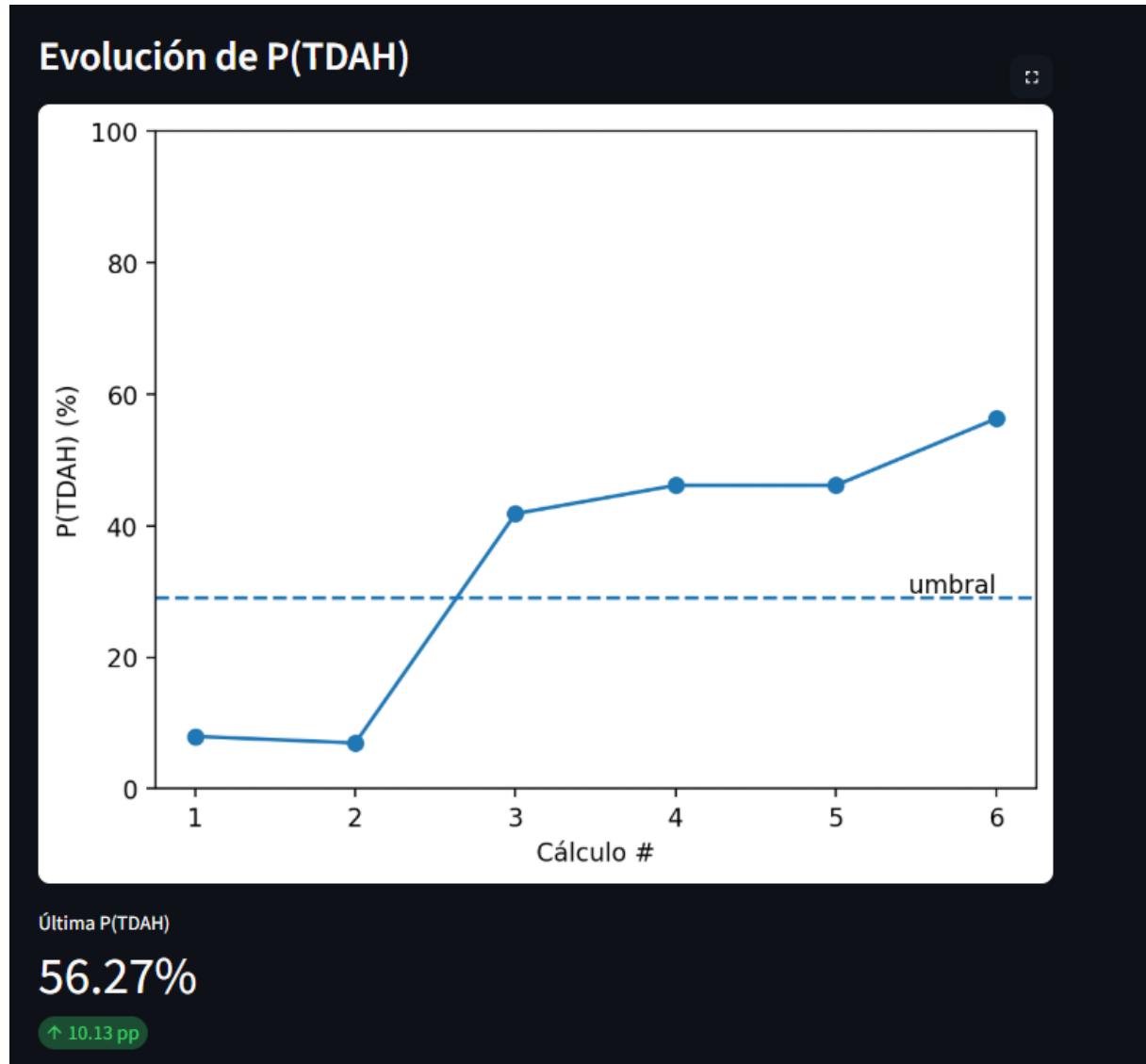


Figura 7.16: Gráfico evolutivo tras 6 ejecuciones (Tendencia a Positivo).

