Formalización métrica del juicio clínico en sistemas LLM de diagnóstico diferencial

Un marco de evaluación híbrido ontológico-semántico para superar la convergencia artificial de rendimiento

Autores

Yago Mendoza
Javier Logroño
Carlos Bermejo

Foundation 29
29 de julio de 2025

Versión 2.1
White Paper Técnico

Resumen

Este informe detalla la evolución metodológica y los hallazgos en la evaluación de cuatro modelos de lenguaje de OpenAI —**GPT-4o, o1, o3 y o3-pro**— en un benchmark de **450 casos clínicos pediátricos**. El estudio partió de la premisa de que los enfoques de evaluación simples generan resultados engañosos. Se demostró que un pipeline basado en códigos (**PV2**: **ICD-10** + **BERT**) penalizaba la especificidad clínica (la "**paradoja del especialista castigado**"), mientras que un juez-**LLM** (**PV3**) enmascaraba las diferencias de rendimiento al premiar la plausibilidad sobre la precisión, provocando una **saturación de la tarea**.

Para superar estas limitaciones, se desarrolló **PV4**, un **pipeline jerárquico** que integra la objetividad de las ontologías médicas (**SNOMED, ICD-10**) con el juicio semántico contextual (**BERT, LLM**). La innovación clave de **PV4** es el uso de la **posición del diagnóstico correcto en la lista diferencial** como métrica principal, evaluando así la capacidad de **priorización clínica** del modelo.

Esta nueva metodología logró romper la aparente convergencia de los modelos, revelando una **clara jerarquía de rendimiento**. El modelo **o3** emergió como el más fiable y con la mejor capacidad de priorización (posición promedio de **1,47** y **311** aciertos en primera posición). En contraste, **o3-pro**, aunque alcanzó la tasa de acierto bruta marginalmente más alta (**96,4 %**), lo hizo a costa de una priorización inferior (posición promedio de **1,60**) y una mayor dependencia de los evaluadores semánticos. Adicionalmente, el análisis de **prompts** confirmó que **exigir un razonamiento explícito** —como listar síntomas a favor y en contra— es la estrategia más eficaz para optimizar la calidad del diagnóstico, superando tanto a los formatos simples como a los sobrecargados.

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

La construcción del dataset final de **450 casos clínicos** se realizó mediante un proceso de **curación algorítmica por fases**, un enfoque diseñado para superar los sesgos de representatividad y la redundancia inherentes al muestreo aleatorio. El objetivo no era crear una muestra estadísticamente representativa de la prevalencia de enfermedades, sino construir un *benchmark* de alta exigencia, optimizado para discriminar el rendimiento de los modelos. La lógica operativa fue la siguiente:

Fase 1: Establecimiento de un núcleo estratificado: El proceso comienza garantizando la inclusión de un conjunto de casos de fuentes consideradas estratégicas. Se preasigna una

cuota mínima o total para datasets de alta relevancia, como los de enfermedades raras (RAMEDIS) o complejidad diagnóstica (NEJM). Esta estratificación inicial asegura la presencia de escenarios de baja prevalencia pero alto valor informativo, que un muestreo aleatorio simple probablemente omitiría.

Fase 2: Llenado por optimización de la diversidad marginal: Una vez asegurado el núcleo, el algoritmo completa el dataset de forma iterativa. En cada paso, en lugar de una selección aleatoria, se implementa una **heurística de puntuación ponderada** para evaluar a todos los candidatos restantes. Se selecciona aquel caso que ofrezca el **máximo valor marginal**, es decir, el que más contribuya a la diversidad y complejidad del conjunto ya formado. Dicha heurística pondera simultáneamente varios ejes de calidad:

- **Maximización de la cobertura taxonómica:** La máxima ponderación se asigna a los casos que introducen un **capítulo ICD-10** no representado. Este criterio fuerza al dataset a cubrir un espectro amplio de dominios médicos.
- **Minimización de la redundancia nosológica:** Se bonifica la inclusión de diagnósticos con códigos ICD-10 específicos aún no presentes, evitando la saturación del *benchmark* con patologías comunes.
- **Balance de fuentes de origen:** Se aplica un factor de corrección que favorece a los casos de fuentes subrepresentadas, mitigando el riesgo de que el dataset final esté dominado por las características de una única colección masiva.
- **Priorización de la complejidad:** Se premian explícitamente los casos con mayores índices de complejidad y severidad[1] para asegurar que el *benchmark* contenga una proporción significativa de escenarios diagnósticos no triviales.

El resultado es un conjunto de pruebas que, a diferencia de una muestra aleatoria, está intencionadamente sesgado hacia la diversidad y la dificultad. Este diseño lo convierte en un *benchmark* con mayor poder discriminativo, capaz de tensionar y diferenciar de manera más fiable las capacidades de razonamiento clínico de los distintos modelos. Una visualización detallada de este proceso de extracción, transformación y carga (ETL) se encuentra en el **Anexo A**.

