

Facultad de Ingeniería en Ingeniería en Electricidad y Computación

**MEJORA DEL PROCESO DE DESARROLLO DE CONSULTAS MYSQL
CON IA GENERATIVA EN UNA EMPRESA DE SOFTWARE
Proyecto de Titulación**

Previo la obtención del Título de:

Magíster en Sistemas de Información Gerencial

Presentado por:
Edgar Leonardo Holguín Figueroa
Joel Andrés Espinoza Delgado

Guayaquil - Ecuador

Año: 2025

Dedicatoria

El presente proyecto lo dedico con todo mi amor y gratitud a mis padres, quienes desde el cielo son mi guía. A mi familia, gracias por su apoyo incondicional, fueron el pilar fundamental para la culminación de este logro.

Ing, Edgar Holguín.

Le dedico no solo la tesis realizada sino todo el postgrado a mi familia y a todos los que confiaron en mí, esto es para ustedes, gracias por confirmar y estar ahí.

Ing, Joel Espinoza.

Agradecimientos

Agradezco profundamente a Dios por brindarme la fuerza, la sabiduría y la fe necesaria para perseverar y alcanzar esta meta tan significativa. A mis padres, que ahora descansan en el cielo: sé que, dondequiero que estén, se sentirán orgullosos de este logro. A toda mi familia, gracias por su apoyo incondicional y amor constante. Y a mis amigos, por estar siempre presentes, ofreciendo compañía, aliento y sonrisas en los momentos más desafiantes.

Ing, Edgar Holguín

Primero le doy gracias a Dios por brindarme la sabiduría y fuerza de voluntad necesaria para perseverar. A mis padres, que me han apoyado y creído en mi durante este nuevo desafío. También a todos que han estado presentes brindando su ayuda a todos ellos les agradezco por estar ahí y confiar en mi.
Ing, Joel Espinoza

Declaración Expresa

Nosotros Edgar Leonardo Holguín Figueroa y Joel Andrés Espinoza Delgado a acordamos y reconocemos que:

La titularidad de los derechos patrimoniales de autor (derechos de autor) del proyecto de graduación corresponderá al autor o autores, sin perjuicio de lo cual la ESPOL recibe en este acto una licencia gratuita de plazo indefinido para el uso no comercial y comercial de la obra con facultad de sublicenciar, incluyendo la autorización para su divulgación, así como para la creación y uso de obras derivadas. En el caso de usos comerciales se respetará el porcentaje de participación en beneficios que corresponda a favor del autor o autores.

La titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, secreto industrial, software o información no divulgada que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada por mí/nosotros durante el desarrollo del proyecto de graduación, pertenecerán de forma total, exclusiva e indivisible a la ESPOL, sin perjuicio del porcentaje que me/nos corresponda de los beneficios económicos que la ESPOL reciba por la explotación de mi/nuestra innovación, de ser el caso.

En los casos donde la Oficina de Transferencia de Resultados de Investigación (OTRI) de la ESPOL comunique a los autores que existe una innovación potencialmente patentable sobre los resultados del proyecto de graduación, no se realizará publicación o divulgación alguna, sin la autorización expresa y previa de la ESPOL.

Guayaquil, 15 de agosto del 2025.

Edgar Leonardo
Holguín Figueroa

Joel Andrés Espinoza
Delgado

Evaluadores

Nombre del Profesor**Msc. Juan Carlos García Plua**

Profesor/Tutor de proyecto

Nombre del Profesor**Msc. José Luis Asencio**

Revisor de proyecto

Resumen

El presente proyecto tiene como objetivo mejorar la eficiencia y autonomía del equipo de desarrollo en una empresa de software, mediante la integración de Inteligencia Artificial Generativa y Modelado de Procesos de Negocio (BPMN) en el desarrollo de consultas MySQL. Se plantea como hipótesis que el uso de herramientas como ChatGPT, combinadas con un rediseño estructurado del flujo de trabajo, reduce significativamente los tiempos de desarrollo y los errores comunes. El proyecto se justifica por la necesidad de optimizar procesos repetitivos, reducir la dependencia del DBA y mejorar la calidad del código entregado.