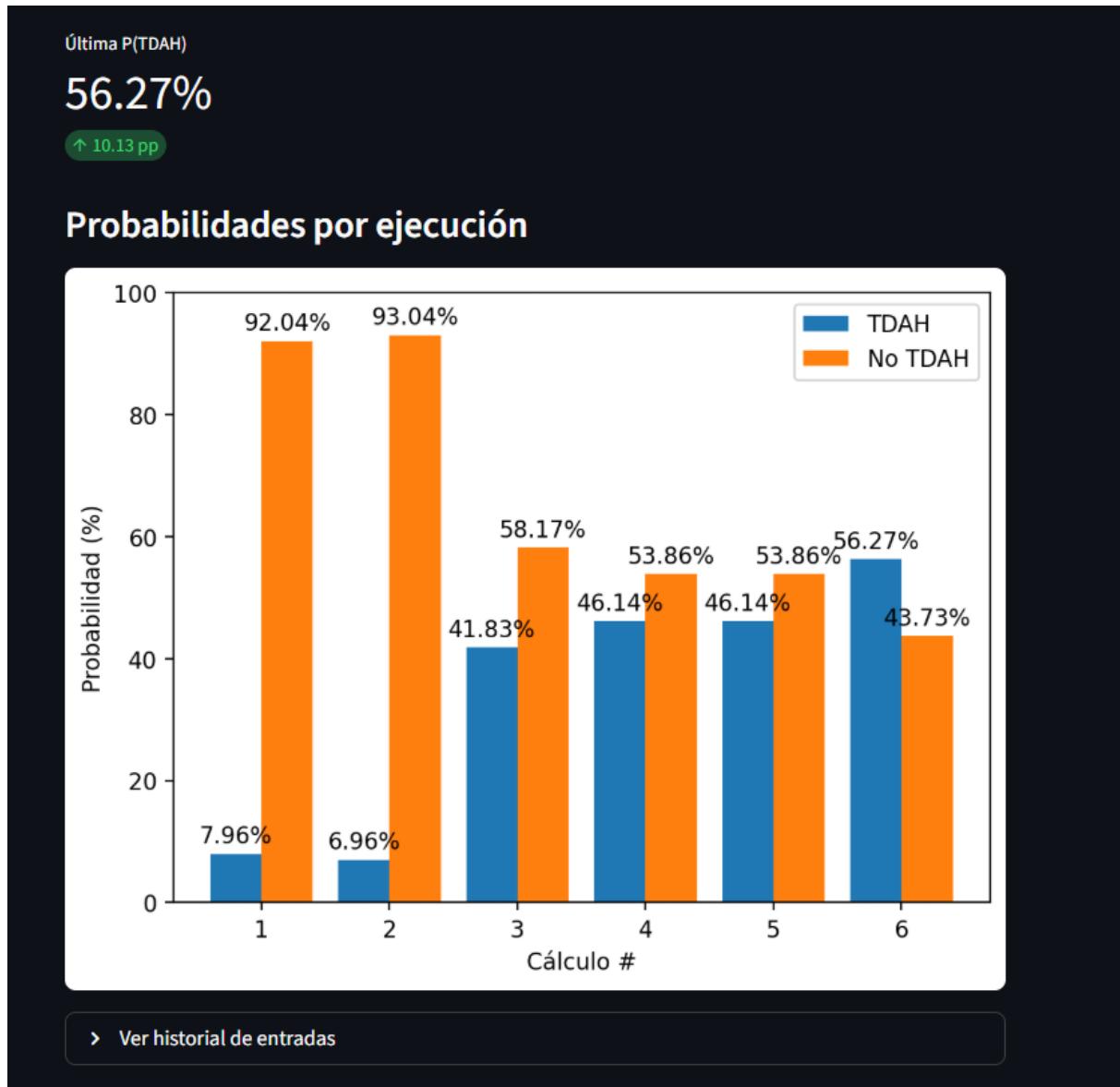


Figura 7.17: Historial de ejecuciones y probabilidades.

7.4. Software, entornos y control de versiones

Con el objetivo de garantizar portabilidad y mantenibilidad en el **desarrollo de la aplicación**, se documentan a continuación las plataformas, *frameworks*/librerías de la capa de aplicación y las herramientas de control de versiones.

7.4.1. Plataformas de desarrollo

La Tabla 7.1 muestra las plataformas utilizadas.

Tabla 7.1: Plataformas de desarrollo utilizadas en la aplicación.

Plataforma	Versión	Propósito
Visual Studio Code	1.104.3	Editor principal; depuración e integración con Git.
Docker + Docker Compose	27.1.1	Empaquetado y orquestación de servicios para despliegue reproducible.

7.4.2. Frameworks y librerías de la aplicación

Tabla 7.2: Stack de la aplicación (backend y frontend).

Componente	Versión	Propósito
FastAPI	0.119.1	Framework ASGI para la API (endpoints, validación, OpenAPI).
Uvicorn	0.38.0	Servidor ASGI para ejecución de la API.
Pydantic	2.12.3	Modelado/validación de entrada y salida (esquemas).
Requests	2.32.5	Cliente HTTP desde el frontend hacia la API.
Streamlit	1.50.0	Interfaz web interactiva y visualización de resultados.

7.4.3. Control de versiones

Tabla 7.3: Herramientas de control de versiones en la aplicación.

Herramienta	Versión	Propósito
Git (repositorio en GitHub)	2.46.0	Control de versiones y CI/CD sobre repositorio remoto (véase Tabla 6.16).

Estructura raíz repositorio

En la Figura 7.4 se presenta el repositorio público utilizado, su estructura y el resumen funcional de los notebooks.

El repositorio se organiza en carpetas temáticas que separan el código de backend y el frontend de la aplicación.

- **Raíz del proyecto.** Se incluye la orquestación con `docker-compose.yml`, el registro de cambios `changelog.md` y la guía de uso `README.md`. La carpeta `.vscode/` contiene ajustes del editor para un entorno de desarrollo consistente.

Tabla 7.4: Repositorio GitHub del proyecto (URL pública).

Elemento	Contenido
URL	https://github.com/dcarmor99/TDAHTool/tree/develop

- **backend/**. Se aloja la implementación del servicio de predicción e inferencia del modelo así como su explicabilidad. La API (FastAPI) con sus contratos, la lógica de preprocesamiento y predicción, los artefactos del modelo (pkl/metrics) y la configuración de dependencias y contenedorización (requirements/Dockerfile).
- **frontend/**. Se aloja la capa de presentación: app Streamlit que orquesta formularios y visualizaciones, consume la API vía HTTP, muestra probabilidades/métricas y artefactos explicativos del modelo, junto con su despliegue contenedorizado.

Capítulo 8

Discusión, conclusiones y trabajos futuros

En este capítulo se sintetizan los hallazgos principales del proyecto, se discuten sus implicaciones prácticas y se proponen líneas realistas de continuidad.

8.1. Conclusiones generales

Las conclusiones se apoyaron en la aplicación de un *pipeline* de *ML* sobre la encuesta NSCH 2023, poniendo el énfasis en rendimiento, interpretabilidad y robustez. El enfoque (con separación temprana del *test*, tratamiento explícito de los *logical skips*, calibración probabilística y ajuste del umbral de decisión) se comportó de forma estable en datos no vistos. En conjunto, se sugirió una buena capacidad de generalización y un equilibrio razonable entre la detección de positivos y el control de falsos positivos.

En cuanto a la selección de variables, el experimento con tres escenarios mostró que un conjunto compacto de 51 variables ofreció prácticamente el mismo rendimiento que el conjunto completo (87), mientras que el paquete de 24 variables definidas por el “experto” se quedó corto bajo este *pipeline*. La mezcla de literatura/criterio clínico con EDA y validación empírica resultó, por tanto, el mejor punto medio entre rendimiento, interpretabilidad y mantenibilidad.

La herramienta **TDAHTool** permitió, además, aterrizar el modelo a situaciones reales —tanto docentes como de atención primaria— con predicción por instancia y análisis exploratorio apoyado en explicaciones. Todo ello se enmarcó en un proceso transparente, automatizable y orientado al despliegue, capaz de integrar dimensiones clínicas, familiares, conductuales y socioeconómicas para detectar patrones asociados a TDAH en población infantil. Finalmente, la poda por coste simplificó el árbol sin sacrificar métricas, y la combinación de interpretabilidad, probabilidades calibradas y umbral óptimo facilitó tanto la trazabilidad clínica/educativa como su transferencia operativa.