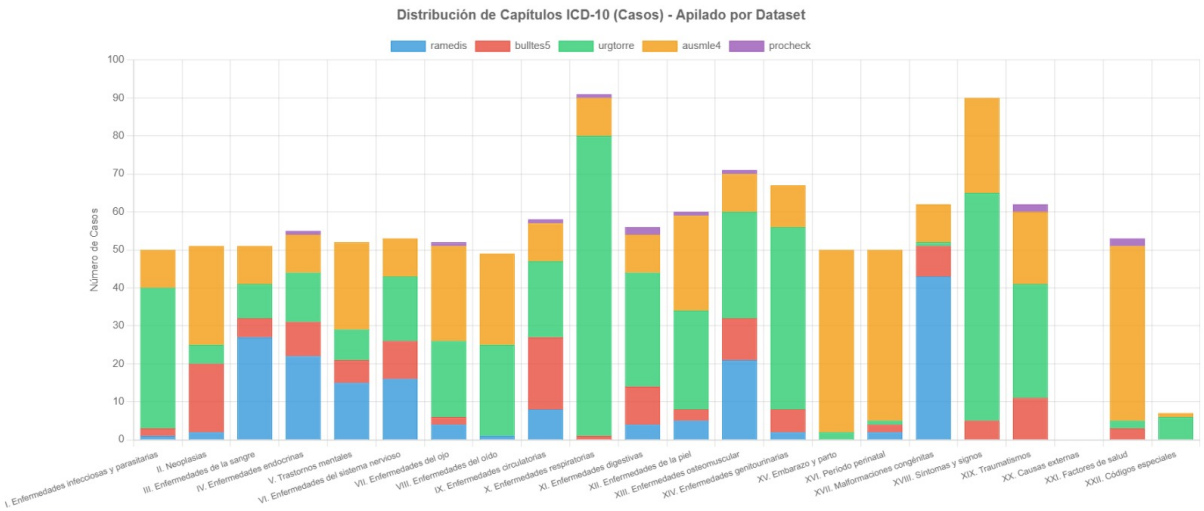

*Figura 1. Distribución de los 450 casos clínicos por capítulos de la codificación **ICD-10**. El eje X muestra los 21 capítulos del CDX, una clasificación clínica de alto nivel; su función es únicamente referencial dentro del patrón general de resultados.*

[1] Índice entendido no como la gravedad clínica para el paciente, sino como una aproximación a la dificultad diagnóstica, generada mediante un modelo. No se utiliza como métrica de rendimiento directa, pero sí como dimensión auxiliar para la visualización de resultados.

Cada uno de los 450 casos del dataset fue extraído de un universo clínico mayor (9.600 casos), transformado mediante un **pipeline de ETL multietapa** documentado dentro del módulo `data29`. Este proceso incluyó: extracción desde fuentes médicas heterogéneas, limpieza semántica, codificación estandarizada (ICD-10, SNOMED CT), enriquecimiento con metadatos, deduplicación y validación de calidad.

En este contexto, la **validez del diagnóstico de referencia (GDX)** de cada caso —es decir, la hipótesis clínica que se considera "verdadera" a efectos de evaluación— no fue asumida directamente ni generada ad hoc, sino construida como una verdad clínica operacional. Aunque no se aplicó una revisión manual caso por caso por parte de un panel externo, el proceso incluyó validaciones cruzadas con ontologías médicas, curación selectiva de fuentes y validación contextual automatizada mediante LLMs.

> Sobre la univalencia del diagnóstico de referencia, es crucial reconocer una limitación inherente al diseño de este y otros benchmarks similares: la asunción de un único diagnóstico de referencia (GDX) para cada caso. La práctica clínica real a menudo admite la **equifinalidad diagnóstica**, donde múltiples condiciones pueden explicar razonablemente un mismo cuadro clínico, especialmente en fases tempranas. El modelo de un único GDX no tiene en cuenta esta complejidad. Un modelo de IA podría proponer un diagnóstico alternativo perfectamente válido desde el punto de vista clínico que, al no coincidir con el GDX predefinido, sería incorrectamente penalizado por el pipeline **PV4**. Por ejemplo, ante un cuadro de dolor torácico agudo, tanto una "pericarditis aguda" como una "disección aórtica" podrían ser hipótesis plausibles dependiendo de los matices. Si el GDX es el primero y el modelo prioriza el segundo, **PV4** lo registraría como un fallo en la primera posición. Esta limitación, aunque común, es particularmente relevante en un estudio que busca medir la sutileza del juicio clínico y debe ser tenida en cuenta al interpretar la tasa de acierto bruta.