Para su desarrollo, se aplicó una metodología de tipo descriptiva no experimental. Se realizó el levantamiento del proceso actual (modelo AS-IS), se evaluaron alternativas de mejora y se construyó un modelo optimizado (TO-BE) con herramientas de uso corporativo como Bizagi Modeler, DBeaver y ChatGPT. Además, se llevaron a cabo simulaciones, encuestas y pruebas controladas con 10 desarrolladores.

Los resultados evidenciaron una reducción promedio del tiempo de desarrollo a menos de un día, disminución de errores de sintaxis y una mejora en la percepción de autonomía del equipo. La propuesta representa una solución viable y escalable para entornos técnicos similares.

Palabras clave:

BPMN, Inteligencia Artificial, Consultas SQL, Optimización de procesos

Abstract

This project aims to improve the efficiency and autonomy of the development team in a software company by integrating Generative Artificial Intelligence and Business Process Modeling Notation (BPMN) into the creation of MySQL queries. The hypothesis is that the use of tools such as ChatGPT, combined with a structured redesign of the workflow, significantly reduces development time and common coding errors. The project is justified by the need to optimize repetitive tasks, decrease reliance on the DBA, and improve the quality of delivered scripts.

The project was developed using a descriptive, non-experimental methodology. The current process (AS-IS) was mapped, multiple improvement alternatives were evaluated, and an optimized process model (TO-BE) was designed using tools like Bizagi Modeler, DBeaver, and ChatGPT. Simulations, user surveys, and controlled testing with ten developers were also conducted.

The results showed an average reduction in development time to less than one day, a decrease in syntax errors, and improved perceptions of autonomy among the development team. The proposed solution proves to be viable and scalable for similar technical environments.

Keywords:

BPMN, Generative AI, SQL Query Development, Process Optimization

Índice general

Resumen	I
Abstract	II
Índice general	III
Abreviaturas	V
Simbología	VI
Índice de figuras	VII
Índice de tablas	VIII
CAPÍTULO 1	1
1.INTRODUCCIÓN	1
1.1 Descripción del Problema	1
1.2 Justificación del Problema	1
1.3 Objetivos	3
1.3.1 Objetivo general	3
1.3.2 Objetivos específicos	3
1.4 Marco teórico	3
1.4.1 Desarrollo de consultas MySQL	3
1.4.2 Inteligencia Artificial Generativa en Desarrollo de Software	4
1.4.3 Modelado de Procesos con BPMN	4
1.4.4 Revisión de trabajos similares	5
CAPÍTULO 2	7
2. METODOLOGÍA.	7
2.1 Enfoque y Diseño Metodológico.	7
2.2 Levantamiento y Modelado ASIS	7
2.2.1 Descripción General	7
2.2.2 Detalles y Casos de Uso	7
2.3 Formulación de alternativas de Solución	14
2.4 Matriz de decisión y selección.	14
2.4.1 Criterios de Evaluación	14
2.4.2 Matriz de decisión	14

2.5 Análisis Costos de la alternativa seleccionada	15
2.6 Justificación de la selección	15
2.7 Modelado TO-BE	16
2.7.1 Desarrollo de un proceso de actualización masiva de registro por bloques.	20
2.7.2 Consulta entre varias tablas primarias para validar información.	21
2.8 Consideraciones Éticas y Legales	22
2.9 Especificaciones Técnicas para el uso de las herramientas	22
3. ANALISIS DE RESULTADOS.	24
3.1 Análisis de rediseño del proceso (modelado TO-BE)	24
3.2 Análisis de limitaciones	25
3.3 Análisis de encuestas a desarrolladores	26
3.4 Análisis de casos de pruebas simulados	27
3.5 Discusión de resultados	29
4. Conclusiones y Recomendaciones	30
4.1 Conclusiones	30
4.2 Recomendaciones	30
Referencias	31
Apéndice A	33

Abreviaturas

ACID	Atomicidad, Consistencia, Aislamiento y Durabilidad.
AS-IS	Estado actual de un proceso.
API	Interfaz de programación de aplicaciones.
BD	Base de Datos.
BPMN	Notación de modelado de Procesos de Negocio.
ChatGPT	Chat Generative Pre-Trained Transformer.
DBA	Administrador de base de datos.
ESPOL	Escuela Superior Politécnica del Litoral.
IA	Inteligencia Artificial.
LLM	Modelos de Lenguaje de Gran Escala.
SGBD	Sistema de Gestión de Base de Datos.
TO-BE	Estado futuro deseado de un proceso.