8.2. Líneas de trabajo futuro

Como línea de continuidad inmediata de la aplicación desarrollada, quedó pendiente la incorporación de funcionalidades orientadas al soporte del cuidado: un módulo de recordatorios de medicación configurable por cuidadores y profesionales, con registro de adherencia y sistema de alertas; un componente de recomendaciones personalizadas de actividades seguras en el hogar, basadas en el perfil de riesgo y la sintomatología, incluyendo pautas de supervisión; y mecanismos de derivación ágil a atención médica especializada (generación de informes estandarizados y vías de contacto).

A corto y medio plazo, se consideró prioritario realizar una validación externa con datos locales. Replicar el estudio en muestras españolas o europeas comparables permitiría valorar la validez externa y ajustar, si fuese necesario, a las particularidades del contexto (lengua, cultura y los propios sistemas educativo y sanitario).

También se consideró deseable ampliar el abanico de señales y modalidades. Incluir información adicional (historial académico objetivo, apoyos educativos, métricas de sueño y actividad, EEG) y, cuando fue ético y viable, fuentes multimodales (tiempo de pantalla real, patrones de uso de dispositivos, escalas estandarizadas) podría ayudar a incrementar el *recall* sin penalizar la precisión.

En paralelo, se recomendó repetir el *pipeline* con otros modelos (XGBOOST, SVM, MLP y redes neuronales), manteniendo la misma disciplina de separación de *test*, calibración y ajuste de umbral, para medir la ganancia real frente al árbol de decisión final. Cobró importancia, además, reforzar el marco ético y de equidad: auditar sesgos por subgrupos (sexo, nivel socioeconómico, estructura familiar), reportar métricas desagregadas y estudiar medidas como la reponderación de pérdidas o técnicas *fairness-aware* que redujesen disparidades sin perder validez clínica.

Por último, se consideró clave trabajar la usabilidad y la explicabilidad con usuarios finales. Estudios con docentes, equipos de orientación y atención primaria ayudarían a medir la comprensión de las rutas del árbol, la utilidad del análisis *what-if* y la claridad de las métricas, y servirían para iterar la interfaz de la aplicación con base en entrevistas y encuestas.

8.3. Discusión

En esta sección se lleva a cabo una comparación con el estado del arte (Sección 8.3.1) así como una discusión metodológica (Sección 8.3.2).

8.3.1. Comparación con el estado del arte

Este trabajo se sitúa en la línea que aplica *ML* a datos poblacionales que proceden de encuestas relacionadas con la salud de la infancia en población infantil y adolescente, concretamente para este proyecto se ha hecho uso de la NSCH.

Como estudio de referencia se considera el trabajo de Maniruzzaman Maniruzzaman et al. (2022), que emplea NSCH (2018-2019) y combina una criba por **Regresión Logística** ($p < 0,05$) con la comparación de ocho clasificadores, destacando finalmente un **Random Forest** tras *grid search*; el desbalance se maneja mediante una combinación de *oversampling* y *undersampling* hasta alcanzar un balance aproximado 1:1.

En contraste, la presente propuesta actualiza la fuente a NSCH 2023 y prioriza un marco de evaluación **realista**: no se fuerza el rebalanceo del conjunto de evaluación; el desbalance se

gestiona con `class_weight`, **calibración probabilística** y **ajuste explícito de umbral** sobre la distribución original, manteniendo un *test* separado del conjunto inicial e independiente del subconjunto de entrenamiento para mitigar el *leakage* y evaluar el modelo entrenado.

Selección de variables En el estudio de referencia, la selección se realiza con **Regresión Logística** tras balancear las clases, reteniendo predictores con $p < 0,05$. En la propuesta actual se adopta una **selección híbrida** que combina literatura y criterio clínico con correlaciones por tipo de variable e **importancia del modelo**, manteniendo trazabilidad de mapeos y evitando variables con fuga de información.

Gestión del desbalance En Maniruzzaman, el **rebalanceo artificial** (sobre/submuestreo) se utiliza para formar un conjunto balanceado previo al modelado. En el presente trabajo, el desbalance se **internaliza** en la función de pérdida mediante `class_weight=balanced`, evitando datos sintéticos, como los que generaría técnicas como *SMOTE* y preservando la distribución original para la evaluación.

Modelo e interpretabilidad El trabajo de referencia prioriza **Random Forest** por rendimiento, a costa de menor trazabilidad para el usuario final. En esta propuesta se prioriza un **Árbol de Decisión calibrado** con **ruta explicada** por caso y **probabilidades calibradas**, favoreciendo la interpretabilidad local y la adopción en contextos educativos y clínicos.

Calibración y umbral operativo Se priorizan la utilidad clínica y la interpretabilidad del sistema. Tras una fase inicial de selección de variables guiada por la importancia de un árbol preliminar entrenado sobre datos balanceados mediante *Random UnderSampling*, se aplica un procedimiento iterativo: en cada iteración se reentrena el árbol con el conjunto de variables candidatas, se restringen los conjuntos de entrenamiento y validación a las características efectivamente utilizadas y se repite hasta estabilizar un subconjunto consistente y explicativo. A continuación, se ajustan hiperparámetros con `GridSearchCV` y validación cruzada estratificada (5 particiones), optimizando el *recall* de la clase positiva para reducir falsos negativos.

Una vez fijada la estructura del modelo, las probabilidades se calibran con `CalibratedClassifierCV` y validación cruzada, entrenando sobre los datos originales (sin *undersampling*) con el fin de corregir el sesgo típico de los árboles a producir probabilidades mal ajustadas. Este paso permite obtener estimaciones probabilísticas más fiables para la toma de decisiones clínicas.

El umbral de decisión no se mantiene en 0,5. En su lugar, se selecciona en función de la curva *precision-recall*: para cada $\tau \in [0, 1]$ se calculan precisión, *recall* y F_1 de la clase TDAH, eligiéndose el τ que maximiza F_1 positivo. Este criterio proporciona un equilibrio operativo entre detección y falsos positivos, priorizando sensibilidad: un falso positivo conlleva una revisión adicional, mientras que un falso negativo puede retrasar diagnóstico e intervención, lo cual se considera crítico en el presente trabajo.

