Evaluación jerárquica de modelos de lenguaje en diagnóstico clínico: superando la saturación mediante un pipeline multicapas

Análisis metodológico de PV0-PV4

Autores: Yago Mendoza, Javier Logroño
Institución: Foundation 29
Fecha: 20 de julio de 2025
Versión del Documento: 2.11

Este documento presenta un análisis pormenorizado de los hallazgos y la evolución de los marcos de evaluación para la herramienta de diagnóstico asistido por IA DxGPT, abarcando desde estudios clínicos iniciales hasta el desarrollo de pipelines automatizados de alta fidelidad.

Resumen

Este informe describe la evolución metodológica y los resultados de la evaluación de cuatro modelos de lenguaje de OpenAI —**GPT-4o, o1, o3 y o3-pro**— sobre **450 casos pediátricos**. El análisis demostró que los evaluadores iniciales producían **realidades distorsionadas**: un pipeline basado en códigos (**PV2**: **ICD-10 + BERT**) **penalizaba la especificidad clínica** (la "**paradoja del especialista castigado**"), mientras que un juez-**LLM** (**PV3**) provocaba una **convergencia artificial de rendimientos** al premiar la **plausibilidad sobre la precisión**. Para superar esta limitación, se desarrolló **PV4**, un **pipeline jerárquico** que combina la **objetividad de los códigos** (**SNOMED, ICD-10**) con el **juicio semántico** (**BERT, LLM**), e introduce una métrica clave: **la posición en la lista diferencial** como indicador de **confianza clínica**. Esta innovación metodológica **rompió la saturación de la tarea** y reveló una **jerarquía de rendimiento clara**. El modelo **o3** emergió como el más confiable y estable, con la **mejor posición promedio** (1,47) y el **mayor número de aciertos en primera posición** (311), a pesar de que **o3-pro** obtuvo una **tasa de acierto bruta marginalmente superior** (96,4 %). Este último, sin embargo, lo logró a costa de una **peor priorización** (posición media de 1,60) y una **mayor dependencia del evaluador semántico**. El análisis de **prompts** confirmó que **exigir razonamiento explícito** —por ejemplo, **listar síntomas a favor y en contra**— es la **estrategia más eficaz**, superando tanto a los formatos más simples como a aquellos que sobrecargan al modelo solicitando campos adicionales como la *confianza*.

Índice

1. Planteamiento del problema de evaluación

La validación de sistemas de IA para el soporte al diagnóstico clínico representa uno de los desafíos metodológicos más complejos y estratégicamente relevantes en la medicina contemporánea. No se trata simplemente de verificar si una IA acierta en su diagnóstico, sino de evaluar la calidad, robustez y relevancia clínica de su razonamiento. Esta distinción es crucial: en el contexto real de atención médica, una hipótesis diagnóstica que suena plausible pero no es precisa puede comprometer tanto la seguridad del paciente como la confianza del profesional.

Como fundación dedicada al desarrollo ético y riguroso de herramientas diagnósticas basadas en IA, nuestro objetivo es establecer un marco de evaluación que supere las métricas superficiales y permita discernir con claridad el rendimiento diferencial entre modelos de distintas generaciones. Esta evaluación no debe limitarse a contar aciertos, sino que debe capturar las dimensiones más profundas del juicio clínico, incluyendo la priorización, la precisión terminológica y la coherencia semántica. Este informe documenta el proceso iterativo que hemos seguido para construir dicho marco cubriendo esos tres aspectos.

Partimos de una premisa de escepticismo científico: toda metodología de evaluación introduce sesgos y artefactos. Por tanto, nuestro trabajo no ha sido solo aplicar métricas, sino también interrogarlas, tensionarlas y rediseñarlas. Cada fase de nuestro pipeline nos ha permitido ver una parte distinta del problema —desde la rigidez de los sistemas de codificación hasta la excesiva generosidad de los evaluadores holísticos basados en **LLMs**— y, en ese recorrido, hemos aprendido tanto sobre los modelos como sobre nuestras propias herramientas de medición.

Más allá de evaluar un conjunto de modelos en un momento concreto, este documento busca sentar las bases para una evaluación clínicamente significativa y transparente.

2. Composición del entorno de pruebas

Para llevar a cabo una evaluación rigurosa, es indispensable contar con un entorno de pruebas que sea representativo y desafiante. El punto de partida es un conjunto agregado de 9.677 casos médicos provenientes de siete fuentes distintas, que incluyen recursos educativos (MedBullet, MedQA), bases de datos de enfermedades raras (RAMEDIS, Rare Synthetic), casos de urgencias (URGTorre) y otros de carácter especializado (Ukrainian, NEJM).

Si bien a lo largo del proyecto se probaron varias combinaciones de casos en función de los trade-offs de coste y tiempo en diversos pipelines, especialmente para pruebas rápidas o confirmaciones de hipótesis menores, para las evaluaciones definitivas de ranking se utilizó el dataset más equilibrado que se encontró. Entre todos los subconjuntos posibles derivados del corpus de 9.677 casos, este conjunto de **450 casos clínicos** fue seleccionado por un algoritmo de diversidad optimizada que maximiza la cobertura diagnóstica y minimiza la redundancia clínica, logrando el mejor equilibrio entre representatividad, complejidad y eficiencia computacional. Una visualización detallada de este proceso de extracción, transformación y carga (ETL) se encuentra en el **Anexo A**.

2.1 Diversidad del dataset final

El dataset de 450 casos fue seleccionado para asegurar una amplia cobertura de patologías, garantizando que la evaluación no estuviera sesgada hacia una especialidad concreta, como se muestra en la Figura 1.

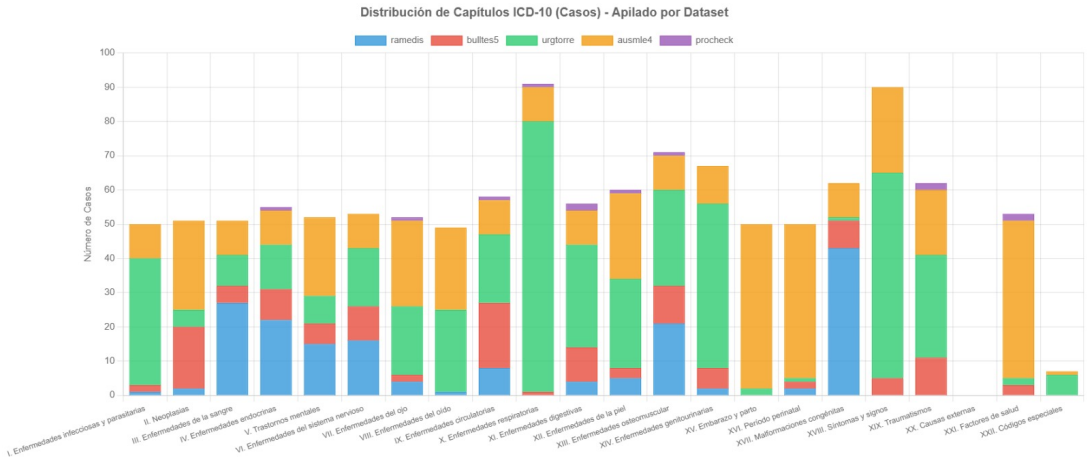

*Figura 1. Distribución de los 450 casos clínicos por capítulos de la codificación **ICD-10**.*

3. Evolución de los pipelines de evaluación

El desarrollo de nuestro framework de evaluación ha sido un proceso iterativo, donde cada pipeline representó una hipótesis sobre la mejor manera de medir el rendimiento. Esta evolución fue necesaria para confrontar y resolver el fenómeno de la saturación de la tarea.