Simbología

E/S Estrada y Salida

Índice de figuras

Fig. 1 Estructura de solución propuesta.....	2
Fig. 2 Simbología básica BPMN – Bizagi Modeler	5
Fig. 3 Modelo AS-IS: Desarrollo de consultas (Inicio del proceso).	12
Fig. 4 Modelo AS-IS: Desarrollo de consultas (Fin del proceso).	13
Fig. 5 Modelo TO-BE: Desarrollo de consultas.....	20
Fig. 6 Modelo TO-BE: SubProceso- Generar Script.....	24

Índice de tablas

Tabla I. Gestores de Bases de Datos.....	3
Tabla II. Indicadores Generales comunes.....	5
Tabla III. Identificando actores.....	8
Tabla IV. Actores y Su Relación con el proceso	8
Tabla V. Objetos de Negocio.....	10
Tabla VI. Matriz de Caso de uso.....	10
Tabla VII. Registro de Excepciones.....	11
Tabla VIII. Matriz Ponderada.....	15
Tabla IX. Identificando actores en proceso optimizado.	16
Tabla X. Actores y Su Relación con el proceso optimizado.	16
Tabla XI. Objetos de Negocio del proceso optimizado.	18
Tabla XII. Matriz de Caso de uso en proceso optimizado.	18
Tabla XIII. Registro de Excepciones en proceso optimizado.....	19
Tabla XIV. Caso de prueba CP01 – Desarrollo de un proceso de actualización de registros por bloques.....	20
Tabla XV. Caso de prueba CP02 – Consulta entre varias tablas primarias para validar información.	21
Tabla XVI. Detalles de los dispositivos usados	22
Tabla XVII. Detalles de Mejora 1	25
Tabla XVIII. Detalles de Mejora 2	25
Tabla XIX. Detalles de Mejora 3	25
Tabla XX. Detalles de Mejora 4	25
Tabla XXI. Detalles de Mejora 5	25
Tabla XXII. Evaluación del Modelo TO-BE	26
Tabla XXIII. Resultados de Desarrollo de un proceso de actualización masiva de registro por bloques.	27
Tabla XXIV Resultados de Consulta entre varias tablas primarias para validar información.....	28

CAPÍTULO 1

1. INTRODUCCIÓN

1.1 Descripción del Problema

Una empresa mediana dedicada al desarrollo de software basado en bases de datos ha identificado la necesidad de optimizar sus procesos para incrementar la eficiencia y fortalecer la autonomía de su equipo técnico. Su enfoque se centra en la creación de soluciones tecnológicas que mejoren la gestión estructurada de la información, permitiendo una adaptación efectiva a las demandas del mercado y a la evolución de las herramientas digitales. Para lograrlo, resulta clave implementar estrategias que faciliten el acceso a conocimientos especializados y optimicen el desarrollo de consultas y la administración de datos.

Actualmente, el proceso de desarrollo de scripts de consulta en MySQL depende del cuestionamiento directo al DBA (Database Administrator) y de documentación técnica cuya claridad y actualización no siempre responden a las necesidades del equipo. Esta situación ha generado tiempos de espera prolongados en el flujo de trabajo y la necesidad de recurrir a fuentes externas, lo que incrementa la complejidad del proceso. Ante esta realidad, se plantea la incorporación de herramientas de apoyo basadas en Inteligencia Artificial, que permitan agilizar la generación de consultas, mejorar el acceso a información estructurada y fortalecer la autonomía del equipo, asegurando mayor eficiencia en el desarrollo de software.