Protocolo de evaluación En la referencia se trabaja con NSCH 2018–2019 y se balancea la clase antes del modelado, reportándose las métricas bajo ese régimen. En este proyecto se mantiene un **subconjunto de prueba independiente** (*holdout*) separado al inicio del estudio y **no utilizado** durante el diseño del *pipeline* ni en la selección de variables, ni en el ajuste de hiperparámetros, ni en la calibración, ni en la elección del umbral.

El pipeline completo (preprocesado, modelo, calibración, umbral, entrenamiento) se ajusta únicamente con los datos de entrenamiento mediante validación cruzada estratificada, garantizando que cualquier estimación intermedia se obtiene sin acceso al conjunto de prueba. La evaluación final se realiza una única vez sobre el *holdout* inicial, evitando reentrenos o reuso de información y reduciendo el optimismo en la estimación de generalización. Con este protocolo, se minimiza el riesgo de *leakage* y se asegura que las métricas reflejan el comportamiento esperado en despliegue.

Operacionalización y aplicación web El estudio de referencia se centra en análisis y comparación de clasificadores sobre microdatos. La presente propuesta **materializa** el pipeline en una **API (FastAPI)** y una **UI (Streamlit)** con **análisis what-if** y **visualización de ruta**, desplegables con **Docker Compose** para portabilidad y reproducibilidad, acercando el sistema al punto de decisión.

8.3.2. Discusión metodológica y conclusiones experimentales

Se propone un **marco riguroso y reutilizable** para futuras ediciones de la NSCH y estudios afines, que combina criterio experto, evidencia clínica, EDA y validación empírica, priorizando modelos interpretables. En los experimentos comparativos por conjuntos de variables, el conjunto completo (87) no aporta mejoras sustanciales y el conjunto experto (24) pierde señal; por el contrario, la **selección híbrida** (en torno a 51 variables) ofrece el mejor equilibrio entre rendimiento e interpretabilidad, reduciendo ruido y favoreciendo la robustez del pipeline. Dado el carácter desbalanceado del problema clínico, se priorizan el **recall de la clase positiva** y el **F1 positivo**, utilizando F1 como resumen del equilibrio entre sensibilidad y precisión para evitar la saturación por falsos positivos. La gestión del desbalance combina un *undersampling* preliminar —empleado para estimar importancias sin dominancia de la clase mayoritaria— con **class_weight='balanced'** en el modelo final y con **calibración probabilística y ajuste de umbral** sobre la distribución original. Tras calibrar, el punto operativo se selecciona en la curva **precision-recall** maximizando el **F1 positivo**, de modo que se equilibra la detección de casos con un control razonable de falsos positivos clínicamente no útiles.

8.4. Limitaciones del estudio

La principal limitación de datos reside en la **falta de una base española específica de TDAH** disponible en plazo; en consecuencia, se utiliza NSCH 2023 (EE. UU.), lo que aporta una muestra amplia y bien documentada pero condiciona la **valididad externa** hasta contar con replicaciones locales. La **evidencia disponible** en la literatura sobre detección temprana mediante ML resulta limitada —por ejemplo, trabajos de Maniruzzaman, Slobodin o Zakani— y a menudo presenta tamaños muestrales reducidos o menor atención al preprocesado y a la validación, dificultando comparaciones directas y robustas. Además, la **naturaleza de autoinforme** de la NSCH, basada en cuestionarios a cuidadores o tutores, introduce posibles **sesgos culturales y personales** que pueden afectar la calidad y estabilidad de las señales observadas, por lo que se recomienda interpretar los resultados en ese marco y reforzarlos con evaluaciones complementarias en contextos clínicos y educativos locales.

8.5. Reflexión crítica: tecnología, concentración y educación

La identificación de variables predictivas —como la persistencia al finalizar tareas— subrayó la relevancia de la atención sostenida y la autorregulación en el desarrollo infantil. En este contexto, las nuevas tecnologías, especialmente el *smartphone*, actuaron como fuente constante de estímulos y distracciones, con potencial impacto en la capacidad de concentración. Se vivió en entornos de alta inmediatez que premiaron la novedad y penalizaron la inactividad. Esta dinámica chocó con la necesidad de cultivar atención profunda para el rendimiento académico y la autorregulación emocional.

A ello se sumó el papel emergente de la inteligencia artificial (IA). Aunque se trató de una herramienta poderosa para personalizar el aprendizaje y ampliar el acceso a recursos, su uso acrítico o sin guía pudo erosionar el desarrollo del pensamiento crítico y la motivación por la investigación propia, competencias esenciales en la formación de ingenieros y científicos. Además, la percepción de que la IA “sustituye” al personal docente pudo disminuir la implicación del alumnado en el aula y convertirse en un foco adicional de desatención. Este riesgo resultó especialmente sensible en estudiantes con TDAH, para quienes el incremento de distractores y la delegación excesiva de procesos cognitivos en la IA pudieron agravar dificultades atencionales y mermar el rendimiento académico.

En consecuencia, en entornos educativos se propuso limitar usos tecnológicos que fomentasen la distracción y promover actividades que reforzasen el razonamiento crítico, la indagación autónoma y el uso ético y explicado de la IA. Priorizar profundidad sobre inmediatez, y explicar cómo y por qué se utilizaron estas herramientas, se alineó con los objetivos de este trabajo y con la transferencia de herramientas explicables al aula y a la práctica clínica.

