HORIZON NEXUS JOURNAL
ISSN: 3073-1275

DOSO EDITORIAL CIENTÍFICA

Artículo

Evaluación comparativa de Claude y ChatGPT en la generación de consultas SQL

Comparative evaluation of Claude and ChatGPT in the generation of SQL queries

David Fabián Zúñiga Ortiz [1*], María José Cobeña Ureta [2], Victor Franklin Sánchez Alvarado [3*], Josselyn Valeria Flores Peña [4] y Jeannette Alexandra Laverde Mena [5]

[1] Escuela Politécnica Nacional, Ecuador, Quito; https://orcid.org/0000-0001-7541-0627
[2] Investigadora independiente, Ecuador, El Empalme; https://orcid.org/0009-0003-0891-8510, mariajose941@hotmail.es
[3] Ministerio de Educación, Deporte y Cultura, Ecuador, Quevedo; https://orcid.org/0009-0004-6567-4687, vito_frank@hotmail.com
[4] Unidad Educativa San Francisco de Asis, Ecuador, Valencia; https://orcid.org/0009-0001-8435-4425, jfloresp16@unemi.edu.ec
[5] Centro de Revisión Técnica Vehicular de Balzar, Ecuador, Balzar; https://orcid.org/0000-0002-1721-5679, jlaverde@asogrup.org

* Correspondencia: davidzunigaortiz92@gmail.com

https://doi.org/10.70881/hnj/v4/n2/138

Cita: Zúñiga Ortiz, D. F., Cobeña Ureta, M. J., Sánchez Alvarado, V. F., Flores Peña, J. V., & Laverde Mena, J. A. (2026). Evaluación comparativa de Claude y ChatGPT en la generación de consultas SQL. *Horizon Nexus Journal*, *4*(2), 171-190. https://doi.org/10.70881/hnj/v4/n2/138

Recibido: 04/05/2026
Revisado: 21/06/2026
Aceptado: 23/06/2026
Publicado: 24/06/2026

Resumen: La inteligencia artificial generativa ha cambiado la forma en la que funciona el desarrollo de software, sin embargo, la capacidad que tienen estos modelos para generar consultas SQL que sean correctas, optimizadas y robustas todavía no ha sido evaluada sistemáticamente en la literatura académica en español. Con el objetivo de llenar este vacío, se realizó una evaluación experimental comparativa entre dos asistentes de inteligencia artificial, Claude 4.6 Sonnet (Anthropic) y ChatGPT-4o (OpenAI), utilizando un esquema de base de datos relacional orientado a la gestión universitaria el cual se estandarizó y se crearon 12 casos de prueba los cuales se distribuyeron en tres niveles de complejidad: básico, intermedio y avanzado. Para la evaluación se aplicó una rúbrica de cinco dimensiones, corrección sintáctica, corrección lógica, optimización, manejo de casos borde y claridad de la explicación, con un puntaje máximo de 120 puntos. Los resultados obtenidos mostraron diferencias significativas entre ambos asistentes, más en las dimensiones de corrección lógica y en el manejo de casos borde, con diferencias que se notaron de forma progresiva en los niveles de mayor complejidad. Se concluye que ninguna herramienta es superior, y que la selección de estas herramientas debe de realizarse en función a los requerimientos.

Palabras clave: inteligencia artificial generativa; modelos de lenguaje de gran escala; generación de consultas SQL; evaluación comparativa; bases de datos relacionales.

Abstract: Generative artificial intelligence has changed the way software development works; however, the ability of these models to generate SQL queries that are correct, optimized, and robust has not yet been systematically evaluated in the Spanish-language academic literature. With the aim of filling this gap, a comparative experimental evaluation was conducted between two artificial intelligence assistants, Claude 4.6 Sonnet (Anthropic) and ChatGPT-4o (OpenAI), using a relational database schema oriented toward university management, which was standardized, and 12 test cases were created and distributed across three levels of complexity: basic, intermediate, and advanced. For the evaluation, a five-dimension rubric was applied—syntactic correctness, logical correctness, optimization, edge case handling, and clarity of explanation—with a maximum score

of 120 points. The results obtained showed significant differences between the two assistants, particularly in the dimensions of logical correctness and edge case handling, with differences that became progressively more noticeable at higher levels of complexity. It is concluded that neither tool is superior, and that the selection of these tools should be made based on specific requirements.

Keywords: generative artificial intelligence; large language models; SQL query generation; comparative evaluation; relational databases.

1. Introducción

La inteligencia artificial generativa es una de las transformaciones más importantes en el ámbito del desarrollo de software actual. Los modelos de lenguaje de gran escala (LLM) tienen gran capacidad para asistir en tareas que comprenden desde la escritura de código hasta la generación de documentación técnica, cambiando y optimizando los métodos y flujos de trabajo de los desarrolladores a nivel mundial (Brown et al., 2020; Zhao et al., 2023; Hou et al., 2023). Considerando este contexto, el lenguaje de consulta estructurado (SQL) es uno de los campos en el cual se realiza mayormente la aplicación práctica, debido a que prácticamente todo sistema de información actual necesita interacción con bases de datos, en su gran mayoría relacionales (Ramakrishnan & Gehrke, 2003; Codd, 1970).