En este contexto, el desarrollo de consultas en MySQL enfrenta problemas críticos que afectan la eficiencia y precisión del proceso. Entre los principales desafíos se encuentra la baja tasa de automatización en tareas repetitivas, lo que obliga a depender de la experticia del DBA en técnicas actualizadas y limita el uso de herramientas tecnológicas avanzadas, como la Inteligencia Artificial. Esta situación contribuye al aumento del riesgo de errores humanos y retrasa la entrega de soluciones. Asimismo, la limitada estandarización de los procesos genera redundancias operativas y dificulta la identificación de puntos críticos, agravando los tiempos prolongados de desarrollo y reduciendo la agilidad necesaria para cumplir con los plazos. Estas limitaciones han afectado negativamente la productividad y la calidad de los proyectos desarrollados, comprometiendo la capacidad de la empresa para responder a las crecientes exigencias del mercado competitivo.

Por lo tanto, se plantea la necesidad de transformar el enfoque actual mediante la integración de herramientas de Inteligencia Artificial generativa, como ChatGPT, en conjunto con el modelado BPMN (Business Process Model and Notation). Estas tecnologías permitirán abordar las causas fundamentales del problema, ofreciendo soluciones para automatizar tareas repetitivas, reducir el margen de error humano, estandarizar los procesos y optimizar los tiempos de desarrollo. En última instancia, esta transformación busca mejorar la precisión y agilidad del desarrollo de consultas en MySQL, garantizando un impacto positivo en la productividad y calidad de los proyectos de software de la empresa. Por lo tanto, el problema central que aborda esta investigación se resume en la siguiente pregunta:

¿Cómo mejorar el tiempo de ejecución de consultas en MySQL mediante el uso de herramientas de Inteligencia Artificial, integradas a través del modelado de procesos con BPMN?

1.2 Justificación del Problema

Para abordar los tiempos excesivos en el desarrollo de consultas MySQL, proponemos una solución híbrida que combina Inteligencia Artificial (IA) con el modelado de procesos de negocio mediante BPMN. Esta estrategia, como se lo puede observar en la **Fig. 1**, se desplegará en tres fases claramente diferenciadas, cada una descrita en el siguiente detalle.

En primer lugar, realizaremos el modelado AS-IS. Mediante un diagrama BPMN, representaremos de forma exhaustiva el flujo actual para la creación y revisión de consultas MySQL. Este mapeo nos permitirá identificar cuellos de botella, pasos manuales redundantes y decisiones que generan retrasos significativos.

A continuación, diseñaremos el modelo TO-BE, incorporando la inteligencia artificial como una herramienta de apoyo en el desarrollo de consultas. La IA analizará y generará los scripts, identificando patrones subóptimos, sugiriendo reescrituras y aplicando optimizaciones estructurales, como la refactorización de joins, la mejora en el uso de índices

y la simplificación de subconsultas. Para ello, se utilizará el API de OpenAI con GPT-4o, integrado en una plataforma conversacional similar a ChatGPT en la empresa y en herramientas especializadas dentro del entorno de desarrollo. Por último, se llevará a cabo la evaluación y el análisis de métricas durante un piloto de una semana con los desarrollos asignados en ese período. Se examinará el tiempo medio de entrega, la cantidad de errores de sintaxis y la percepción de los desarrolladores respecto a las herramientas utilizadas.