Bibliografía

- Abdelnour, E., Jansen, M. O., & Gold, J. A. (2022). ADHD Diagnostic Trends: Increased Recognition or Overdiagnosis? *Missouri Medicine*, 119(5), 467-473.
- American Psychiatric Association. (2013). *Diagnostic and Statistical Manual of Mental Disorders* (5.^a ed.). American Psychiatric Publishing.
- Association, A. P. (2013). *Diagnostic and Statistical Manual of Mental Disorders* (5.^a ed.). American Psychiatric Publishing.
- AUC ROC Curve in Machine Learning* [Last updated: 29 Oct 2025]. (2025, 29 de octubre). GeeksforGeeks. Consultado el 10 de noviembre de 2025, desde <https://www.geeksforgeeks.org/machine-learning/auc-roc-curve/>
- Bordin, I. A., Rocha, M. M., Paula, C. S., Teixeira, M. C. T., Achenbach, T. M., Rescorla, L. A., & Silvares, E. F. (2013). Child Behavior Checklist (CBCL), Youth Self-Report (YSR) and Teacher's Report Form (TRF): an overview of the development of the original and Brazilian versions. *Cadernos de saúde pública*, 29, 13-28.
- Catalá-López, F., Peiró, S., Ridao, M., Sanfélix-Gimeno, G., Gènova-Maleras, R., & Catalá, M. A. (2012). Prevalence of attention deficit hyperactivity disorder among children and adolescents in Spain: a systematic review and meta-analysis of epidemiological studies. *BMC Psychiatry*, 12, 168. <https://doi.org/10.1186/1471-244X-12-168>
- Centers for Disease Control and Prevention. (2023). *What are developmental disabilities?* [Accedido el 15 de enero de 2025]. <https://www.cdc.gov/ncbddd/developmentaldisabilities/index.html>
- Checa Fernández, P. (s.f.). *Purificación Checa Fernández* [Perfil personal]. Universidad de Granada. Consultado el 23 de octubre de 2025, desde <https://www.ugr.es/personal/purificacion-checha-fernandez>
- Conners, K. C. (2000). *Conners' Continuous Performance Test II (CPT II)*. Multi-Health Systems.
- diagrams.net. (2025). *diagrams.net (formerly draw.io)* [Computer software]. Consultado el 14 de octubre de 2025, desde <https://www.diagrams.net/>
- Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, 55(10), 78-87. <https://doi.org/10.1145/2347736.2347755>
- Faraone, S. V., & Larsson, H. (2019). Genetics of attention deficit hyperactivity disorder. *Molecular Psychiatry*, 24(4), 562-575. <https://doi.org/10.1038/s41380-018-0070-0>
- GeeksforGeeks. (2025, octubre). What is Exploratory Data Analysis? [Last updated: 14 Oct 2025]. Consultado el 19 de octubre de 2025, desde <https://www.geeksforgeeks.org/data-analysis/what-is-exploratory-data-analysis/>
- Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2.^a ed.). O'Reilly Media.
- Gioia, G. A., Isquith, P. K., Guy, S. C., & Kenworthy, L. (2000). *Behavior Rating Inventory of Executive Function*. Psychological Assessment Resources.

- Kazmierczak, J., Salama, K., & Huerta, V. (2024). MLOps: flujos de procesamiento de entrega continua y automatización en el aprendizaje automático [Última actualización: 2024-08-28 (UTC)]. Otro colaborador: Sunil Kumar Jang Bahadur]. Consultado el 19 de octubre de 2025, desde <https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning?hl=es>
- Kim, G., Humble, J., Debois, P., & Willis, J. (2021). *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations* (2.^a ed.). IT Revolution Press.
- Maniruzzaman, M., Shin, J., & Hasan, M. A. R. (2022). Predicting Children with ADHD Using Behavioral Activity: A Machine Learning Analysis. *Applied Sciences*, 12(5), 2737. <https://doi.org/10.3390/app12052737>
- Millisecond Software. (2025). CPT (Continuous Performance Test) [Accessed: 2025-07-07]. <https://www.millisecond.com/download/library/cpt>
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.
- Overleaf. (2025). Overleaf (Computer software; Ver. v2). <https://www.overleaf.com/>
- Plot Precision–Recall Curve. (s.f.). scikit-learn. Consultado el 10 de noviembre de 2025, desde https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html
- Polanczyk, G., de Lima, M. S., Horta, B. L., Biederman, J., & Rohde, L. A. (2007). The worldwide prevalence of ADHD: a systematic review and metaregression analysis. *The American Journal of Psychiatry*, 164(6), 942-948. <https://doi.org/10.1176/ajp.2007.164.6.942>
- Rivera, F. B. (2016). La elevada prevalencia del TDAH: posibles causas y repercusiones socio-educativas. *Psicología Educativa*, 22(2), 81-85.
- Slobodin, O., Yahav, I., & Berger, I. (2020). A machine-based prediction model of ADHD using CPT data. *Frontiers in human neuroscience*, 14, 560021.
- Sommerville, I. (2015). *Software Engineering* (10th). Pearson.
- Starmer, J. (2022). *The StatQuest Illustrated Guide to Machine Learning* (1^a Edición). StatQuest Press.
- StataCorp LLC. (s.f.). *Stata: Software for Statistics and Data Science*. Consultado el 24 de julio de 2025, desde <https://www.stata.com/>
- U.S. Census Bureau. (2023). National Survey of Children's Health (NSCH) - 2023 Datasets [Accessed July 18, 2025]. <https://www.census.gov/programs-surveys/nsch/data/datasets/nsch2023.html>
- U.S. National Library of Medicine. (2023). *Trastorno por Déficit de Atención con Hiperactividad (TDAH)* [Información médica revisada sobre TDAH y otros trastornos del neurodesarrollo]. <https://medlineplus.gov/spanish/attentiondeficithyperactivitydisorder.html>
- van Rossum, G., Warsaw, B., & Coghlan, N. (2001). PEP 8 – Style Guide for Python Code [Python Enhancement Proposal].
- Venkat. (2017, noviembre). CRISP-DM. Consultado el 19 de octubre de 2025, desde <https://dslytics.wordpress.com/2017/11/15/crisp-dm/>
- Zakani, Z., Moradi, H., Ghasemzadeh, S., Riazi, M., & Mortazavi, F. (2023). The Validity of a Machine Learning-Based Video Game in the Objective Screening of Attention Deficit Hyperactivity Disorder in Children Aged 5 to 12 Years. *arXiv preprint arXiv:2312.11832*.
- Zhou, Z.-H. (2021). *Machine Learning* (1st). Springer. <https://doi.org/10.1007/978-981-15-8288-5>

Apéndices

Apéndice A

Variables utilizadas en el modelo

A continuación, se presenta la tabla con las variables seleccionadas para el entrenamiento del modelo de aprendizaje automático:

Tabla A.1: Variables seleccionadas para entrenamiento del modelo

Nombre de variable	Descripción	Categoría
sc_k2q22_bin	Necesita tratamiento conductual	Binaria
memorycond_bin	Dificultad para concentrarse o recordar	Binaria
k4q23_bin	Medicación para emociones	Binaria
allergies_bin	Alergias	Binaria
concussion_bin	Conmoción cerebral	Binaria
k2q40a_bin	Asma	Binaria
a1_born_bin	Nacimiento del menor	Binaria
birthwt_1_bin	Prematuro	Binaria
autoimmune_bin	Enfermedad autoinmune	Binaria
ace9_bin	Convivencia con drogadicto o alcohólico	Binaria
headache_bin	Dolores de cabeza	Binaria
grades_Mostly D's or lower	Notas suspensas	Categórica
family_r_Two parents (at least one not biological/adoptive), not currently married	Familia mixta no casada	Categórica
family_r_Two parents (at least one not biological/adoptive), currently married	Familia mixta casada	Categórica
family_r_Two biological/adoptive parents, not currently married	Padres biológicos/adoptivos no casados	Categórica
grades_This child's school does not give these grades	Sistema distinto de calificación	Categórica
family_r_Single mother	Relación familiar: madre soltera	Categórica
family_r_Two biological/adoptive parents, currently married	Padres biológicos/adoptivos casados	Categórica
fpl_group_num	Ratio de pobreza familiar	Categórica
sc_sex_bin	Sexo del menor	Categórica
grades_Mostly A's and B's	Notas sobresalientes	Categórica
family_r_Other relation	Relación familiar: otra	Categórica

Nombre de variable	Descripción	Categoría
grades_Mostly C's and D's	Notas suficientes	Categórica
family_r_Single father	Relación familiar: padre soltero	Categórica
grades_Mostly B's and C's	Notas regulares	Categórica
sc_age_years	Edad del menor	Numérica
a1_age	Edad del adulto 1 (madre)	Numérica
birth_yr	Año de nacimiento del adulto 1 (madre)	Numérica
addtreat_clean_num	TDAH con tratamiento psicológico	Ordinal
makefriend_num	Facilidad para hacer amigos	Ordinal
adhd_medicated_num	TDAH con medicación	Ordinal
adhd_status_num	Estado del TDAH	Ordinal
conducta_status_num	Problemas de conducta	Ordinal
ansiedad_status_num	Ansiedad	Ordinal
educacion_especial_status_num	Necesidad de educación especial	Ordinal
depresion_status_num	Depresión	Ordinal
hcability_num	Problemas de salud frecuentes	Ordinal
bullied_r_num	Bullying sufrido	Ordinal
k8q31_num	Dificultad para cuidar al menor	Ordinal
k7q84_r_num	Termina tareas que empieza	Ordinal
k8q32_num	Pérdida de nervios hacia el menor	Ordinal
outdoorswkday_clean_num	Juego entre semana fuera de casa	Ordinal
sharetoys_clean_num	Capacidad para compartir juguetes	Ordinal
outdoorswkend_clean_num	Juego en fin de semana fuera de casa	Ordinal
focuson_clean_num	Capacidad de concentración	Ordinal
screentime_num	Tiempo de pantalla	Ordinal
hardwork_clean_num	Frecuencia con la que trabaja duro	Ordinal
sc_english_clean_num	Idioma que habla el menor	Ordinal
k7q70_r_num	Discute mucho	Ordinal
calmdown_r_clean_num	Capacidad para calmarse	Ordinal
higrade_num	Nivel educativo de los padres	Ordinal
k2q31a_bin	Diagnóstico binario de TDAH	Binaria

Apéndice B

VARIABLES DEL CONJUNTO DE ENTRENAMIENTO FINAL

En esta sección se documentan todas las variables incluidas en el conjunto final de entrenamiento, con su descripción y tipo correspondiente.

Tabla B.1: Descripción y tipo de las variables incluidas en el conjunto de entrenamiento final.

Nombre de variable	Descripción	Categoría
addtreat_clean_num	Tiene TDAH y está en tratamiento psicológico	Ordinal
adhd_medicated_num	Diagnosticado y tomando medicación para TDAH	Ordinal
adhd_status_num	Estado del TDAH en el menor	Ordinal
conducta_status_num	Problemas de conducta del menor	Ordinal
ansiedad_status_num	Ansiedad en el menor	Ordinal
educacion_especial_status_num	Necesidad de atención educativa especial	Ordinal
depresion_status_num	Depresión en el menor	Ordinal
hcability_num	Problemas de salud que limitan la actividad diaria	Ordinal
bullied_r_num	Bullying sufrido por el menor	Ordinal
k8q31_num	Dificultad para cuidar al menor	Ordinal
k8q32_num	Sensación de pérdida de nervios hacia el menor	Ordinal
outdoorswkday_clean_num	Tiempo al aire libre entre semana	Ordinal
sharetoys_clean_num	Capacidad para compartir juguetes	Ordinal
outdoorswkend_clean_num	Tiempo al aire libre en fin de semana	Ordinal
focuson_clean_num	Capacidad de concentración y finalización de tareas	Ordinal
screentime_num	Tiempo de pantalla diario	Ordinal
hardwork_clean_num	Perseverancia ante tareas difíciles	Ordinal
sc_english_clean_num	Nivel de inglés del menor	Ordinal
k7q70_r_num	Frecuencia con la que discute el menor	Ordinal
calmdown_r_clean_num	Capacidad para calmarse tras excitación	Ordinal
higrade_num	Nivel educativo del adulto responsable	Ordinal
makefriend_num	Facilidad del menor para hacer amistades	Ordinal
k7q84_r_num	Perseverancia en finalizar tareas	Ordinal
k4q23_bin	Toma medicación para regular emociones o conducta	Binaria