Cuadro 1. Evolución de los pipelines de evaluación: ventajas, inconvenientes y lecciones aprendidas.

Pipeline	Ventajas	Inconvenientes	Lección
PV0 (BERT)	Simple, rápido y totalmente automatizado. Ideal para una primera criba de similitud semántica.	Ingenuo y ciego al contexto clínico. No distingue la sinonimia de la relevancia diagnóstica.	La similitud textual por sí sola es una métrica insuficiente y profundamente engañosa para el diagnóstico.
PV2 (ICD10+BERT)	Introduce objetividad y estructura usando códigos médicos estándar. **BERT** actúa como red de seguridad.	Excesiva rigidez que causa la "paradoja del especialista castigado", penalizando respuestas más específicas.	La codificación estricta es demasiado frágil y no captura la flexibilidad inherente al lenguaje clínico.
PV3 (Juez LLM)	Comprende el contexto, los matices y las relaciones clínicas complejas que los códigos ignoran.	Excesivamente generoso, causando la "saturación de la tarea". Iguala los rendimientos y oculta diferencias.	Un juicio semántico sin restricciones no mide la precisión, solo una plausibilidad que enmascara el rendimiento real.
PV4 (Híbrido)	Equilibra la objetividad de los códigos con la flexibilidad semántica, usando la posición para discriminar.	Mayor complejidad computacional y de implementación al requerir múltiples componentes y modelos.	La evaluación de alta fidelidad exige una cascada jerárquica que combine objetividad, semántica y prioridad.

- **PV0:** Fue el primer intento de automatización, utilizando exclusivamente un modelo **BERT** para medir la similitud semántica. Se aplicó principalmente a modelos de Hugging Face, entre ellos:
  - **sakura-solar-instruct-carbon-yuk**
  - **llama3-openbiollm-70b-zgh**
  - **jonsnowlabs-medellama-3-8b-v2-0-bfs**
  - **medgemma-27b-text-it**

  Su limitación era la incapacidad para entender el contexto clínico o las relaciones jerárquicas, reduciendo la evaluación a una simple comparación de proximidad textual.
- **PV1/PV2 (ICD10+BERT):** Representó un salto en sofisticación al introducir los códigos **ICD-10** como primer criterio de evaluación. Si la coincidencia de código fallaba, se utilizaba **BERT** como red de seguridad. Este método, aunque más estructurado, introdujo la "paradoja del especialista castigado", penalizando respuestas clínicamente superiores pero terminológicamente más específicas.
- **PV3 (Juez LLM):** Para corregir la rigidez anterior, este pipeline delegó la evaluación completa a un **Large Language Model** (**LLM**) que actuaba como "juez". Su capacidad de razonamiento contextual le permitió entender relaciones clínicas complejas, pero su excesiva "generosidad"llevó a la saturación de los resultados, como se discutirá en la siguiente sección.

La Figura 2 ilustra visualmente la diferencia en las distribuciones de resultados entre el enfoque de **PV0** (aplicado a modelos Hugging Face) y el de **PV3** (aplicado a modelos OpenAI), mostrando cómo diferentes metodologías producen realidades"de rendimiento muy distintas.

*Figura 2. Comparativa de resultados entre **PV0** (**BERT**, izquierda) y **PV3** (Juez **LLM**, derecha).*

Estas diferencias metodológicas no son triviales; de hecho, son la clave para entender el fenómeno de la saturación. En la siguiente sección se analizan en detalle las lecciones aprendidas de cada pipeline y cómo sus sesgos inherentes nos condujeron al diseño de un sistema de evaluación más robusto.

4. Análisis del fenómeno de la saturación de la tarea

Durante el desarrollo de nuestros pipelines, nos enfrentamos a un fenómeno tan persistente como problemático: la "saturación de la tarea". Con este término describimos la tendencia observada de que modelos de IA de diferentes generaciones y capacidades obtuvieran puntuaciones notablemente similares bajo ciertas métricas, creando una aparente meseta de rendimiento que contradecía el rápido avance teórico del campo. Este fenómeno no es una curiosidad, sino un obstáculo fundamental para la correcta valoración del progreso. Entenderlo es entender las trampas de la evaluación de la IA.

Este fenómeno se manifestó de formas distintas pero relacionadas en nuestros pipelines intermedios. Fue como observar un objeto distante a través de diferentes lentes: cada lente corregía una distorsión anterior, pero introducía una nueva, hasta que encontramos la combinación correcta que nos permitió ver con claridad.

4.1 La distorsión de la rigidez: PV2

Nuestro primer intento de automatización (**PV2**) buscaba la objetividad a través de la rigidez de los códigos médicos (**ICD-10**) y la sinonimia (**BERT**). El resultado fue un sistema que, si bien era objetivo, era ingenuo. Penalizaba la precisión clínica superior (la "paradoja del especialista castigado") y era ciego a cualquier relación que no fuera una equivalencia terminológica. El ranking que producía era claro, pero estaba basado en una visión del mundo clínico excesivamente simplificada. La Figura 3 muestra la distribución de puntuaciones de este sistema: un paisaje de picos discretos, reflejo de su naturaleza binaria, incapaz de capturar los matices.

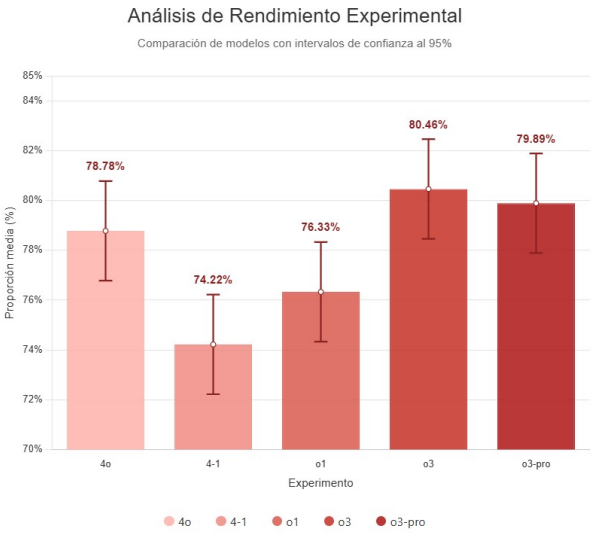

*Figura 3. Histograma de resultados para **PV2 (ICD10+BERT)**.*

4.2 La distorsión de la generosidad: PV3

Para corregir esta rigidez, **PV3** empleó un Juez **LLM**, esperando que su capacidad de razonamiento contextual proporcionara una evaluación más matizada. El resultado fue la manifestación más clara de la saturación. Como se observa en la Figura 4, las puntuaciones de todos los modelos se inflaron y se agruparon en una franja muy estrecha en el extremo superior de la escala. Un modelo de una generación anterior como **o1** obtuvo una puntuación casi idéntica a los de vanguardia como **o3**.