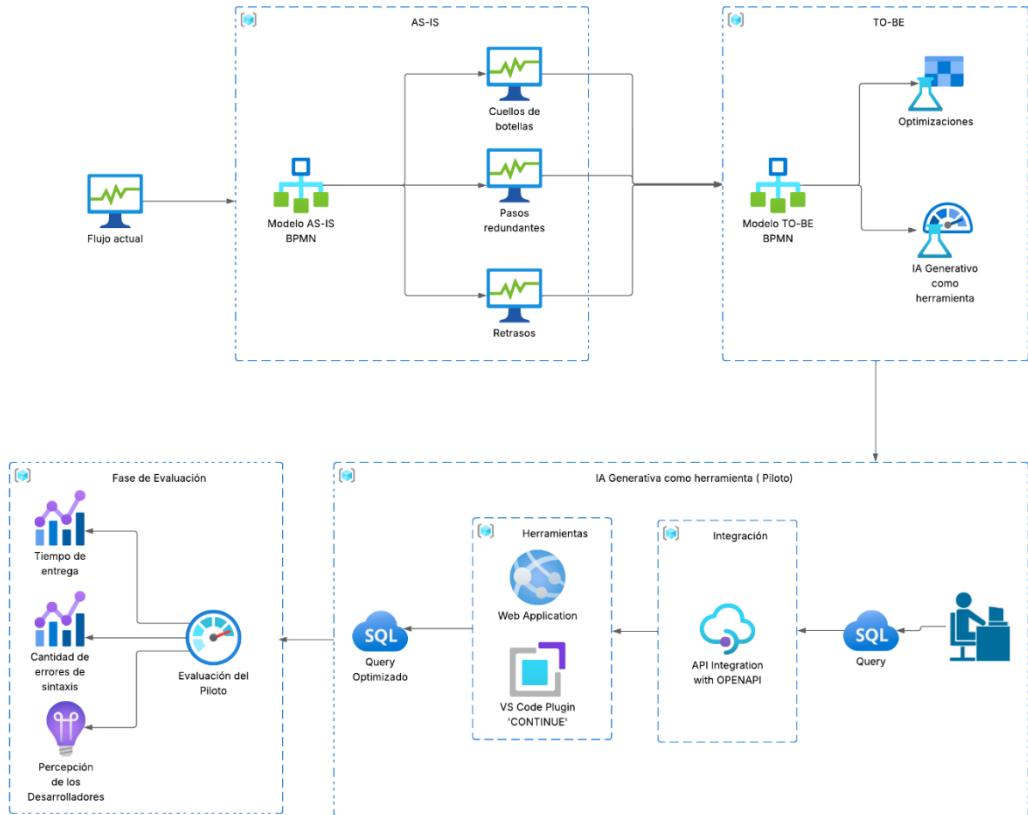


Fig. 1 Estructura de solución propuesta.

Para optimizar el desarrollo, es esencial estructurar consultas eficientes, automatizar validaciones y promover soluciones robustas que eviten retrabajo y deuda técnica. Un proceso estandarizado con IA generativa y BPMN garantiza fiabilidad, seguridad y mantenibilidad. Como advierte [1], las consultas MySQL mal diseñadas generan errores que requieren intervención del DBA, mientras que en [5] la falta de validación automatizada produce código frágil y soluciones provisionales. Además, la documentación obsoleta y ausencia de guías incrementan errores y retrabajos [6]. El modelado AS-IS identifica puntos críticos para automatización, y el TO-BE estandariza validaciones entre desarrolladores, IA y DBA [2]. Asistentes como ChatGPT [3] y DeepSeek [4] reducen carga manual mediante sugerencias, explicaciones y correcciones. Este enfoque, documentado en BPMN interactivo, puede implementarse en 2.5 meses sin afectar operaciones diarias.

El estudio presenta limitaciones significativas. Primero, depende de herramientas externas como OpenAI, donde cambios en APIs, latencia o costos pueden afectar disponibilidad y resultados. Segundo, aunque el modelo genera consultas sintácticamente correctas, pueden ser lógicamente erróneas o ineficientes, requiriendo siempre revisión del DBA antes de ejecutarse. Tercero, compartir esquemas o datos sensibles con la IA implica riesgos de filtraciones, dependiendo críticamente del cifrado punto a punto en las APIs [4]. Además, los hallazgos se basan en solo 10 desarrolladores de una empresa, por lo que las mejoras en tiempo, precisión y percepción podrían variar en organizaciones con distinta cultura de datos. Finalmente, el mayor entusiasmo de desarrolladores junior hacia la IA frente a los seniors podría sesgar percepciones y mediciones de precisión.