Traducir lenguaje natural a consultas SQL, lo cual es conocido como text-toSQL sigue siendo uno de los problemas abiertos más importantes en el procesamiento de lenguaje natural. Conocer la sintaxis del lenguaje es solo el punto de partida; el principal obstáculo aparece cuando hay que interpretar cómo está organizada la base de datos, cuáles son las reglas de negocio y bajo qué criterios una consulta resulta más eficiente en la práctica (Deng et al., 2022; Qin et al., 2022). Los LLM han dado señales prometedoras en pruebas controladas, pero ese rendimiento cambia cuando el esquema se complica o cuando la consulta exige encadenar varias condiciones lógicas al mismo tiempo, ahí es donde los modelos empiezan a tener fallas. (Guo et al., 2019; Shi et al., 2024; Yu et al., 2018).

El recorrido por la literatura muestra un campo que crece rápido, pero que todavía tiene vacíos difíciles de ignorar. Por ejemplo, Guo et al. (2019) proponen el benchmark Spider, evidenciando algo que muchos sospechaban; cuando los esquemas involucran varias tablas relacionadas, la precisión de los sistemas text-to-SQL cae de forma bastante notoria. Años después, Rajkumar et al. (2022) pusieron a prueba GPT-3 en esa misma tarea y los resultados confirmaron el patrón, las consultas con JOIN

entre múltiples tablas seguían siendo un punto de inflación para el modelo. Por otra parte, Poesia et al. (2022) aportaron algo distinto: en lugar de quedarse con la corrección sintáctica como único criterio, propusieron verificar también si el código generado era lógicamente válido al ejecutarlo, lo que abrió una forma más honesta de medir el desempeño real.

Este trabajo parte precisamente de ese vacío. La idea principal fue diseñar un protocolo experimental que sea reproducible y verificable, y desde ahí comparar como se desempeñan los modelos Claude y ChatGPT-4 cuando se les pide generar consultas SQL sobre un mismo esquema relacional. La evaluación y comparación no estuvo limitada a revisar la correcta ejecución o no de la consulta obtenida por estos modelos; se evaluaron cinco aspectos en concretos, corrección sintáctica, corrección lógica, optimización, manejo de casos borde y claridad en la explicación entregada por cada modelo al usuario.

La hipótesis de este trabajo sostiene que los dos asistentes no se comportan igual a medida que la complejidad de lo que se les pide aumenta, y que esas diferencias son suficientemente consistentes como para ser medidas. El artículo está organizado de la siguiente forma: la sección 2 describe los materiales y el método seguido; la sección 3 reporta los resultados obtenidos; la sección 4 discute su significado; y la sección 5 cierra con las conclusiones y algunas recomendaciones prácticas.

2. Materiales y Métodos

2.1. Diseño del estudio

Para este estudio se optó por un diseño experimental comparativo-descriptivo, donde cada asistente de IA actuó como unidad independiente de análisis (Hernández-Sampieri et al., 2014; Creswell & Creswell, 2018). El enfoque fue cuantitativo con alcance explicativo-comparativo, lo que se buscaba, era detectar si existían diferencias reales en como respondía cada modelo cuando las condiciones de pruebas se mantenían fijas.

Todo se ejecutó bajo un entorno computacional controlado, sin participaciones externas que pudieran alterar los resultados, en términos metodológicos esto corresponde a una investigación de laboratorio.

Dado que en el presente estudio no se involucró personas como sujetos de análisis, no fue necesario tramitar consentimientos informado ni someter el protocolo a un comité de ética. Lo que si se cuidó fue que en cada una de las etapas el proceso fuera completamente transparente: los prompts utilizados, los criterios de evaluación y los datos obtenidos quedaron disponibles en un repositorio público de GitHub, de modo que cualquier investigador pueda revisar, auditar o replicar este experimento si lo considera pertinente.

2.2. Herramientas evaluadas y período de pruebas

Para el desarrollo de este estudio se emplearon dos asistentes de inteligencia artificial de uso general: Claude Sonnet 4.6, desarrollado por Anthropic, y GPT-4o, desarrollado por OpenAI. Ambas herramientas fueron consultadas únicamente a través de sus plataformas web oficiales, Claude.ai y chatgpt.com respectivamente. Con el objetivo de asegurar que cada interacción partiera sin información contextual previa, todas las sesiones fueron iniciadas desde cero.

La recolección de datos estuvo a cargo de cinco evaluadores independientes, quienes operaron bajo condiciones técnicas homogéneas: un único equipo de cómputo, la misma versión del sistema gestor de bases de datos y una conexión de red compartida. Esta estandarización del entorno buscó disminuir las variaciones atribuibles a factores externos como el sesgo propio de las apreciaciones individuales. Las pruebas fueron realizadas el 9 de junio de 2026, momento en que ambos asistentes se encontraban disponibles en las versiones públicas descritas previamente.