El motivo de este comportamiento es que el Juez **LLM**, al evaluar la "plausibilidad clínica", se había vuelto un evaluador excesivamente generoso. Entendía las relaciones causa-efecto, las

manifestaciones clínicas y las asociaciones diagnósticas, y premiaba todas estas conexiones. Al hacerlo, eliminó la distinción crucial entre una respuesta **correcta y precisa** y una respuesta meramente **relevante y plausible**. Esta generosidad actuó como un gran ecualizador, borrando las diferencias de rendimiento y creando una falsa meseta. La tarea para los modelos ya no era ser preciso, sino sonar lo suficientemente convincente para otro **LLM**.

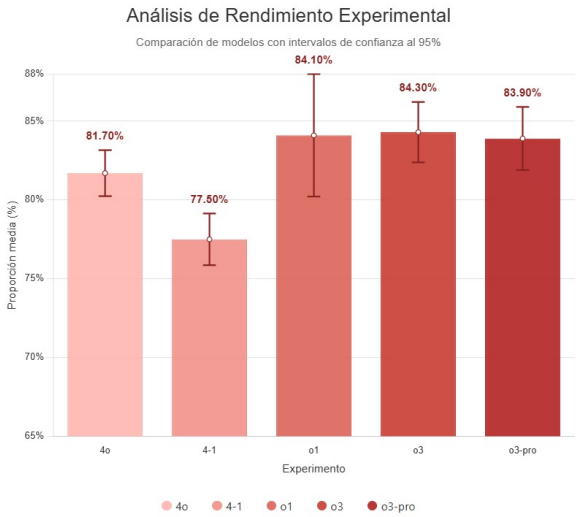

*Figura 4. Histograma de rendimiento para **PV3** (Juez **LLM**), mostrando la saturación de los resultados en la parte alta de la escala.*

La Figura 5 es la evidencia visual más contundente de este efecto. Muestra la transición de un paisaje de puntuaciones dispersas (**PV2**) a uno de convergencia casi total (**PV3**).

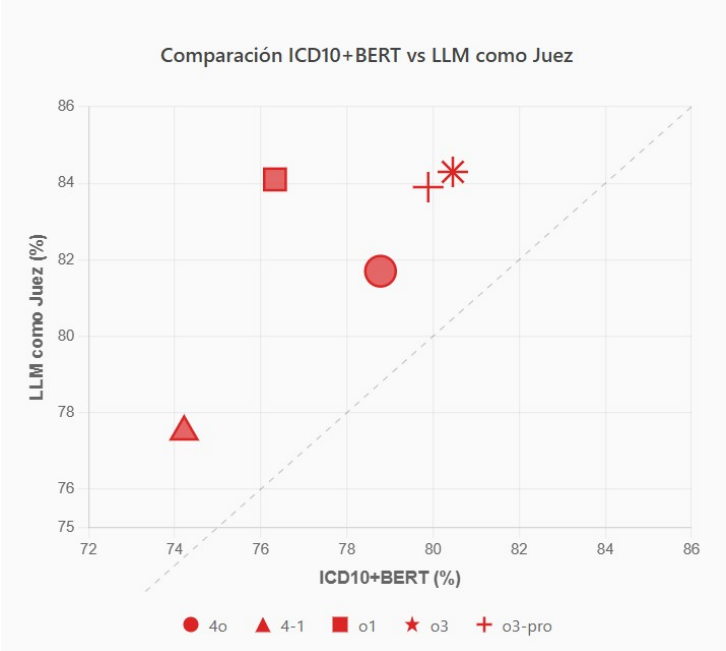

*Figura 5. Comparativa directa de resultados entre el método **ICD10+BERT** (**PV2**) y el Juez **LLM** (**PV3**).*

4.3 La naturaleza de la tarea y la naturaleza de los LLM

La raíz de este fenómeno yace en la interacción de tres factores: la **definición de la tarea**, la **naturaleza probabilística de los LLM** y el **criterio de evaluación**. Nuestra tarea, generar una lista de 5 diagnósticos diferenciales, es fundamentalmente una tarea de recuperación y ranking de información, no de creación desde cero. Los **LLM** modernos, desde **o1** hasta **o3**, poseen bases de conocimiento vastas. Con un prompt claro y restrictivo, todos son capaces de identificar un conjunto de diagnósticos plausibles.

La diferencia entre un modelo bueno y uno excelente no reside tanto en *si* puede encontrar el diagnóstico correcto, sino en *con qué prioridad y confianza* lo presenta. **PV3**, al ser tan generoso, fallaba en medir esta dimensión de confianza. **PV4** fue diseñado específicamente para resolver este problema, reintroduciendo la rigidez de forma controlada y haciendo de la **precisión posicional** un criterio de desempate clave. Al hacerlo, finalmente logramos romper la ilusión de la convergencia y medir lo que realmente importa: no solo el acierto, sino la calidad y confianza de ese acierto.

5. Diseño y lógica de PV4

PV4 es el resultado de este proceso de aprendizaje. No es un método único, sino un sistema jerárquico que sintetiza las lecciones de sus predecesores, buscando un equilibrio entre la objetividad de los códigos y la inteligencia del análisis semántico.

5.1 Análisis preliminar y justificación del diseño

El diseño de **PV4** se fundamentó en un análisis detallado de los resultados intermedios producidos por el set de evaluación en pipelines anteriores. Este análisis mostró hechos clave que condicionaron su arquitectura:

1. **SNOMED CT es la columna vertebral semántica.** Aunque la codificación es multiontológica, **SNOMED CT** es el sistema más prevalente, cubriendo el 76 % de los diagnósticos de referencia (**GDX**) y el 87 % de los diagnósticos diferenciales propuestos (**DDX**). Su rol es central para establecer coincidencias directas y fiables (**Nivel 1**).

2. **La granularidad de ICD-10 requiere flexibilidad.** Se construyó una matriz de transiciones para cuantificar la proximidad en la jerarquía **ICD-10**, mostrando que más del 92 % de las coincidencias clínicamente útiles se ubicaban a una arista de distancia (padre $\leftrightarrow$ hijo). En consecuencia, **PV4** considera como acierto válido cualquier hijo o padre inmediato en la jerarquía (Figura 6), superando la rigidez de un match exacto.

3. **La inevitable brecha semántica del 43.3 %.** En 195 de los 450 casos (43.3 %), no existe ningún código compartido (**SNOMED**, **ICD-10** u **OMIM**) entre el diagnóstico de referencia y las cinco propuestas del modelo. Esta "brecha semántica"hace indispensable contar con métodos de validación que no dependan de ontologías, justificando el uso de **BERT** y **LLMs** (**Nivel 2**) para rescatar aciertos semánticamente cercanos pero no codificados.