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** [0.2813, 0.6521]
  - **llama3-openbiollm-70b-zgh** [0.3979, 0.6439]
  - **jonsnowlabs-medellama-3-8b-v2-0-bfs** [0.5052, 0.5971]
  - **medgemma-27b-text-it** [0.4318, 0.5782]

  Nota: El primer valor corresponde a una estimación de severidad realizada mediante juicio contextual con GPT; no es el objetivo central del estudio, pero contribuye a estructurar la representación de los resultados y facilita su interpretación. El segundo valor proviene de un modelo BERT que calcula similitud semántica entre enunciados.

- **PV1/PV2 (ICD10+BERT):** Este pipeline introdujo un artefacto metodológico crítico que se produce cuando el evaluador, limitado a una coincidencia de códigos estricta, penaliza una hipótesis diagnóstica por una discrepancia en la **granularidad taxonómica**, a pesar

de su validez clínica[2]. En esencia, la evaluación castiga una inferencia clínica superior por su incapacidad para resolver la subsunción semántica inherente a las jerarquías de codificación médica.

- **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.

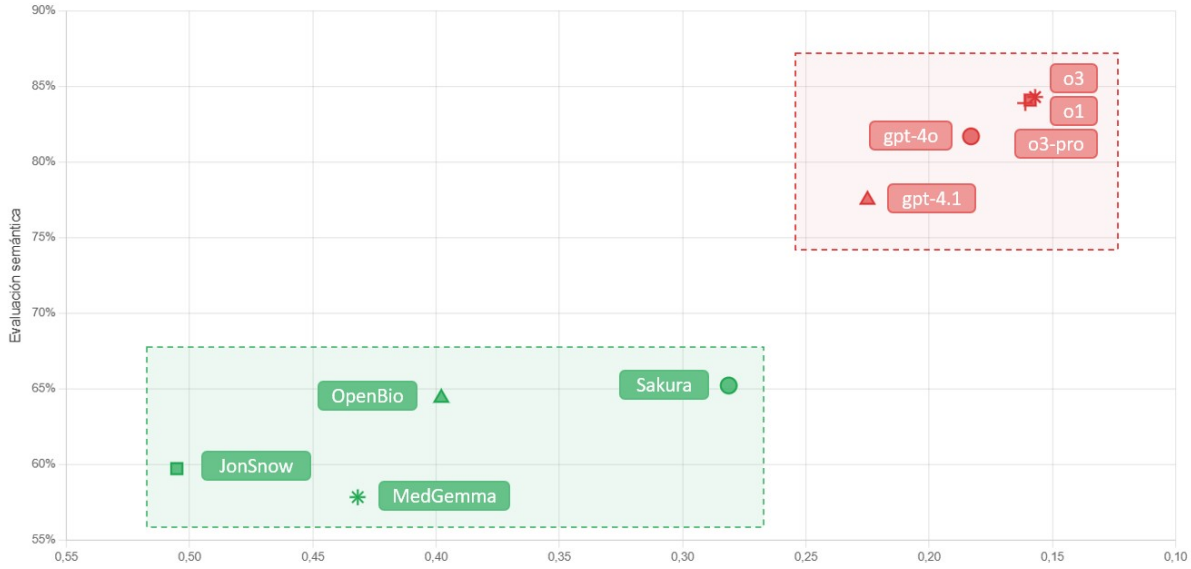

Figura 2. Comparación visual entre los resultados de PV0 y PV3, que representan los extremos del espectro evaluativo: desde una métrica puramente textual (BERT) hasta juicios contextuales realizados por LLMs.

Estas diferencias metodológicas no son triviales; de hecho, son la clave para entender el fenómeno de la saturación.

[2]Por ejemplo, si el modelo propone una entidad nosológica de alta especificidad como "Glomerulonefritis membranoproliferativa tipo I" y el diagnóstico de referencia está codificado a un nivel ontológico superior, como "Síndrome nefrítico crónico" (ICD-10: N03), el sistema acostumbrará a registrar un fallo.

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.

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 3, 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**.

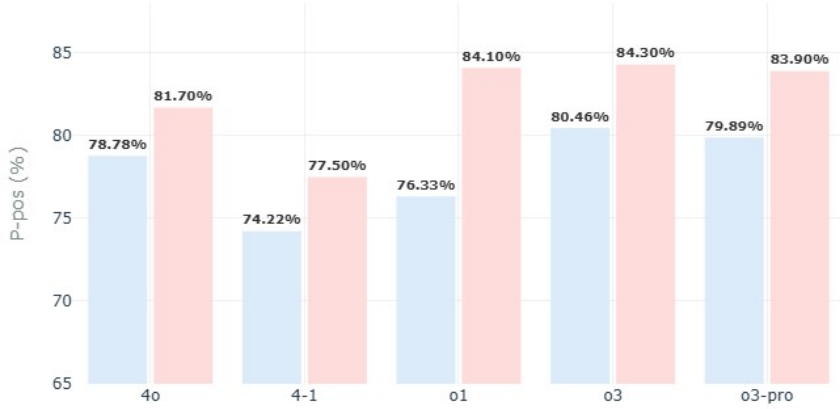

*Figura 3. Comparativa directa de resultados entre el método **ICD10+BERT (PV2)** y el Juez **LLM (PV3)**.*

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 priorización. **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 priorización 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 ↔ hijo). En consecuencia, **PV4** considera como acierto válido cualquier hijo o padre inmediato en la jerarquía (Figura 4), 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.