Para el desarrollo de la investigación se contó con dos ordenadores de escritorios de idénticas características: procesador Intel Core i7-11700 a 2,50 GHz, 16 GB de memoria RAM DDR4 a 2133 MT/s, una tarjeta gráfica NVIDIA GeForce GTX 1650 con 4 GB de memoria dedicada, y el sistema operativo Windows 11 Home de 64 bits (versión 25H2, compilación 26200.8457). Como sistema gestor de bases de datos se utilizó Microsoft SQL Server 2019 Express Edition arquitectura de 64 bits (build 15.0.2000.5), cuya administración se la realizó mediante SQL Server Management Studio en su versión 19.3.

2.3. Esquema de base de datos

Con el objetivo de asegurar condiciones equitativas para ambos asistentes y que los resultados puedan ser replicables por cualquier investigador, se elaboró un esquema relacional propio basado en un entorno académico universitario. El esquema quedó conformado por cinco tablas vinculadas entre sí a través de claves foráneas.

- ESTUDIANTE (id_estudiante, nombre, apellido, cedula, fecha_nacimiento, id_carrera);
- CARRERA (id_carrera, nombre_carrera, facultad, creditos_totales);
- MATERIA (id_materia, nombre_materia, creditos, id_carrera);
- MATRICULA (id_matricula, id_estudiante, id_materia, periodo, nota_final, estado);
- DOCENTE (id_docente, nombre, apellido, titulo, id_materia).

Las relaciones de clave foránea entre las tablas permitieron evaluar consultas que requerían operaciones JOIN, subconsultas y funciones de ventana, incrementando la complejidad progresiva del esquema.

2.4. Casos de prueba

Se elaboraron un total de 12 casos de prueba para esta evaluación, agrupados en tres niveles de dificultad: básico (C1–C4), intermedio (C5–C8) y avanzado (C9–C12). Para cada caso se definió un prompt estandarizado en español, enviado a ambos asistentes sin contexto adicional. El diseño de prompts estandarizados en lenguaje natural sigue las recomendaciones de ingeniería de prompts documentadas para tareas de generación de código (White et al., 2023). El prompt siguió el siguiente formato: *"Dado el siguiente esquema de base de datos [esquema], genera una consulta SQL que [requerimiento]. Explica brevemente cómo funciona la consulta."* Los casos de prueba se presentan en la Tabla 1.

Tabla 1. Casos de prueba por nivel de complejidad

#	Requerimiento de la consulta	Nivel
1	Listar todos los estudiantes con su nombre completo y carrera.	Básico
2	Obtener los estudiantes que aprobaron todas sus materias (nota >= 7).	Básico
3	Contar cuántos estudiantes hay por carrera.	Básico
4	Mostrar las materias con más de 3 créditos ordenadas de mayor a menor.	Básico
5	Obtener el promedio de notas por materia usando JOIN entre tablas.	Intermedio
6	Listar los 5 estudiantes con mejor promedio general.	Intermedio

7	Encontrar estudiantes matriculados en más de 3 materias el mismo periodo.	Intermedio
8	Obtener docentes que enseñan materias sin estudiantes matriculados.	Intermedio
9	Calcular el porcentaje de aprobación por carrera con subconsulta.	Avanzado
10	Listar estudiantes que reprobaron la misma materia más de una vez.	Avanzado
11	Obtener la materia con mayor variación de notas (desviación estándar).	Avanzado
12	Generar un ranking de estudiantes por facultad usando funciones de ventana.	Avanzado

2.5. Métricas de evaluación

Para cada caso de prueba se aplicó una rúbrica de evaluación conformada por cinco dimensiones, calificadas individualmente en una escala de 0 a 2 puntos, lo que equivale a un máximo de 10 puntos por caso y a un total de 120 puntos para el conjunto de los 12 casos evaluados. Esta forma de construcción de rúbricas analíticas constituye un método ampliamente utilizado para la evaluación objetiva en el ámbito de la investigación educativa y de sistemas (Brookhart, 2013). Las dimensiones evaluadas fueron las siguientes:

- **Corrección sintáctica (CS):** se verificó que la consulta SQL fuera ejecutable sin errores de sintaxis en el motor de base de datos (0 = error grave, 1 = error menor corregible, 2 = ejecución sin errores);
- **Corrección lógica (CL):** se comprobó que la consulta devolviera el resultado esperado al ejecutarse con datos de prueba reales (0 = resultado incorrecto, 1 = parcialmente correcto, 2 = completamente correcto);
- **Optimización (OP):** se evaluó que la consulta evitara redundancias, empleara índices apropiados y no realizara operaciones innecesarias (0 = muy ineficiente, 1 = aceptable, 2 = optimizada);
- **Manejo de casos borde (CB):** se verificó el comportamiento de la consulta ante valores NULL, tablas vacías o condiciones extremas (0 = no considera casos borde, 1 = considera algunos, 2 = manejo completo);
- **Claridad de explicación (CE):** se valoró que la explicación generada por la IA fuera correcta, comprensible y útil para el aprendizaje (0 = incorrecta o ausente, 1 = parcial, 2 = completa y correcta).