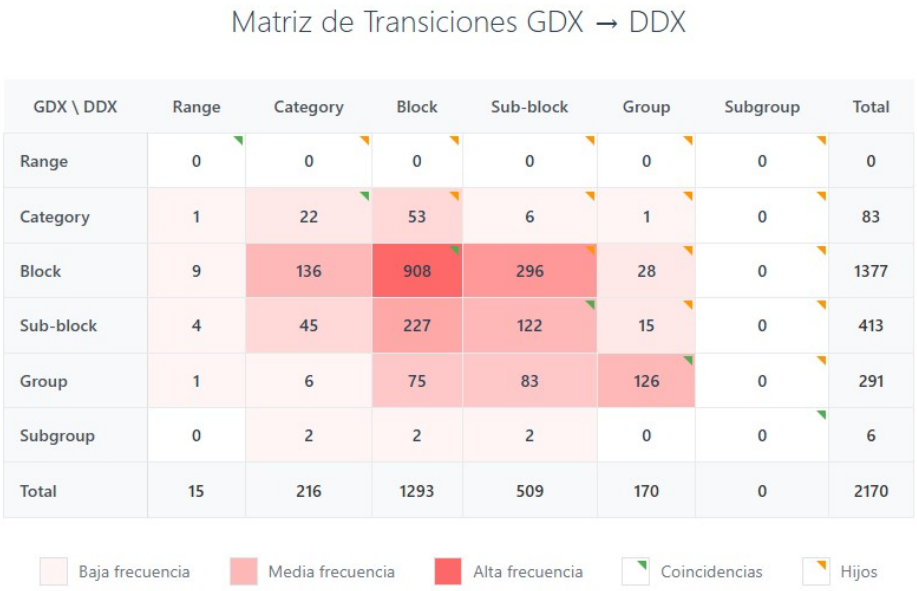

*Figura 6. Análisis de las relaciones jerárquicas **ICD-10** en una muestra de los casos, justificando un scoring más sofisticado para **PV4**.*

5.2 Lógica operativa y flujo de decisión

La lógica de **PV4** es una cascada determinista que evalúa cada una de las 5 propuestas diagnósticas (**DDX**) de un modelo en orden, desde la posición 1 hasta la 5. El proceso se detiene en cuanto encuentra la primera coincidencia válida con el diagnóstico de referencia (**GDX**), y la puntuación del caso se determina por la posición de ese primer acierto. Esto prioriza la confianza clínica del modelo.

Para cada **DDX** individual, el pipeline (Figura 7) sigue una jerarquía de validación que va de lo más objetivo a lo más interpretativo:

Nivel 1: Verificación por códigos (máxima objetividad): Se busca una coincidencia de código **SNOMED CT** (match exacto). Si no se encuentra, se verifica una coincidencia **ICD-10** (exacta, padre, hijo o hermano). Si cualquiera de estos métodos tiene éxito, el **DDX** se considera un acierto, el caso se da por resuelto y se registra su posición.

Nivel 2: Juicio semántico (red de seguridad): Solo si los códigos fallan, se recurre al análisis semántico. Primero, se evalúa si la similitud del coseno de **BERT** supera un umbral de alta confianza ($> 0{,}925$). Si es así, se considera un acierto.

Nivel 3: Desempate semántico (IA vs. IA): Si la similitud **BERT** se encuentra en una zona de incertidumbre ($0{,}89 < \cos\theta < 0{,}925$), se activa un juicio competitivo. Se compara el veredicto de **BERT** con el de un **Juez LLM**. Si ambos proponen que el **DDX** es un acierto, se elige aquel que aparezca en la posición más alta de la lista diferencial original, premiando la confianza del modelo. Si solo uno lo aprueba, se acepta ese veredicto. Si ninguno de los dos métodos considera que hay un acierto, el **DDX** se descarta.

Este proceso se repite para cada **DDX** en la lista hasta que se encuentra un acierto o se agotan las 5 propuestas.

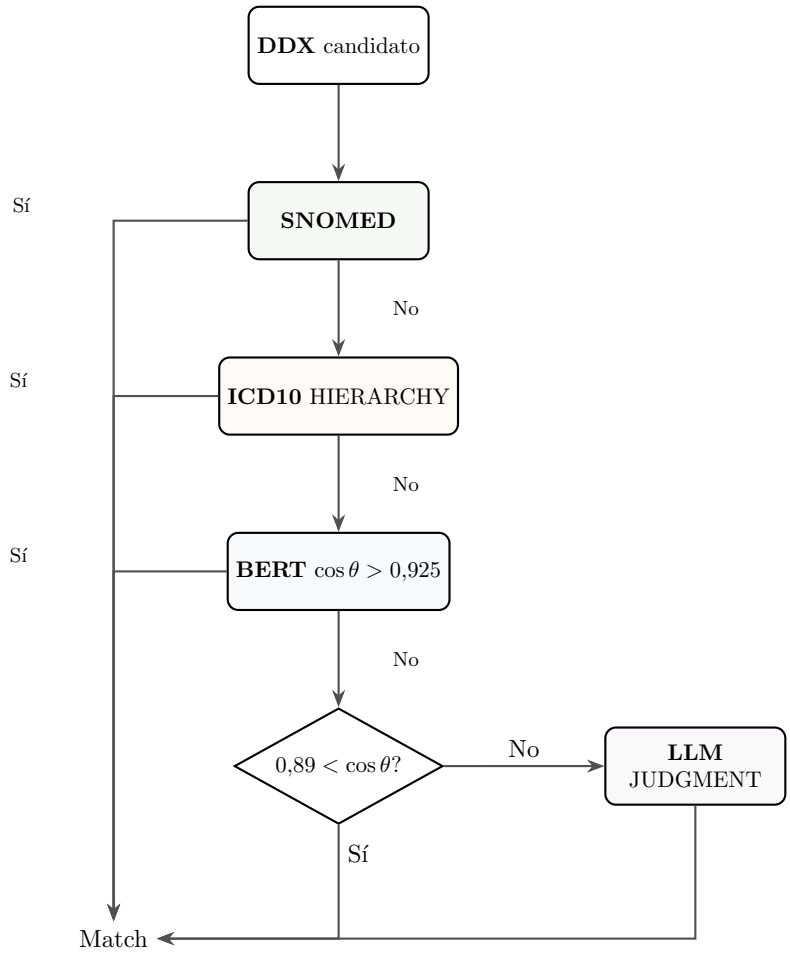

*Figura 7. Diagrama de flujo del proceso de evaluación de **PV4** para un único **DDX**, mostrando su cascada jerárquica.*

6. Resultados y análisis de PV4

La aplicación de este marco de alta fidelidad mostró una jerarquía de rendimiento clara y robusta, cuya sensibilidad nos permitió no solo comparar modelos, sino también el impacto de la ingeniería de prompts.

6.1 Análisis preliminar de la distribución de métodos (PV4)

Antes de analizar el rendimiento de los modelos, es crucial entender el comportamiento del propio pipeline de evaluación. Un análisis preliminar sobre un subconjunto de casos (Figura 8) muestra cómo se distribuyen las resoluciones entre los distintos métodos de **PV4**. Se observa que la combinación de **SNOMED** (coincidencia exacta) y el **Juez LLM** (juicio semántico) explican la gran mayoría de los aciertos, con un **76.16 %** del total. Esto confirma que la objetividad de los códigos es la base, pero se necesita un evaluador contextual para los casos semánticamente complejos. Por su parte, **ICD-10** (12.16 %) y **BERT** (11.68 %) actúan como mecanismos secundarios pero importantes para capturar relaciones jerárquicas y sinonimias no cubiertas por el primer nivel.