Estas limitaciones exigen prácticas éticas y operativas claras. Debemos registrar meticulosamente cada consulta, respuesta y metadatos (fecha, versión del modelo) para garantizar trazabilidad y permitir auditorías rápidas ante fallos en producción [5]. Simultáneamente, es crucial revisar licencias y condiciones de uso, ya que herramientas como ChatGPT pueden generar conflictos de autoría o exponer datos sensibles [5]. La validación humana combinada con auditorías de código es indispensable, pues las recomendaciones basadas en patrones genéricos pueden producir código erróneo o inseguro en contextos específicos [4]. Para proteger información confidencial, priorizamos esquemas anonimizados o simulados, implementando controles de acceso que aseguren un uso ético y responsable.

1.3 Objetivos

1.3.1 Objetivo general

Diseñar un modelo de desarrollo de consultas en MySQL que integre IA generativa mediante el uso de notación de modelado de procesos de negocio (BPMN) para el mejoramiento en la eficiencia en proyectos de software.

1.3.2 Objetivos específicos

1. Diagnosticar el proceso actual de consultas MySQL en un diagrama BPMN AS-IS utilizando Bizagi Modeler.
2. Elaborar un modelo BPMN TO-BE que integre inteligencia artificial generativa para desarrollar consultas MySQL, usando Bizagi Modeler.
3. Validar el modelo TO-BE mediante pruebas controladas con herramientas de IA, evaluando la reducción del tiempo de desarrollo (días), la percepción de los desarrolladores (encuestas de satisfacción) y la precisión del código (errores de sintaxis detectados).

1.4 Marco Teórico

La Inteligencia Artificial Generativa optimiza el desarrollo de software y bases de datos mediante la automatización de consultas MySQL, mejorando la gestión y recuperación de información. Su integración con el Modelado de Procesos BPMN permite estructurar flujos de trabajo eficientes, optimizando la creación de consultas MySQL en el desarrollo de proyectos de software.

1.4.1 Desarrollo de consultas MySQL

Sistemas de Gestión de Bases de Datos son aplicaciones claves para almacenar, organizar y acceder a grandes volúmenes de información durante el ciclo de desarrollo. Como señala [18], optimizan consultas, controlan transacciones y definen esquemas, asegurando eficiencia en operaciones con datos. Entre los gestores destacados podemos ver en la **Tabla I**:

Tabla I. Gestores de Bases de Datos.

Gestor de base de datos	Característica
MySQL	Versátil y eficiente para manejo masivo de datos [14].
InnoDB	Garantiza transacciones ACID (atomicidad, consistencia, aislamiento, durabilidad).
MyISAM	Ideal para procesamiento a gran escala [14].

Podemos **definir y fundamentar** que según [14], MySQL, como RDBMS de código abierto, destacan por su velocidad y capacidad para gestionar información compleja. Además, que por [19], motores como InnoDB se usan para transacciones seguras y MyISAM para consultas rápidas. Su arquitectura distribuye operaciones de E/S entre discos, evitando cuellos de botella, mientras que transacciones ACID e índices mejoran el rendimiento.

Los **Desafíos** según [16] como generar consultas SQL precisas sigue siendo crítico. Herramientas automatizadas manejan bien sentencias simples, pero fallan en requerimientos complejos, dialectos SQL variables o consultas avanzadas. Los métodos manuales aumentan errores de sintaxis y carga cognitiva [18], exigiendo validaciones rigurosas para entornos productivos.

1.4.2 Inteligencia artificial generativa en desarrollo de software

Modelos como **GPT-4** actúan como asistentes en el ciclo de desarrollo. Según [1], aceleran tareas técnicas con indicaciones precisas, reduciendo dependencia de expertise especializado. No obstante, [5] advierte que sin supervisión pueden producir código inválido o vulnerar seguridad, subrayando la necesidad de combinar IA con criterio humano.

Beneficios de usar IA generativa para asistencia en código.