GDX \ DDX	Range	Category	Block	Sub-block	Group	Subgroup	Total
Range	0	0	0	0	0	0	0
Category	1	22	53	6	1	0	83
Block	9	136	908	296	28	0	1377
Sub-block	4	45	227	122	15	0	413
Group	1	6	75	83	126	0	291
Subgroup	0	2	2	2	0	0	6
Total	15	216	1293	509	170	0	2170

Baja frecuencia · Media frecuencia · Alta frecuencia · Coincidencias · Hijos

Figura 4. Matriz de transiciones jerárquicas entre el diagnóstico de referencia (GDX) y las propuestas del modelo (DDX), mostrando la frecuencia con que difieren en distintos niveles de la jerarquía CIE-10.

Como se observa en la Figura 4, la matriz de transiciones captura con qué frecuencia las predicciones del modelo (*DDX*) divergen del diagnóstico de referencia (*GDX*) a lo largo de los diferentes niveles jerárquicos[3] de la CIE-10: desde niveles generales como *Range* (rango) hasta niveles muy específicos como *Subgroup* (subgrupo).

Los colores indican la densidad de ocurrencias: celdas en rojo representan transiciones con alta frecuencia, mientras que los tonos más claros indican menor frecuencia. Es evidente que la mayoría de las discrepancias se dan entre niveles adyacentes, como *Categoría* → *Bloque* o *Bloque* → *Sub-bloque*, lo cual sugiere que, aunque no se alcance coincidencia exacta, las propuestas del modelo suelen ser jerárquicamente próximas y clínicamente relevantes.

Este hallazgo justifica el diseño de la métrica **PV4**, que no se limita a validar coincidencias exactas, sino que también reconoce como válidas aquellas predicciones que coinciden con el diagnóstico real en un nivel jerárquicamente relacionado: padre, hijo o hermano. Así, se evita penalizar al modelo por aciertos clínicamente válidos que no coinciden exactamente en código, pero sí en significado médico.

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.

[3]Ejemplo ilustrativo de la jerarquía adaptada: **range**: Lesiones y consecuencias externas → `S`; **category**: Lesiones en muñeca y mano → `S60--S69`; **block**: Fracturas a nivel de muñeca y mano → `S62`; **sub-block**: Fractura de metacarpiano (otro/no especificado) → `S62.3`; **group**: Fractura no especificada de metacarpiano → `S62.30`; **subgroup**: Fractura no especificada del segundo metacarpiano, encuentro inicial → `S62.301A`.

Para cada **DDX** individual, el pipeline (Figura 5) 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: Si la similitud **BERT** no alcanza el umbral de alta confianza ($< 0{,}89$), se activa un juicio competitivo. Se consulta un **Juez LLM** que evalúa la relación semántica entre el **DDX** y el **GDX**. Si ambos métodos coinciden en que hay un acierto, se elige el diagnóstico que aparece en una posición más alta dentro de la lista original de diferenciales, premiando la capacidad de priorización del modelo. Si solo uno lo valida, se acepta su veredicto. Si ninguno identifica una relación válida, 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.