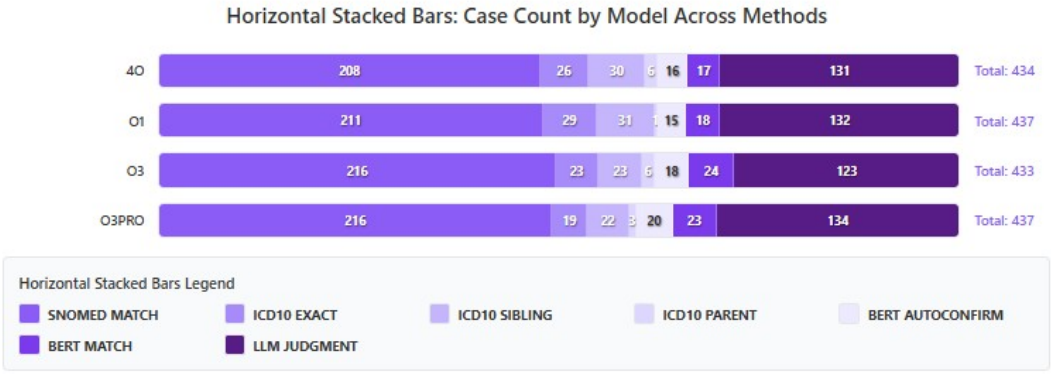

*Figura 8. Estudio preliminar del desglose de los métodos de resolución del pipeline **PV4**. Muestra la contribución de cada componente a la validación de aciertos.*

6.2 Optimización de prompts y su impacto en el rendimiento

El benchmark no solo permite comparar modelos, sino también la eficacia de diferentes prompts. La sensibilidad de **PV4** nos ha permitido cuantificar cómo la estructura del prompt influye en la calidad de la respuesta, mostrando patrones clave para la optimización. Las Figuras 9 y 10 ilustran la variabilidad del rendimiento en función del prompt utilizado para un mismo modelo y viceversa. En el **Anexo B** se adjuntan los prompts que obtuvieron los mejores resultados.

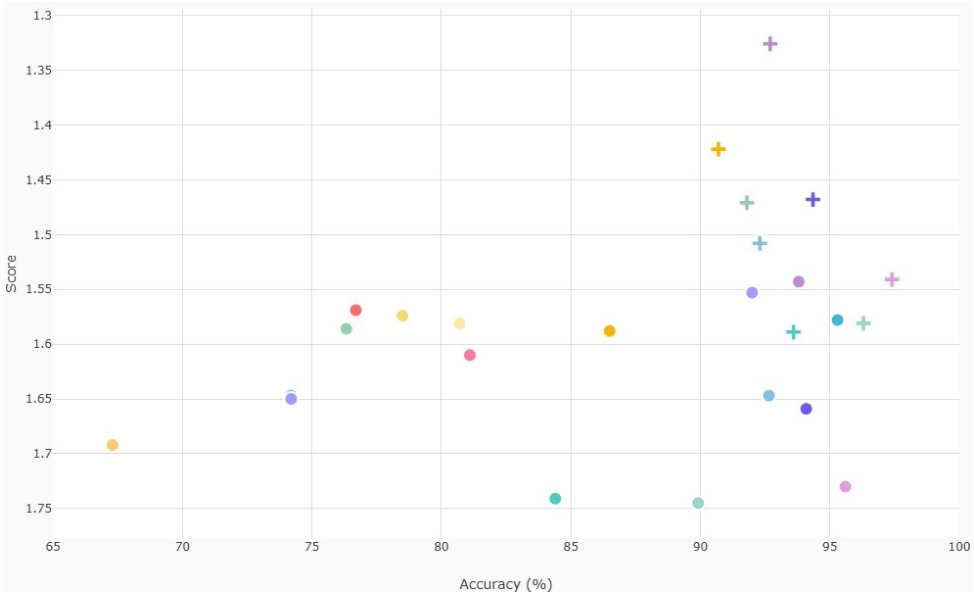

*Figura 9. Comparativa de rendimiento entre diferentes variantes de prompts (las cruces indican ejecuciones con **o3** y los círculos con **4o**).*

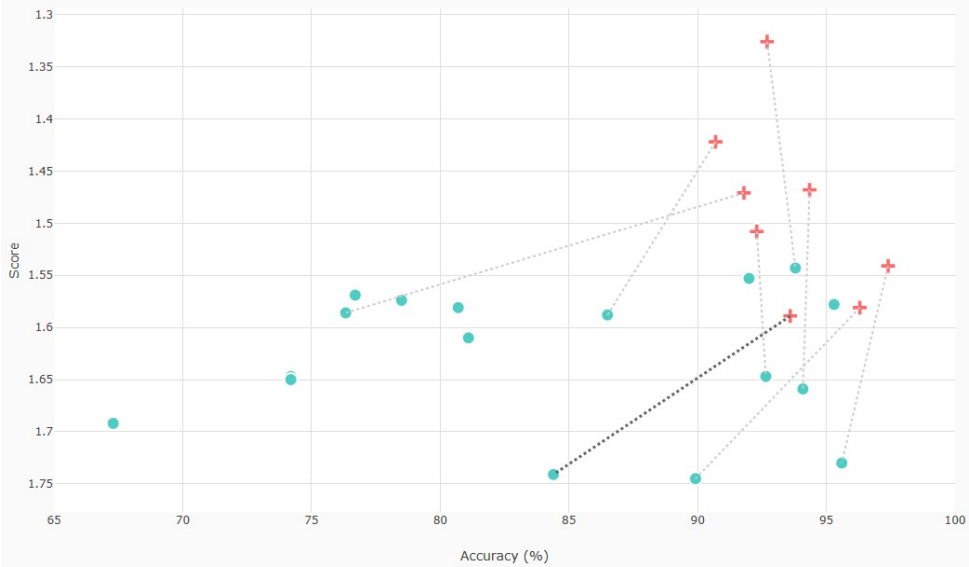

*Figura 10. Comparativa de rendimiento entre diferentes modelos bajo un mismo prompt (en rojo se muestran los prompts ejecutados con el modelo **o3**, en azul los de **4o**; las líneas discontinuas conectan los prompts que se probaron en ambos modelos para visualizar la diferencia de rendimiento).*

Un análisis más detallado sobre el tipo de salida solicitado en el prompt arroja conclusiones significativas:

Cuadro 2. Impacto del formato de salida del prompt en el rendimiento.

Tipo de Salida Solicitada	**Nº Prompts**	**PPos Media**	**% Acierto Medio**
Lista sencilla de strings	9	1.630	88.36 %
Objeto con *rationale* + *confidence*	6	1.604	77.42 %
Objeto con síntomas (in/out)	**6**	**1.506**	**91.44 %**
Objeto con síntomas + envoltura XML	3	1.560	93.48 %

Además del formato de salida, la longitud del prompt es otro factor determinante. El análisis sobre 24 variantes (7 para **o3** y 17 para **4o**) muestra que la longitud y la complejidad del output interactúan de forma decisiva. En promedio, añadir 50 palabras a un prompt mejora la tasa de acierto en unos 3 puntos porcentuales. Los prompts cortos (¡80 palabras) rinden un 76.1 %, los de longitud media (80-160 palabras) un 86.3 %, y los largos (¿160 palabras) un 92.9 %, acumulando una ganancia de hasta 16.8 puntos por longitud. Sin embargo, esta mejora puede verse anulada por la complejidad de la salida. Al controlar la longitud (120-140 palabras), exigir campos adicionales como `rationale` o `confidence` provoca una caída de la precisión de hasta 22 puntos porcentuales frente a formatos de salida más simples (p.ej., de 95-97 % a 67-76 %). Esto indica que la penalización por un formato de salida complejo puede pesar más que el beneficio obtenido por un prompt más largo y detallado.