La rúbrica completa con los criterios por dimensión se presenta en la Tabla 2.

Tabla 2. Rúbrica de evaluación por dimensión

Dimensión	0 puntos	1 punto	2 puntos	Peso
CS - Corrección sintáctica	Error que impide ejecución	Error menor, fácil de corregir	Ejecuta sin errores	20%
CL - Corrección lógica	Resultado completamente erróneo	Resultado parcialmente correcto	Resultado 100% correcto	20%
OP - Optimización	Consulta redundante o muy lenta	Aceptable pero mejorable	Uso eficiente de joins e índices	20%
CB - Casos borde	No considera NULL ni vacíos	Considera algunos casos	Manejo completo y robusto	20%
CE - Claridad explicación	Incorrecta o ausente	Parcial o confusa	Clara, correcta y completa	20%

2.6. Procedimiento

El procedimiento experimental siguió los siguientes pasos: (1) se presentó el esquema completo de la base de datos al inicio de cada sesión nueva con cada asistente; (2) se envió el prompt estandarizado para cada caso de prueba; (3) se copió la consulta SQL generada y se ejecutó en un entorno SQL SERVER con datos de prueba precargados; (4) se registraron los resultados en la rúbrica de evaluación; (5) se repitió el proceso de forma independiente para el segundo asistente. Las pruebas fueron realizadas por cinco evaluadores independientes bajo el mismo entorno computacional durante un periodo de 3 días consecutivos.

3. Resultados

El análisis de los resultados se estructuró en torno a cinco criterios de evaluación, aplicadas sobre el conjunto de 12 casos de pruebas diseñados previamente. En cada una de estas dimensiones, las consultas generadas recibieron una puntuación de 0 a 2, lo que permitió alcanzar un total acumulado de 120 puntos. En los apartados siguientes se describen los resultados obtenidos para cada dimensión, considerando además los distintos niveles de complejidad establecidos.

3.1. Corrección sintáctica

Esta dimensión permitió determinar si las consultas SQL generadas por cada asistente podían ejecutarse correctamente en el motor de SQL Server, sin presentar errores. La

Tabla 3 recoge los resultados obtenidos para cada uno de los casos de prueba evaluados en este criterio.

Tabla 3. Puntuación de corrección sintáctica por caso de prueba

Caso	Nivel	Claude (0-2)	ChatGPT (0-2)	Ganador	Observación
C1	Básico	2	2	Empate	
C2	Básico	2	2	Empate	
C3	Básico	2	2	Empate	
C4	Básico	2	2	Empate	
C5	Intermedio	2	2	Empate	
C6	Intermedio	2	2	Empate	
C7	Intermedio	2	2	Empate	
C8	Intermedio	2	2	Empate	
C9	Avanzado	2	2	Empate	
C10	Avanzado	2	2	Empate	
C11	Avanzado	2	2	Empate	
C12	Avanzado	2	2	Empate	
TOTAL		24/24	24/24		

3.2. Corrección lógica

La corrección lógica evalúa si la consulta generada devuelve el resultado esperado cuando se ejecuta con datos de prueba reales. Esta dimensión es más exigente que la sintáctica, ya que requiere que la IA comprenda correctamente el requerimiento de negocio expresado en lenguaje natural.

Tabla 4. Puntuación de corrección lógica por caso de prueba

Caso	Nivel	Claude (0-2)	ChatGPT (0-2)	Ganador	Observación
C1	Básico	1	2	ChatGPT	
C2	Básico	1	2	ChatGPT	
C3	Básico	2	2	Empate	
C4	Básico	2	2	Empate	
C5	Intermedio	2	2	Empate	

C6	Intermedio	2	2	Empate	
C7	Intermedio	2	2	Empate	
C8	Intermedio	2	2	Empate	
C9	Avanzado	2	2	Empate	
C10	Avanzado	2	2	Empate	
C11	Avanzado	2	2	Empate	
C12	Avanzado	2	2	Empate	
TOTAL		22/24	24/24		

3.3. Optimización, casos borde y claridad

Para las dimensiones de Optimización (OP), Manejo de Casos Borde (CB) y Claridad de Explicación (CE) se aplicó el mismo formato de tabla que en las secciones anteriores (Tablas 5, 6 y 7). Para la optimización se evaluó si la consulta usa índices adecuados, evita SELECT * y no realiza subconsultas innecesarias. Para casos borde se verificó el comportamiento ante valores NULL y tablas vacías. Para claridad se valoró la comprensibilidad de la explicación generada por la IA.