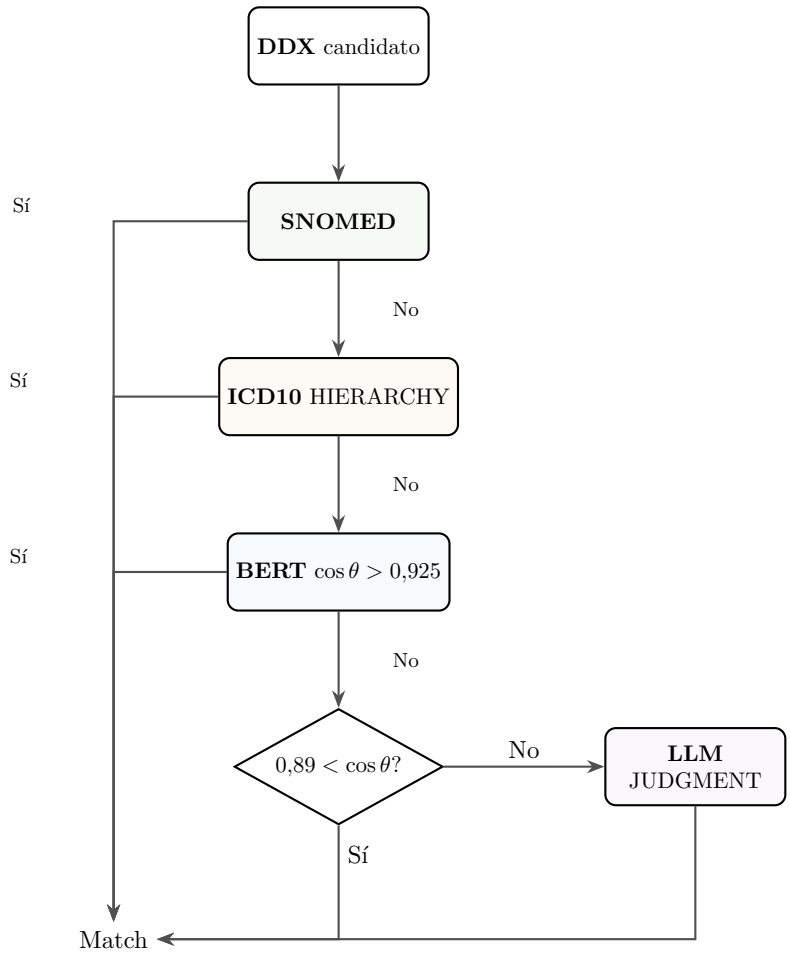

*Figura 5. Diagrama de flujo del proceso de evaluación de **PV4** para un único **DDX**, mostrando su cascada jerárquica.*

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 6) 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.

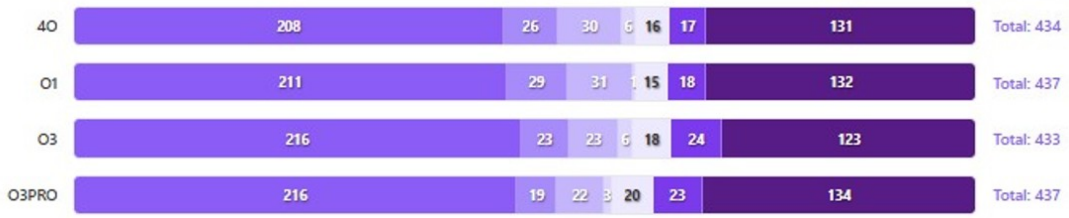

*Figura 6. Estudio preliminar del desglose de los métodos de resolución del pipeline **PV4**. Muestra la contribución de cada componente (**SNOMED match**, **ICD10 exact-sibling-parent**, **BERT autoconfirm**, **BERT match** y **LLM**) a la validación de aciertos.*

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. La Figura 7 ilustra 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.

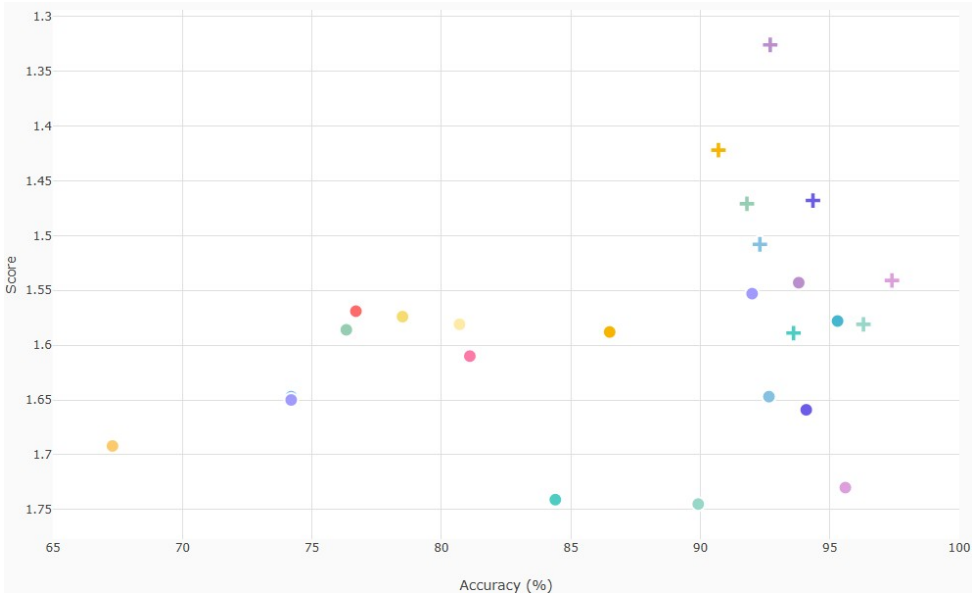

*Figura 7. Comparativa de rendimiento entre diferentes variantes de prompts. Las cruces indican ejecuciones con **o3** y los círculos con **4o**; los colores, distintos prompts (para una versión de esta gráfica con cada prompt identificado por su nombre, facilitando un análisis más granular, véase el **Anexo C**.).*

Un análisis más detallado sobre el tipo de salida solicitado en el prompt arroja conclusiones significativas (Tabla 2). Exigir un razonamiento explícito, como listar síntomas a favor y en contra, produce el mejor equilibrio entre posición promedio y tasa de acierto. En contraste, sobrecargar el modelo pidiendo campos como `confidence` o `rationale` degrada el rendimiento de la tarea principal.