Finalmente, se analizó el efecto de incluir una cláusula "failsafe.[en] el prompt (p. ej., *"FAIL-SAFE: If the input is not a clinical case, output []"*). De 24 plantillas, 10 la incluían. La tasa de acierto media para estos prompts fue del 83.1 %, inferior a la media global (86.9 %), sugiriendo una ligera penalización. Esto indica que la validación del tipo de input debería gestionarse mediante un algoritmo previo, más simple y rápido, en lugar de sobrecargar el prompt principal.

6.3 Ranking final de modelos y análisis comparativo

Tras la fase de optimización de prompts, se procedió a la evaluación final de los modelos. Los parámetros presentados en la Tabla 3 son el resultado de un proceso de agregación estadística. Se obtuvieron promediando los resultados de múltiples ejecuciones, filtrando por modelo para obtener estimadores robustos y representativos de su rendimiento general, mitigando así la variabilidad inherente a una única configuración.

*Cuadro 3. Ranking y métricas clave de rendimiento por modelo (**PV4**).*

Métrica	**o3**	**o1**	**o3-pro**	**4o**
Tasa de Acierto (%)	93.65 %	91.44 %	**96.40 %**	94.31 %
Posición Promedio	**1.474**	1.585	1.597	1.629
Aciertos en Posición 1 (P1)	**311**	305	299	299
Aciertos en Posición 5 (P5)	**9**	17	24	20

Los resultados muestran una interesante dicotomía: el modelo **o3-pro** alcanza la mayor tasa de acierto global (96.4 %), pero es el modelo **o3** el que demuestra una confianza clínica superior, con la mejor posición promedio (1.474) y el mayor número de aciertos en primera posición (311). La Figura 11 visualiza esta ventaja posicional de **o3**, que relega el acierto a la última posición con mucha menos frecuencia que sus competidores.

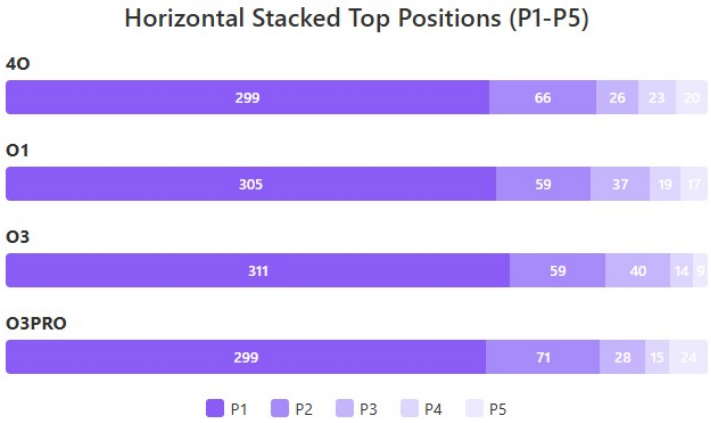

Figura 11. Distribución de los aciertos en las Top 5 posiciones por modelo.

6.3.1 Análisis de estabilidad del rendimiento

Para profundizar en la robustez de los modelos, se analizó la variabilidad de su rendimiento a través de diferentes prompts. La Tabla 4 compara directamente **o3** y **4o**.

*Cuadro 4. Análisis de estabilidad del rendimiento entre **o3** y **4o**.*

Modelo	**Nº Prompts (n)**	**PPos Media ($\mu \pm \sigma$)**	**% Acierto ($\mu \pm \sigma$)**
o3	7	1.474 ± 0.083	93.65 % ± 2.46
4o	17	1.629 ± 0.066	84.31 % ± 8.91

El análisis muestra que el rendimiento de **o3** no solo es superior en promedio, sino también mucho más estable. Su coeficiente de variación (CV) para la tasa de acierto es de **0.026**, mientras que el de **4o** es de **0.106**. Esto significa que **la variabilidad de 4o cuadruplica la de o3**,

lo que lo hace menos predecible y consistente en entornos de producción donde la fiabilidad es crítica.

6.3.2 Desglose del método de resolución por modelo

PV4 nos permite ver la utilidad validadora de cada modelo. El análisis (Figura 12) muestra que **o3** basa su éxito en una mayor disciplina taxonómica. Por el contrario, **o3-pro** necesita un mayor apoyo semántico y presenta una cola más larga de aciertos en posiciones tardías (P4-P5), lo que sugiere que obtiene cobertura a costa de la precisión en el ranking.

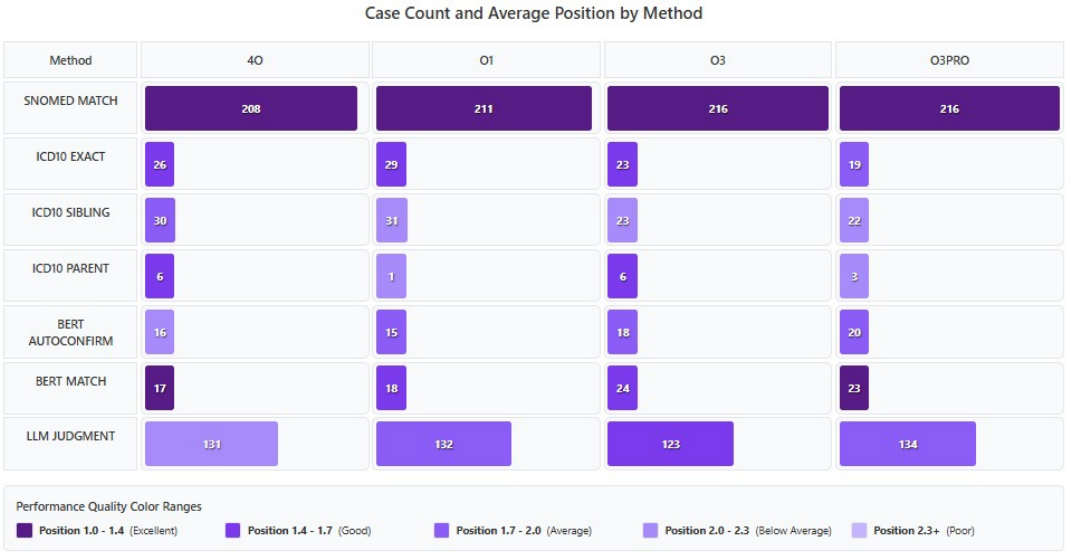

Figura 12. Número de casos resueltos y posición promedio por cada método y modelo. Los tonos más oscuros representan mejor puntuación.

7. Discusión

La metodología incremental documentada en este informe nos ha llevado a una conclusión compleja y matizada. Si bien **PV4** representa nuestro esfuerzo más sofisticado por medir el rendimiento de la IA diagnóstica, sus resultados, lejos de ofrecer una respuesta definitiva, nos enfrentan a una manifestación aún más sutil del fenómeno de la saturación de la tarea.