Tabla 5. Puntuación de optimización por caso de prueba

Caso	Nivel	Claude (0-2)	ChatGPT (0-2)	Ganador	Observación
C1	Básico	2	2	Empate	
C2	Básico	2	1	Claude	La Consulta 2 es mejor para producción a gran escala porque usa WHERE.
C3	Básico	2	1	Claude	La consulta de Claude es más eficiente porque agrupa por la llave primaria indexada (id_carrera), lo que reduce el uso de CPU y memoria frente al agrupamiento por texto.
C4	Básico	2	2	Empate	
C5	Intermedio	2	1	Claude	Claude gana en optimización al agrupar por IDs indexados en vez de texto, acelerando el proceso en SQL Server, y su filtro WHERE reduce memoria al descartar notas nulas antes de calcular el promedio.

C6	Intermedio	2	1	Claude	Claude optimiza la consulta agrupando por el ID indexado (no por texto como ChatGPT) y filtra con WHERE los nulos antes de calcular, agilizando el promedio y el TOP 5.
C7	Intermedio	2	1	Claude	Claude optimiza la consulta agrupando por el ID indexado del estudiante, reduciendo el consumo de CPU al evitar comparar texto y evitando mezclar datos de alumnos con nombres idénticos.
C8	Intermedio	1	2	ChatGPT	ChatGPT gana: NOT EXISTS se detiene en la primera coincidencia, mientras que el NOT IN con DISTINCT de Claude procesa toda la tabla, es más lento y falla con nulos.
C9	Avanzado	1	2	ChatGPT	Ganó ChatGPT: calcula los porcentajes de forma directa con NULLIF, mientras que Claude usa subconsultas anidadas innecesarias y comete un error lógico al buscar la carrera en la tabla de materias.
C10	Avanzado	2	1	Claude	Ganó Claude: agrupa por las llaves primarias (id_estudiante, id_materia), permitiendo a SQL Server usar índices numéricos en vez de texto, y su WHERE descarta nulos antes de agrupar.
C11	Avanzado	2	1	Claude	Ganó Claude: usa HAVING COUNT(*) >= 2 para evitar que STDEV falle con menos de dos notas, y agrupa eficientemente por la llave primaria id_materia.
C12	Avanzado	2	1	Claude	Ganó Claude: agrupa por el ID indexado del estudiante para mayor rendimiento, y su WHERE filtra notas nulas antes de aplicar RANK, evitando procesarlas innecesariamente.
TOTAL		22/24	16/24		

Tabla 6. Puntuación de manejo de casos borde por caso de prueba

Caso	Nivel	Claude (0-2)	ChatGPT (0-2)	Ganador	Observación
C1	Básico	2	1	Claude	Ganó Claude: incluir id_estudiante en el SELECT permite usar el índice clúster, útil si la consulta se reutiliza como vista o subconsulta.
C2	Básico	2	1	Claude	Ganó Claude: maneja bien a estudiantes sin notas, mientras que ChatGPT los muestra erróneamente como si hubieran aprobado todo.
C3	Básico	2	1	Claude	Ganó Claude: COUNT(E.id_estudiante) da 0 para carreras sin alumnos, mientras que COUNT(*) de ChatGPT genera falsamente 1.
C4	Básico	2	2	Empate	
C5	Intermedio	2	1	Claude	Ganó Claude: su filtro WHERE nota_final IS NOT NULL evita que materias sin calificación distorsionen el promedio.
C6	Intermedio	2	1	Claude	Ganó Claude: evita promediar nulos, resuelve empates por apellido, y ChatGPT mezcla homónimos al no validar el ID.
C7	Intermedio	2	1	Claude	Ganó Claude: agrupar por ID evita fusionar alumnos homónimos y sumar materias erróneamente.
C8	Intermedio	1	2	ChatGPT	Ganó ChatGPT: NOT EXISTS maneja bien los nulos, mientras que el NOT IN de Claude falla y devuelve cero registros si hay un solo nulo.
C9	Avanzado	1	2	ChatGPT	Ganó ChatGPT: usa NULLIF para evitar división por cero, mientras Claude no se protege y hereda errores lógicos.
C10	Avanzado	2	1	Claude	Ganó Claude: su filtro IS NOT NULL limpia

					registros vacíos y agrupar por ID evita que alumnos homónimos alteren las estadísticas del otro.
C11	Avanzado	2	1	Claude	Ganó Claude: usa HAVING COUNT(*) >= 2 para evitar que STDEV falle si una materia tiene solo una nota.
C12	Avanzado	2	1	Claude	Ganó Claude: su filtro de nulos evita que alumnos sin notas aparezcan en el ranking, y maneja bien homónimos.
TOTAL		22/24	16/24		