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, se observó que la longitud y la complejidad del prompt interactúan de forma decisiva. Los prompts más largos (> 160 palabras) tienden a mejorar la tasa de acierto, pero este beneficio se anula si el formato de salida es demasiado complejo. Finalmente, incluir cláusulas "failsafe" en el prompt principal penaliza ligeramente el rendimiento, sugiriendo que la validación del tipo de input debe ser un paso previo.

6.3 Análisis comparativo y ranking final de modelos

El análisis culmina con la comparación directa de los modelos, utilizando los datos agregados de múltiples ejecuciones para mitigar la variabilidad. La Tabla 3 presenta los resultados consolidados.

*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

A primera vista, los resultados exhiben una dicotomía: **o3-pro** obtiene la máxima tasa de acierto, mientras que **o3** logra la mejor posición promedio. Un análisis estadístico y cualitativo es indispensable para resolver esta tensión y determinar el verdadero rendimiento.

6.3.1 Confianza y estabilidad estadísticas

La métrica más relevante para la utilidad clínica es la **posición promedio**, pues indica la capacidad de priorización del modelo. Aquí, **o3** lidera de forma contundente (1.474). Un test de significancia estadística (Prueba U de Mann-Whitney) contra su rival más cercano, **o3-pro** (1.597), arroja un **valor p $<$ 0.001**. Esto confirma que la superioridad de **o3** en la priorización del diagnóstico es **estadísticamente muy significativa** y no fruto del azar.

Esta calidad de priorización se refleja en el número de aciertos en primera posición (P1), donde **o3** también lidera con 311 casos, como se visualiza en la Figura 8.

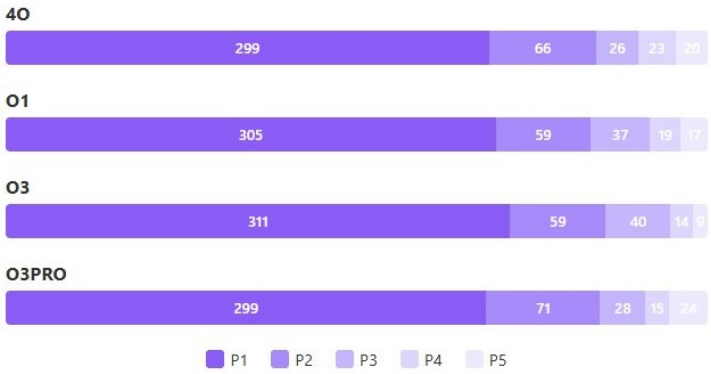

Figura 8. Distribución de los aciertos en las Top 5 posiciones por modelo.

Además, **o3** demuestra una **estabilidad** superior. Al comparar la variabilidad de su rendimiento frente al modelo más reciente, **4o**, se observa que el coeficiente de variación (CV) de la tasa de acierto de **o3** (0.026) es cuatro veces menor que el de **4o** (0.106). Esto convierte a **o3** en un modelo mucho más predecible y robusto para entornos de producción.

6.3.2 Cobertura en la tasa de aciertos

Aunque **o3-pro** presenta una tasa de acierto nominalmente superior (96.40 % vs. 93.65 %), un análisis estadístico (Test Z para dos proporciones) revela que esta diferencia no es significativa, con un **valor p de 0.11**. Al ser p $>$ 0.05, no podemos concluir que **o3-pro** sea realmente superior en cobertura; la diferencia observada es probablemente casual.

El desglose de los métodos de resolución (Figura 9) explica cómo **o3-pro** alcanza esta aparente ventaja. El modelo se apoya más en el "juicio semántico del LLM", un método más flexible, mientras que **o3** basa una mayor proporción de sus aciertos en la disciplina taxonómica de los códigos **SNOMED** e **ICD10**. En esencia, **o3-pro** çompraüna cobertura marginalmente mayor (y estadísticamente no significativa) a costa de sacrificar la precisión en el ranking y la rigurosidad metodológica.

Figura 9. 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 con el **rendimiento más estable y predecible**, con la mejor posición promedio y la menor variabilidad entre prompts. Esta tensión entre **cobertura (o3-pro) y calidad de priorización (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.

7.3 El riesgo del sesgo autorreferencial en el juicio semántico

El sistema de evaluación en tres etapas de PV4 reduce en gran medida las discrepancias determinísticas y semánticas que afectaban a evaluaciones anteriores. No obstante, el Nivel 3 sigue resolviendo una pequeña fracción de empates recurriendo a un juez LLM independiente. Dado que este juez es, en sí mismo, un modelo de lenguaje, existe —al menos en principio— la posibilidad de que favorezca respuestas cuya redacción, estilo de razonamiento o representaciones latentes se asemejen a las suyas propias, en lugar de valorar exclusivamente la corrección clínica.