Nombre de variable	Descripción	Categoría
memorycond_bin	Dificultades para concentrarse, recordar o decidir	Binaria
sc_k2q22_bin	Necesita tratamiento emocional o de desarrollo	Binaria
ace9_bin	Convivencia con personas con adicciones	Binaria
headache_bin	Dolores de cabeza frecuentes	Binaria
allergies_bin	Presencia de alergias	Binaria
concussion_bin	Conmoción cerebral previa	Binaria
k2q40a_bin	Diagnóstico de asma	Binaria
a1_born_bin	Año de nacimiento del menor (umbral)	Binaria
birthwt_l_bin	Bajo peso al nacer	Binaria
autoimmune_bin	Enfermedad autoinmune diagnosticada	Binaria
fpl_group_num	Ratio de pobreza familiar (grupos)	Categórica
sc_sex_bin	Sexo del menor	Categórica
grades_Mostly A's and B's	Rendimiento académico alto	Categórica
grades_Mostly B's and C's	Rendimiento académico medio	Categórica
grades_Mostly C's and D's	Rendimiento académico bajo	Categórica
grades_Mostly D's or lower	Rendimiento académico muy bajo	Categórica
grades_This child's school does not give these grades	Sistema de calificación alternativo	Categórica
family_r_Other relation	Relación familiar no directa	Categórica
family_r_Single father	Familia monoparental (padre)	Categórica
family_r_Single mother	Familia monoparental (madre)	Categórica
family_r_Two biological/adoptive parents, currently married	Padres biológicos/adoptivos casados	Categórica
family_r_Two biological/adoptive parents, not currently married	Padres biológicos/adoptivos no casados	Categórica
family_r_Two parents (at least one not biological/adoptive), currently married	Familia mixta, casados	Categórica
family_r_Two parents (at least one not biological/adoptive), not currently married	Familia mixta, no casados	Categórica
sc_age_years	Edad del menor	Numérica
a1_age	Edad del cuidador principal	Numérica
birth_yr	Año de nacimiento del cuidador principal	Numérica
k2q31a_bin	Diagnóstico de TDAH (variable objetivo)	Binaria

Apéndice C

Entrenamiento modelo Random Forest

Aquí se muestra cómo fue el entrenamiento alternativo de un modelo Random Forest y el rendimiento obtenido.

Las conclusiones a alto nivel:

- Ligera mejora en *recall* y F1.
- Pérdida moderada en *precision* respecto a Decision Tree.
- Aparición de nuevas variables interesantes para futuras líneas clínicas.

Apéndice D

Entrenamiento y validación del Random Forest

El clasificador *Random Forest* se entrenó sobre *train* y se evaluó en validación. La Tabla D.1 resume métricas por clase.

Tabla D.1: Métricas por clase del Random Forest.

Clase	Precisión	Recall	F1
0 (No TDAH)	0.96	0.93	0.94
1 (TDAH)	0.55	0.71	0.62

Interpretación y líneas de mejora. (Texto original sin cambios.)

Apéndice E

Variables dataset decodificadas

En esta sección se documentan todas las variables originales del dataset tras aplicar la decodificación en STATA.

Tabla E.1: Descripción y tipo de las variables incluidas en el conjunto de entrenamiento final.

Nombre de variable	Descripción	Categoría
hhlanguage	Language Spoken at Home	Categórica
sc_age_years	SC Age in Years	Numérica
fpl_i1	Family Poverty Ratio, First Implicate	Numérica

Apéndice F

Diagramas

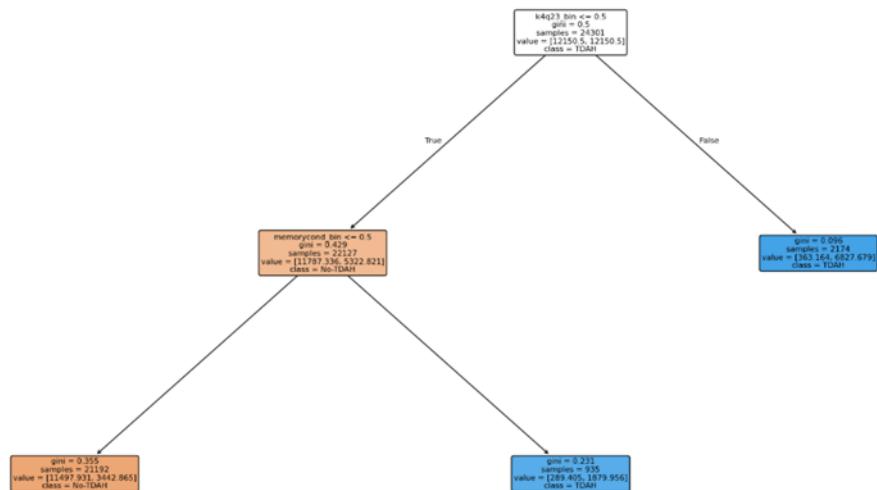


Figura F.1: Árbol de decisión — segunda ejecución. Fuente: elaboración propia.