Tabla 7. Puntuación de claridad de explicación por caso de prueba

Caso	Nivel	Claude (0-2)	ChatGPT (0-2)	Ganador	Observación
C1	Básico	2	1	Claude	Claude gana por estructurar la explicación con tablas y un diagrama relacional muy didáctico.
C2	Básico	2	1	Claude	Claude gana por incluir alertas técnicas sobre nulos y una tabla comparativa didáctica entre WHERE y HAVING.
C3	Básico	2	1	Claude	Claude gana por contrastar didácticamente los JOINs y alertar del peligro de usar COUNT(*) con nulos.
C4	Básico	2	1	Claude	Claude gana al justificar el ordenamiento determinista y explicar conceptualmente la diferencia entre WHERE y HAVING.
C5	Intermedio	2	1	Claude	Claude gana por incluir alertas de diseño relacional, comportamiento de nulos y un diagrama de flujo.
C6	Intermedio	2	1	Claude	Claude gana por detallar el orden de ejecución de TOP, el uso de WITH TIES y portabilidad.
C7	Intermedio	2	1	Claude	Claude gana por ejemplificar visualmente con una tabla el comportamiento de la agrupación combinada.

C8	Intermedio	2	1	Claude	Claude gana por explicar el impacto de los NULLs en subconsultas y comparar múltiples técnicas.
C9	Avanzado	2	1	Claude	Claude gana al desglosar las subconsultas en capas didácticas con tablas de resultados intermedios.
C10	Avanzado	2	1	Claude	Claude gana en claridad al explicar la lógica relacional con tablas paso a paso y mostrar cómo afecta el orden de filtrado (WHERE vs HAVING).
C11	Avanzado	2	1	Claude	Claude destaca al explicar la métrica estadística con ejemplos prácticos y comparar con precisión las funciones STDEV y STDEVP.
C12	Avanzado	2	1	Claude	Claude obtiene la puntuación máxima al comparar de forma didáctica el comportamiento de los tres métodos de numeración en casos de empate.
TOTAL		24/24	12/24		

3.4. Tabla resumen comparativa

La Tabla 8 consolida los resultados de las cinco dimensiones para ambos asistentes, mostrando el puntaje total obtenido sobre 120 puntos posibles (12 casos × 5 dimensiones × 2 puntos).

Tabla 8. Resumen comparativo por dimensión y score global

Dimensión	Claude (pts)	ChatGPT (pts)	Máx.	Dif.	Ganador
Corrección sintáctica (CS)	24	24	24	0	Empate
Corrección lógica (CL)	22	24	24	-2	ChatGPT
Optimización (OP)	22	16	24	+6	Claude
Casos borde (CB)	22	16	24	+6	Claude
Claridad explicación (CE)	24	12	24	+12	Claude
SCORE GLOBAL	114	92	120 (100%)	+22	Claude

3.5. Análisis por nivel de complejidad

La Tabla 9 desagrega el score global según el nivel de complejidad de la consulta (básico, intermedio, avanzado), con el objetivo de identificar si las diferencias entre asistentes se acentúan o disminuyen a medida que aumenta la complejidad del requerimiento.

Tabla 9. Score por nivel de complejidad (puntos sobre 40 por nivel)

Nivel	Claude (pts/40)	ChatGPT (pts/40)	Diferencia
Básico (C1–C4)	38	31	+7
Intermedio (C5–C8)	38	30	+8
Avanzado (C9–C12)	38	30	+8

4. Discusión

Los resultados obtenidos evidencian que Claude 4.6 Sonnet y ChatGPT- 4o presentan un patrón distinto de aciertos y limitaciones en la elaboración de consultas SQL. Respecto a la corrección sintáctica, ambos modelos alcanzan un rendimiento igual (24/24) en los casos de nivel básico, lo cual concuerda con lo señalado por Brown et al. (2020), quienes indican que los modelos de lenguaje de gran escala han alcanzado un dominio sólido en la generación de código con estructuras sencillas. No obstante, tal como se evidencia en la Tabla 8, la brecha entre ambos asistentes se sostiene de manera constante conforme aumenta la complejidad de las consultas, lo cual coincide con lo planteado por Guo et al. (2019), quienes señalan que la exactitud de los sistemas text-to-SQL se reduce de forma considerable frente a esquemas con múltiples tablas y condiciones lógicas combinadas.

Al observar la Tabla 8, se aprecia que la correlación lógica fue el aspecto donde ambas herramientas mostraron mayor distancia entre sí, con 2 puntos de diferencia sobre un total de 24. Este resultado da soporte parcial a la hipótesis planteada inicialmente en esta investigación, según la cual cabía esperar diferencias notables entre los dos asistentes dependiendo del grado de complejidad de las tareas. Tal dificultad para llevar un requerimiento de negocio, formulado en lenguaje natural, hacia una consulta SQL correcta en términos lógicos, va en línea con lo planteado por Deng et al. (2022), quienes indican que los principales desafíos del problema text-to-SQL residen

precisamente en la fase de comprensión semántica del enunciado, no en la generación sintáctica (Zhong et al., 2017).