Para asegurarnos de que este riesgo de "homofilia"se mantuviera en el plano teórico y no práctico, instrumentamos el sistema para registrar cada vez que se invocaba al juez LLM, junto con las respuestas candidatas y la justificación ofrecida por el juez. Posteriormente, analizamos una muestra de estos casos de forma manual y los contrastamos con etiquetas de referencia.

Dado que estas medidas de control formaron parte del flujo de análisis desde el principio, tenemos la confianza de que cualquier posible sesgo autorreferencial residual es insignificante en comparación con el desempeño diagnóstico observable. En resumen, el juez LLM actúa como un mecanismo de respaldo útil, sin distorsionar la clasificación general de los modelos.

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 10) 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.

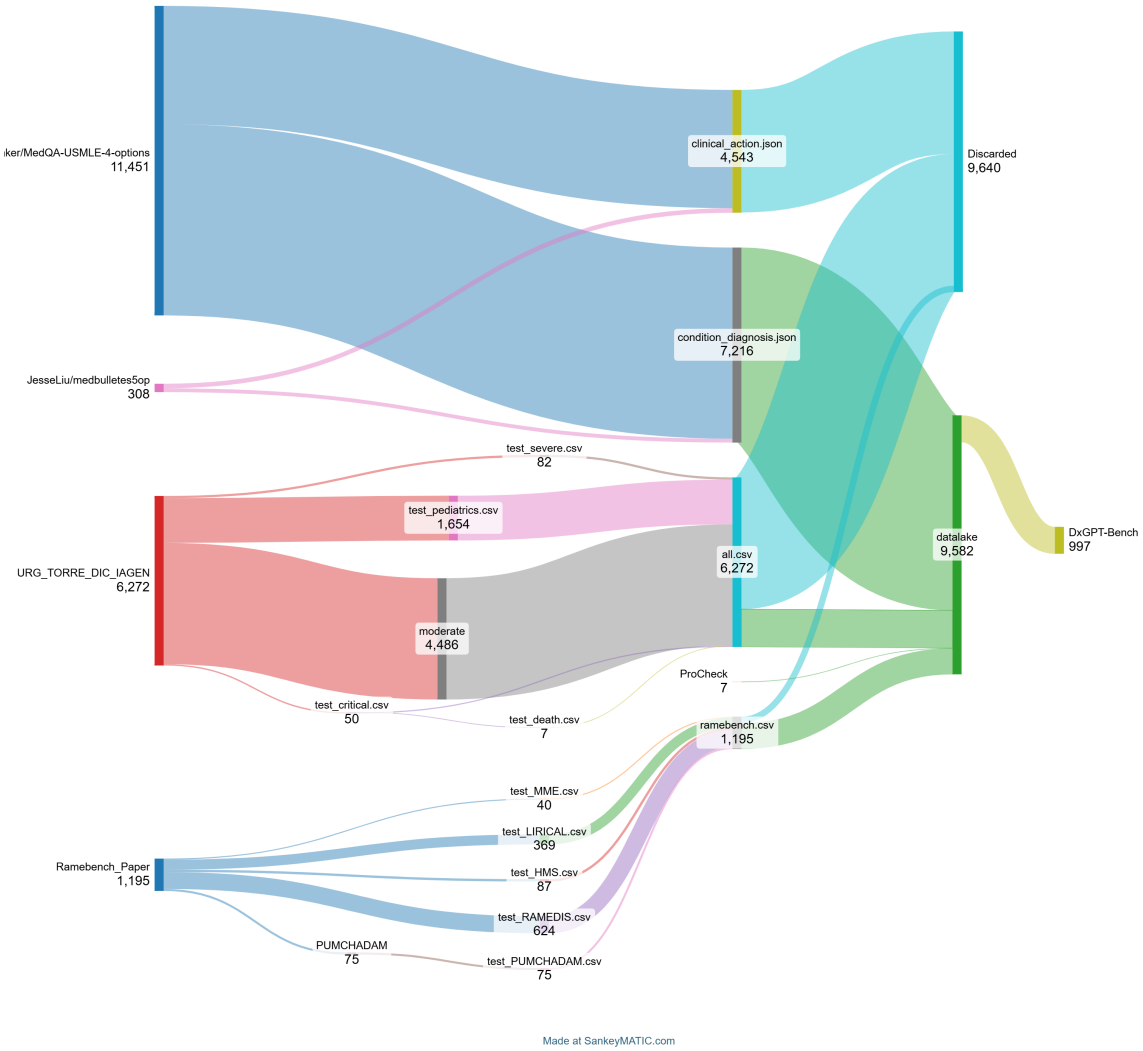

Figura 10. Diagrama de Sankey que visualiza el proceso de ETL para la composición del dataset de evaluación de 450 casos a partir de las fuentes originales.