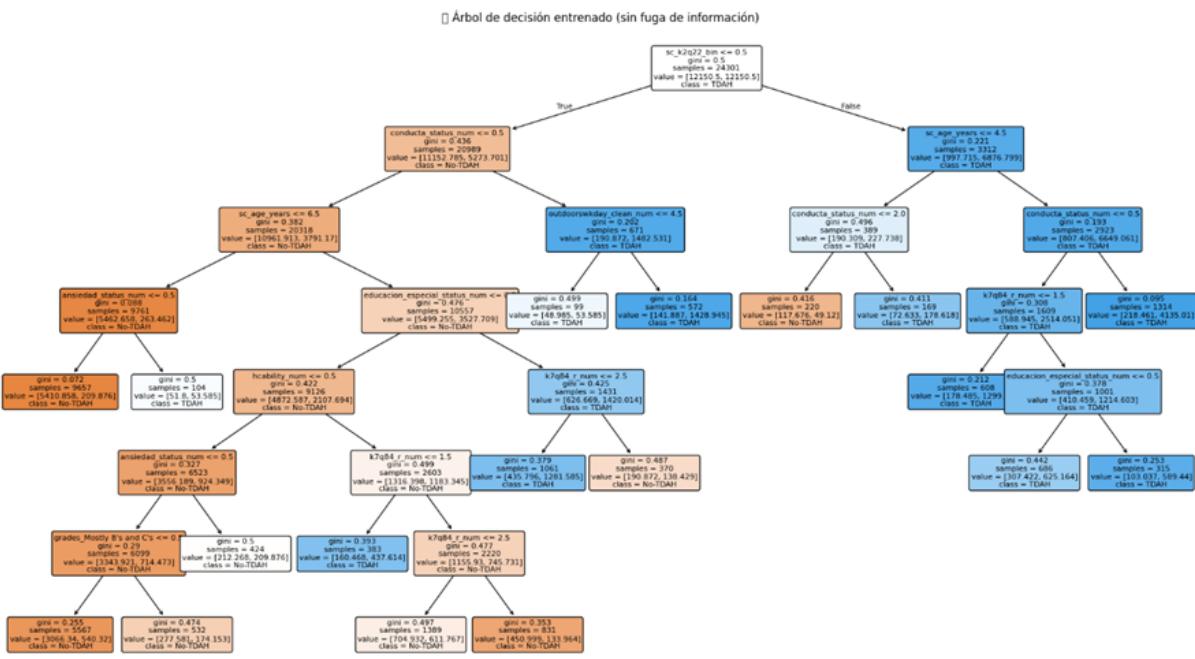


Figura F.2: Árbol de decisión — tercera ejecución. Fuente: elaboración propia.

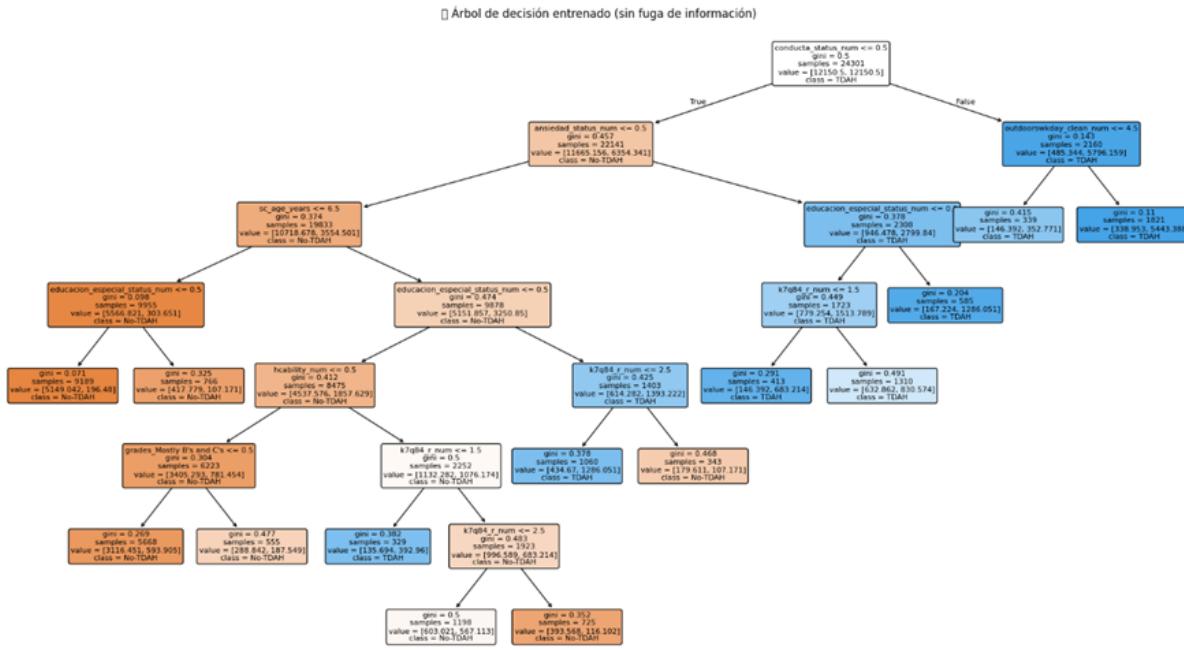


Figura F.3: Árbol de decisión — cuarta ejecución. Fuente: elaboración propia.

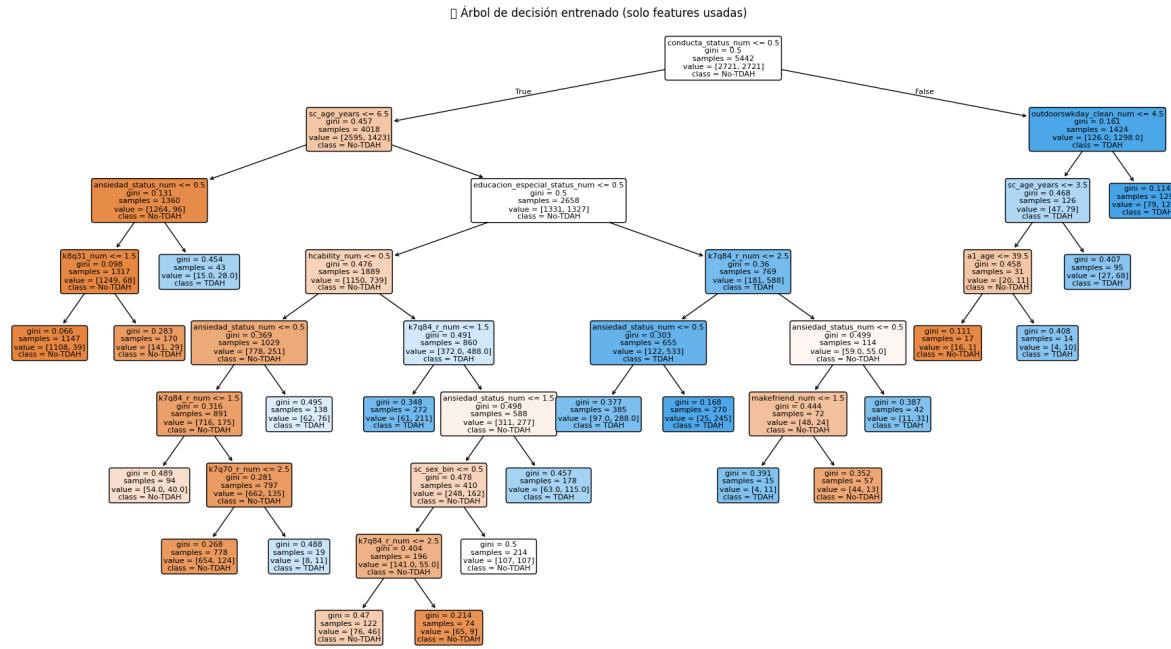


Figura F.4: Árbol de decisión — versión final. Fuente: elaboración propia.