7.1 Análisis crítico de los resultados de PV4

A primera vista, **PV4** establece una jerarquía clara. Sin embargo, una mirada más crítica muestra un panorama complejo. El modelo **o3-pro** alcanza la máxima tasa de acierto, pero a costa de una peor priorización y mayor dependencia de la evaluación semántica. Por otro lado, **o3** emerge como el modelo más confiable y estable, con la mejor posición promedio y la menor variabilidad entre prompts. Esta tensión entre **cobertura (o3-pro) y confianza (o3)** es un hallazgo clave.

La diferencia en la tasa de acierto entre los modelos de alto rendimiento es relativamente estrecha, lo que podría interpretarse como una señal de saturación. Planteamos la siguiente hipótesis: el dataset de 450 casos, a pesar de su diversidad, podría estar concentrado en un espectro de dificultad que se resuelve mediante un reconocimiento de patrones.[a]ltamente sofisticado, más que un razonamiento de primeros principios". Si la mayoría de los casos se resuelven identificando constelaciones de síntomas que los modelos ya han internalizado masivamente, es

lógico que converjan en rendimiento. La tarea no estaría midiendo su capacidad de "pensar", sino la exhaustividad de su "memoria" de patrones clínicos.

7.2 Significancia estadística e incertidumbre

La consistencia de los resultados a través de 450 casos diversos y múltiples variantes de prompts aporta una base empírica razonable para las conclusiones. La ventaja posicional y de estabilidad de **o3** sobre **4o** es estadísticamente notable, como demuestra la diferencia de cuatro veces en su coeficiente de variación. De manera similar, la dicotomía entre la tasa de acierto de **o3-pro** y la calidad del ranking de **o3** es un patrón robusto. Por tanto, consideramos que la jerarquía y los perfiles de rendimiento observados son señales significativas dentro del marco evaluado, más que artefactos del azar.

En este contexto, **PV4** ha demostrado tener un poder discriminativo suficiente no solo para rankear modelos, sino para caracterizar sus perfiles de rendimiento (p. ej., estabilidad, confianza vs. cobertura) y para guiar la ingeniería de prompts.

8. Conclusiones

El proceso iterativo de diseño y validación de pipelines nos ha proporcionado una comprensión profunda no solo del rendimiento de los modelos, sino de la naturaleza misma de la evaluación de IA en un dominio tan complejo como el diagnóstico clínico. Las conclusiones se pueden estructurar en tres áreas clave: el rendimiento de los modelos, las lecciones sobre la metodología y las directrices para la ingeniería de prompts.

Veredicto del rendimiento de los modelos

- **o3** destaca por su fiabilidad: mejor posición promedio (1,47) y mayor número de aciertos en P1.
- **o3-pro** logra la máxima cobertura (96,4 % de aciertos) a costa de una peor priorización en el ranking.
- La estabilidad de **o3** contrasta con la alta variabilidad de **4o**, haciéndolo más predecible para producción.
- La estrecha diferencia de rendimiento global sugiere que la tarea se acerca a un límite de discriminación.

Lecciones sobre la metodología de evaluación

- **PV2** (**ICD10+BERT**) demostró que la rigidez de códigos castiga injustamente la especificidad clínica superior.
- **PV3** (Juez **LLM**) enseñó que la generosidad semántica crea una falsa convergencia de rendimientos (saturación).
- **PV4** funciona al combinar objetividad (códigos) y semántica (**LLM**), usando la posición como desempate clave.
- El evaluador no es un observador pasivo; define activamente la métrica de éxito de la tarea.

Claves para la ingeniería de prompts

- Exigir un razonamiento estructurado (síntomas a favor/en contra) mejora la precisión del diagnóstico principal.
- Sobrecargar el prompt con tareas secundarias (`confidence`, `rationale`) degrada el rendimiento de la tarea primaria.
- Las cláusulas de seguridad (`FAILSAFE`) penalizan el rendimiento; la validación debe ser un paso previo.

A. Composición detallada del dataset de evaluación

El dataset final de 450 casos utilizado para la evaluación comparativa de los modelos se construyó a partir de un universo de 9.677 casos clínicos agregados de siete fuentes diferentes. La selección no fue aleatoria, sino que se basó en un proceso de Extracción, Transformación y Carga (ETL) diseñado para garantizar la diversidad y representatividad del conjunto de pruebas. El siguiente diagrama de Sankey (Figura 13) visualiza este flujo, mostrando cómo los casos de cada fuente original contribuyen al dataset final. Este método de muestreo estratificado es fundamental para asegurar que los resultados de la evaluación sean robustos y generalizables.

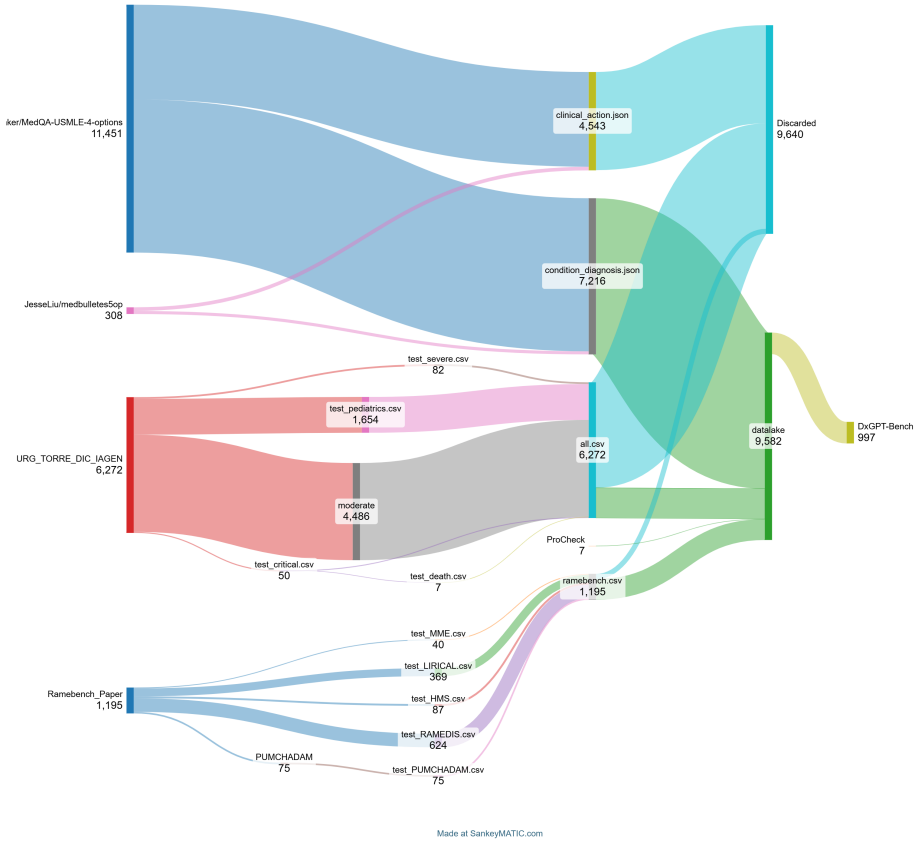

Figura 13. Diagrama de Sankey que visualiza el proceso de ETL para la composición del dataset de evaluación de 450 casos a partir de las fuentes originales.

B. Prompts con mayor rendimiento

A continuación se presentan los prompts que obtuvieron los mejores resultados para los modelos **o3** y **4o**, junto con sus puntuaciones de rendimiento (posición promedio y tasa de acierto).