B. Análisis detallado del rendimiento por prompt

La siguiente figura ofrece una versión detallada y etiquetada de la Figura 7 del cuerpo principal del informe. Esta visualización permite correlacionar directamente el rendimiento (posición promedio y tasa de acierto) de cada punto de datos con un prompt específico, cuyos textos completos se pueden consultar en el Anexo C.

Para garantizar la total transparencia y reproducibilidad de este estudio, el código fuente completo de todos los prompts, así como los scripts de evaluación y los datasets anonimizados, están disponibles en el repositorio público del proyecto en GitHub.

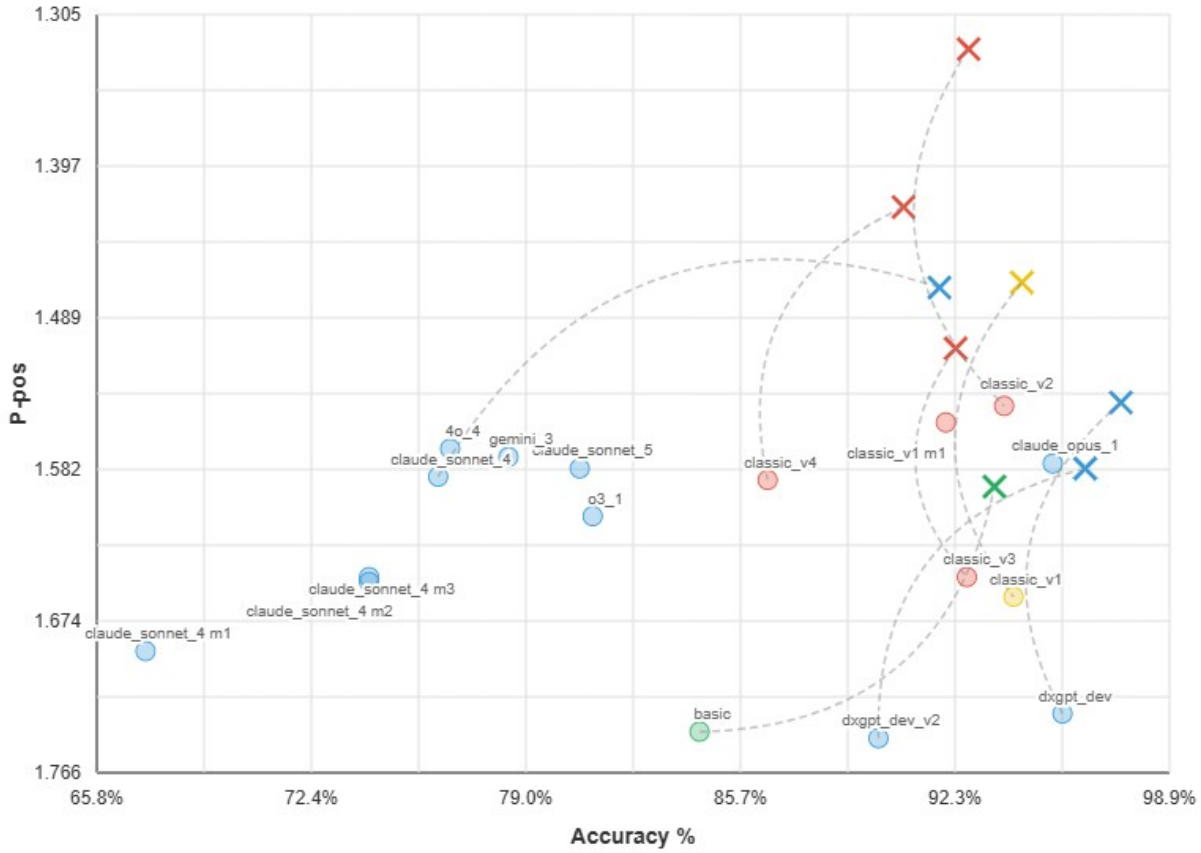

Figura 11. Comparativa etiquetada del rendimiento entre distintos prompts. Cada etiqueta identifica un prompt específico empleado durante las ejecuciones (los nombres pueden incluir referencias a modelos, pero únicamente reflejan la base sobre la que se realizaron ajustes evolutivos en el diseño del prompt, sin implicar un cambio de modelo subyacente).

C. 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).

C.1 Mejores prompts para o3 (TOP4)

classic_v2

Puntuación: 1.326 - 92.7 %

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

- 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}
```

classic_v4

Puntuación: 1.422 - 90.7 %

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

- 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}
```

classic_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}
```

C.2 Mejores prompts para 4o (TOP4)

classic_v2

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

classic_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}
```