En cuanto a la optimización, los resultados obtenidos muestran que Claude es más eficiente principalmente en los niveles intermedio y avanzado. Esta característica es importante especialmente en entornos de producción ya que ahí se manejan grandes volúmenes de datos, ya que una consulta poco eficiente puede afectar negativamente el rendimiento general del sistema (Ramakrishnan & Gehrke, 2003; Date, 2019). Precisamente, la capacidad de recomendar el correcto uso de índices, suprimir subconsultas, relaciones innecesarias y recurrir a funciones cuando es pertinente, marca una diferencia entre una respuesta funcional y una respuesta de calidad profesional, distinción que este estudio logra medir de forma objetiva.

Respecto al manejo de casos de borde, los resultados presentados en la Tabla 8 evidencian una considerable diferencia entre ambos asistentes en este punto: Claude obtuvo 22 puntos y ChatGPT 16 puntos, sobre un total posible de 24, lo que nos indica que este último presenta algunas dificultadas en este criterio. Este resultado toma importancia considerando que en base de datos los valores NULL y las tablas vacías son situaciones habituales y reales, y que una consulta que no maneja de forma adecuada esto puede generar resultados incorrectos de forma silenciosa, sin que el desarrollador note un error evidente.

Por último, la dimensión de claridad en la explicación toma una relevancia bastante especial en contextos educativos debido a que estudiantes de programación hacen uso de estas herramientas no solo para obtener código, sino también para comprender los fundamentos teóricos detrás de cada consulta (Rajkumar et al., 2022; Kasneci et al., 2023). Los resultados obtenidos señalan que Claude ofrece explicaciones más completas y precisas, lo que apunta a una ventaja diferencial en su uso como apoyo al aprendizaje en cursos de bases de datos. Cabe destacar que una explicación incorrecta puede causar confusiones sobre el funcionamiento de SQL, riesgo que también es señalado por Shi et al. (2024).

En cuanto a limitaciones, este estudio contó con la participación de 5 evaluadores independientes que trabajaron bajo un entorno computacional estandarizado, esto contribuye a reducir sesgos subjetivos en las diferentes dimensiones como la claridad y la optimización. Sin embargo, las pruebas se realizaron exclusivamente usando el

sistema gestor de base de datos SQL Server y con prompts en español, teniendo en cuenta esto lo resultados obtenidos no pueden generalizarse de manera directa con otros motores de bases de datos ni otros idiomas. Para futuras investigaciones se recomienda que realizar evaluación por múltiples jueces, junto con el cálculo de nivel de concordancia entre ellos, esto constituye una práctica recomendada para garantizar la validez de este tipo de instrumentos (Landis & Koch, 1977)

5. Conclusiones

El presente estudio logró su objetivo principal al realizar una evaluación comparativa del rendimiento de los asistentes de inteligencia artificial Claude 4.6 Sonnet y ChatGPT-4o en la creación de consultas SQL, utilizando un protocolo experimental controlado, reproducible y aplicado en español, un aspecto que no había sido abordada sistemáticamente en la literatura académica previa.

La contribución clave de la presente investigación radica en el desarrollo y validación de una rubrica de evaluación que abarca cinco dimensiones: corrección sintáctica, corrección lógica, optimización, manejo de casos borde y claridad de explicación, aplicada sobre un esquema relacional estandarizado con 12 casos de prueba de 3 niveles y diferentes complejidades que van creciendo a medida que suben de nivel. Este instrumento representa una aportación metodológica replicable para investigaciones futuras que busquen evaluar asistentes de inteligencia artificial en tareas de generación de código técnico.

Los resultados obtenidos permiten concluir que ambas herramientas presentan un desempeño alto y bastante similar en las dimensiones de evaluación de nivel básico (38/40 ambas), pero las diferencias se evidencian a medida que la complejidad incrementa. La corrección lógica y el manejo de casos borde resultan ser las dimensiones con más diferencias entre ambos asistentes, lo que implica que la elección de la herramienta debe de considerar el tipo de tarea a realizar y no asumir equivalencia de desempeño para todos los contextos.

Desde un punto de vista práctico, los resultados orientan a tres perfiles de usuarios. Los programadores en entornos de producción, se debería de priorizar la dimensión de optimización al evaluar qué asistente utilizar, teniendo en cuenta que la ineficiencia de una consulta con grandes volúmenes de datos puede llegar a comprometer el rendimiento del sistema. Los estudiantes de informática se benefician especialmente

de la dimensión de claridad de explicación, está dimensión es útil ya que les permite a los estudiantes a determinar si el asistente de inteligencia artificial contribuye efectivamente a su aprendizaje o simplemente entrega código sin comprensión. Mientras que los docentes, disponen de un marco de evaluación de objetivos para incorporar o comparar estas herramientas en sus entornos de enseñanza de base de datos.

Entre las limitaciones del presente estudio se reconoce que, si bien la evaluación fue realizada por cinco evaluadores de forma independiente, todas las pruebas fueron ejecutadas utilizando exclusivamente SQL Server y con prompts completamente en español. Estas condiciones limitan la posibilidad de extender los resultados a otros motores de bases de datos, otros idiomas, o versiones de modelos analizados posteriormente.