B.1 Mejores prompts para o3 (TOP4)

diagnosis_description_symptoms_classic_v2

Puntuación: 1.326 - 92.7 %

```
You are a diagnostic assistant. Given the patient case below, generate N possible diagnoses.

- Give a brief description of the disease
- List symptoms the patient has that match the disease
- List patient symptoms that are not typical for the disease

Output format:
Return a JSON array of N objects, each with the following keys:
- "diagnosis": disease name
- "description": brief summary of the disease
- "symptoms_in_common": list of matching symptoms
- "symptoms_not_in_common": list of patient symptoms not typical of that disease

Output only valid JSON (no extra text, no XML, no formatting wrappers).

Example:
‘‘‘json
[
  {{
    "diagnosis": "Disease A",
    "description": "Short explanation.",
    "symptoms_in_common": ["sx1", "sx2"],
    "symptoms_not_in_common": ["sx3", "sx4"]
  }},
  ...
]

PATIENT DESCRIPTION:
{case_description}
```

diagnosis_description_symptoms_classic_v4

Puntuación: 1.422 - 90.7 %

```
You are a diagnostic assistant. Given the patient case below, generate N possible diagnoses.

- Give a brief description of the disease
- List symptoms the patient has that match the disease
- List patient symptoms that are not typical for the disease

Output format:
```

```
Return a JSON array of N objects, each with the following keys:
- "diagnosis": disease name
- "description": brief summary of the disease
- "symptoms_in_common": list of matching symptoms
- "symptoms_not_in_common": list of patient symptoms not typical of that disease

Output only valid JSON (no extra text, no XML, no formatting wrappers).

Example:
‘‘‘json
[
  {{
    "diagnosis": "Disease A",
    "description": "Short explanation.",
    "symptoms_in_common": ["sx1", "sx2"],
    "symptoms_not_in_common": ["sx3", "sx4"]
  }},
  ...
]

PATIENT DESCRIPTION:
{case_description}
```

juanjo_v1 (original)

Puntuación: 1.468 - 94.35 %

```
Behave like a hypothetical doctor tasked with providing N hypothesis diagnosis for a
patient based on their description. Your goal is to generate a list of N potential
diseases, each with a short description, and indicate which symptoms the patient has
in common with he proposed disease and which symptoms the patient does not have
in common.

Carefully analyze the patient description and consider various potential diseases
that could match the symptoms described. For each potential disease:
1. Provide a brief description of the disease
2. List the symptoms that the patient has in common with the disease
3. List the symptoms that the patient has that are not in common with the disease

Present your findings in a JSON format within XML tags. The JSON should contain
the following keys for each of the N potential disease:
- "diagnosis": The name of the potential disease
- "description": A brief description of the disease
- "symptoms_in_common": An array of symptoms the patient has that match the disease
- "symptoms_not_in_common": An array of symptoms the patient has that are not
                            in common with the disease

Here’s an example of how your output should be structured:

<diagnosis_output>
[
{{
```

```
    "diagnosis": "some disease 1",
    "description": "some description",
    "symptoms_in_common": ["symptom1", "symptom2", "symptomN"],
    "symptoms_not_in_common": ["symptom1", "symptom2", "symptomN"]
}},
...
{{
    "diagnosis": "some disease n",
    "description": "some description",
    "symptoms_in_common": ["symptom1", "symptom2", "symptomN"],
    "symptoms_not_in_common": ["symptom1", "symptom2", "symptomN"]
}}
]
</diagnosis_output>

Present your final output within <diagnosis_output> tags as shown in the example above.

Here is the patient description:
<patient_description>
{case_description}
</patient_description>
```

claude_sonnet_4

Puntuación: 1.471 - 91.8 %

```
Generate 5 differential diagnoses from the clinical case below.

ANALYSIS: Consider common through rare conditions, metabolic/structural causes,
demographics, timeline, and clinical epidemiology. Prioritize treatable conditions.

OUTPUT: JSON array of objects:
[{{"dx":"Disease","rationale":"Brief reason","confidence":"High/Medium/Low"}}]

CASE: {case_description}
```

B.2 Mejores prompts para 4o (TOP4)

diagnosis_description_symptoms_classic_v2

*Este prompt, ya presentado en la sección anterior, también obtiene el mejor rendimiento para el modelo **4o**.*

juanjo_v1 (sin `description`)

```
Behave like a hypothetical doctor tasked with providing N hypothesis diagnosis for a
patient based on their description. Your goal is to generate a list of N potential
diseases and indicate which symptoms the patient has in common with the proposed
disease and which  symptoms the patient does not have in common.

Carefully analyze the patient description and consider various potential diseases
that could match the symptoms described. For each potential disease:
1. List the symptoms that the patient has in common with the disease
```

```
2. List the symptoms that the patient has that are not in common with the disease

Present your findings in a JSON format within XML tags. The JSON should contain
the following keys for each of the N potential disease:
- "diagnosis": The name of the potential disease
- "symptoms_in_common": An array of symptoms the patient has that match the disease
- "symptoms_not_in_common": An array of symptoms the patient has that are not
                            in common with the disease

Here's an example of how your output should be structured:

<diagnosis_output>
[
{{
    "diagnosis": "some disease 1",
    "symptoms_in_common": ["symptom1", "symptom2", "symptomN"],
    "symptoms_not_in_common": ["symptom1", "symptom2", "symptomN"]
}},
...
{{
    "diagnosis": "some disease n",
    "symptoms_in_common": ["symptom1", "symptom2", "symptomN"],
    "symptoms_not_in_common": ["symptom1", "symptom2", "symptomN"]
}}
]
</diagnosis_output>

Present your final output within <diagnosis_output> tags as shown in the example above.

Here is the patient description:
<patient_description>
{case_description}
</patient_description>
```

4o_4

```
TASK: Given the patient case, return 5 most likely diagnoses (ranked).

RULES:
• Include only diseases that plausibly explain most symptoms.
• Use standard medical terms (precise, specific).
• Always include rare/treatable/metabolic if fitting.
• Prefer unifying Dx > partials.
• Penalize weak matches or noise.
• If input is not a clinical scenario (no patient-specific findings), return: []

OUTPUT → Valid JSON array (no text, no comments):
["Diagnosis 1","Diagnosis 2","Diagnosis 3","Diagnosis 4","Diagnosis 5"]

PATIENT:
{case_description}
```

claude_opus_1

```
You are a world-class diagnostic clinician with expertise across all medical specialties.
Generate exactly 5 differential diagnoses for the patient case below.

CRITICAL INSTRUCTIONS:
- Rank diagnoses by probability given ALL clinical features (most to least likely)
- Consider the COMPLETE clinical picture: demographics, timeline, severity, progression
patterns
- Include ALL plausible conditions: common, rare, genetic, metabolic, structural,
infectious, autoimmune
- Weight classic presentations heavily but don't ignore atypical variants
- Never dismiss treatable conditions regardless of rarity
- Apply Occam's razor AND Hickam's dictum appropriately

OUTPUT FORMAT: Return ONLY a JSON array of disease names as strings, nothing else.
Example: ["Disease A","Disease B","Disease C","Disease D","Disease E"]

CLINICAL CASE:
{case_description}
```