Como líneas de investigación futura se sugiere ampliar a otros motores de bases de datos como PostgreSQL, Oracle y MySQL, incrementar el número de evaluadores agregando un cálculo formal. Incluir otros asistentes de inteligencia artificial como Gemini o GitHub Copilot para fortalecer la comparación, y analizar de forma más profunda como influye el idioma del prompt en la calidad de las consultas generadas por cada una de las herramientas.

Contribución de los autores: Conceptualización, DFZO; metodología, MJCU y VFSA; análisis formal, JALM y JVFP; investigación, DFZO y JVFP; redacción del borrador original, DFZO; redacción, revisión y edición, DFZO y JALM; supervisión, MJCU y VFSA. Todos los autores han leído y aceptado la versión final del manuscrito.

Financiamiento: El proceso investigativo no ha recibido financiación externa.

Conflicto de intereses: Los autores declaran no tener ningún conflicto de intereses

Declaración de disponibilidad de los datos: Los datos están disponibles previa solicitud a los autores de correspondencia: davidzunigaortiz92@gmail.com

Referencias Bibliográficas

Brookhart, S. M. (2013). How to Create and Use Rubrics for Formative Assessment and Grading. ASCD.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language Models are Few-Shot Learners (Versión 4). arXiv. https://doi.org/10.48550/ARXIV.2005.14165

Codd, E. F. (1970). A relational model of data for large shared data banks. Communications of the ACM, 13(6), 377-387. https://doi.org/10.1145/362384.362685

Creswell, J. W., & Creswell, J. D. (2018). Research design: Qualitative, quantitative, and mixed methods approaches (Fifth edition). SAGE.

Date, C. J. (2019). Database Design and Relational Theory: Normal Forms and All That Jazz. Apress. https://doi.org/10.1007/978-1-4842-5540-7

Deng, N., Chen, Y., & Zhang, Y. (2022). Recent Advances in Text-to-SQL: A Survey of What We Have and What We Expect (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2208.10099

Guo, J., Zhan, Z., Gao, Y., Xiao, Y., Lou, J.-G., Liu, T., & Zhang, D. (2019). Towards Complex Text-to-SQL in Cross-Domain Database with Intermediate Representation. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 4524-4535. https://doi.org/10.18653/v1/P19-1444

Hernández-Sampieri, R., Fernández-Collado, C., & Baptista-Lucio, P. (2014). Metodología de la investigación (6.a ed.). McGraw-Hill.

Hou, X., Zhao, Y., Liu, Y., Yang, Z., Wang, K., Li, L., Luo, X., Lo, D., Grundy, J., & Wang, H. (2023). Large Language Models for Software Engineering: A Systematic Literature Review (Versión 6). arXiv. https://doi.org/10.48550/ARXIV.2308.10620

Kasneci, E., Sessler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., Gasser, U., Groh, G., Günnemann, S., Hüllermeier, E., Krusche, S., Kutyniok,

G., Michaeli, T., Nerdel, C., Pfeffer, J., Poquet, O., Sailer, M., Schmidt, A., Seidel, T., … Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. Learning and Individual Differences, 103, 102274. https://doi.org/10.1016/j.lindif.2023.102274

Landis, J. R., & Koch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. Biometrics, 33(1), 159. https://doi.org/10.2307/2529310

Poesia, G., Polozov, O., Le, V., Tiwari, A., Soares, G., Meek, C., & Gulwani, S. (2022). Synchromesh: Reliable code generation from pre-trained language models (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2201.11227

Qin, B., Hui, B., Wang, L., Yang, M., Li, J., Li, B., Geng, R., Cao, R., Sun, J., Si, L., Huang, F., & Li, Y. (2022). A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2208.13629

Rajkumar, N., Li, R., & Bahdanau, D. (2022). Evaluating the Text-to-SQL Capabilities of Large Language Models (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2204.00498

Ramakrishnan, R., & Gehrke, J. (2003). Database management systems (3. ed., internat. ed). McGraw-Hill.

Shi, L., Tang, Z., Zhang, N., Zhang, X., & Yang, Z. (2024). A Survey on Employing Large Language Models for Text-to-SQL Tasks (Versión 5). arXiv. https://doi.org/10.48550/ARXIV.2407.15186

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT (Versión 1). arXiv. https://doi.org/10.48550/ARXIV.2302.11382

Yu, T., Zhang, R., Yang, K., Yasunaga, M., Wang, D., Li, Z., Ma, J., Li, I., Yao, Q., Roman, S., Zhang, Z., & Radev, D. (2018). Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 3911-3921. https://doi.org/10.18653/v1/D18-1425

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., … Wen, J.-R. (2023). A Survey of Large Language Models (Versión 19). arXiv. https://doi.org/10.48550/ARXIV.2303.18223

Zhong, V., Xiong, C., & Socher, R. (2017). Seq2SQL: Generating Structured Queries from Natural Language using Reinforcement Learning (Versión 7). arXiv. https://doi.org/10.48550/ARXIV.1709.00103