La integración de inteligencia artificial (IA) generativa en el desarrollo de software ha demostrado ventajas significativas, respaldadas por varias investigaciones agrupadas en tres áreas clave:

Incremento de Productividad: La automatización de tareas repetitivas, como la escritura de código básico o la corrección de errores sintácticos, permite a los desarrolladores dedicar más tiempo a problemas complejos. Por ejemplo, el estudio de [1] evidenció una reducción del 70% en el tiempo de desarrollo en proyectos pequeños al emplear ChatGPT para generar especificaciones técnicas, código y casos de prueba. Esto destaca su utilidad para acelerar fases iniciales del ciclo de desarrollo.

Mejora en la Calidad del Código: Las herramientas basadas en IA, como ATHENA [3], ofrecen sugerencias adaptables que promueven buenas prácticas. Este estudio mostró que funciones como la personalización de la granularidad del código que van desde fragmentos hasta funciones completas y las explicaciones integradas como documentación de APIs, que ayudan a mantener coherencia y reducir errores comunes, especialmente en equipos colaborativos.

Apoyo al Aprendizaje: Los modelos generativos no solo resuelven problemas, sino que también funcionan como recursos educativos. Por ejemplo, [1] observó que desarrolladores novatos usaron ChatGPT para explorar patrones de diseño como el Modelo-Vista-Controlador), mientras que [3] destacó que las explicaciones interactivas en ATHENA facilitan la comprensión de lógicas complejas para principiantes.

En bases de datos, según [17], la IA genera consultas SQL sintáctica y semánticamente precisas para casos como detección de ciberataques. La integración de prototipos con algoritmos de IA adapta consultas a necesidades específicas, validándolas y ajustándolas con técnicas avanzadas de aprendizaje automático para mejorar eficiencia.

1.4.3 Modelado de procesos con BPMN

Un **proceso** es un conjunto estructurado de actividades que nos permite lograr un objetivo. **BPMN** es el estándar para representar procesos de negocio de forma clara [10], facilitando su comprensión por técnicos y no técnicos, y reduciendo errores de interpretación. Además, optimiza la ejecución operativa [8].

Según [10] **BPMN** es el estándar global para representar procesos de negocio de forma clara y estructurada. Su principal ventaja, como nos indica [8], radica en definir flujos con alto nivel semántico, facilitando la comprensión tanto para técnicos como no técnicos, lo que minimiza errores de interpretación. En la práctica, modela interacciones entre actores en organizaciones complejas y complementa requisitos funcionales en desarrollo de software, visualizando actividades ejecutables dentro del sistema [11]. Además, optimiza la ejecución operativa mediante estandarización y automatización [8].

En el análisis de procesos de negocio, se utilizan los modelos **AS-IS** y **TO-BE** que son clave identificar oportunidades de mejora continua:

- **AS-IS** retrata el estado actual del proceso, exponiendo cómo se ejecutan las tareas sin modificaciones [11]. Su valor está en identificar cuellos de botella, redundancias y oportunidades de optimización.
- **TO-BE** propone el estado futuro ideal [11], incorporando mejoras como automatización o IA para maximizar eficiencia.

Los **modelos de desarrollo de software** han migrado desde enfoques rígidos (como el de cascada) hacia metodologías ágiles (prototipado rápido) y modulares (basadas en componentes reutilizables) esto como se relata en [9]. BPMN y elsus en componentes que se puede observar en la **Fig. 2.** ha potenciado esta evolución:

- **Identifica problemas:** Detecta cuellos de botella y redundancias en flujos gráficos [13].
- **Estandariza:** Homologa validaciones entre equipos, IA y DBAs [2].

Se adapta a innovaciones: Integra tecnologías como IA generativa para automatización inteligente, reduciendo carga operativa y mejorando precisión en decisiones [15].



Fig. 2 Simbología básica BPMN – Bizagi Modeler.

1.4.4 Revisión de trabajos similares

La **IA generativa revoluciona el desarrollo SQL** automatizando el ciclo de vida del software con herramientas como se muestra en [2] con Langchain. Este enfoque reduce tareas manuales: planificación, codificación y pruebas, optimizando tiempos y precisión mediante LLMs ajustados con repositorios públicos, que generan código estructurado, pruebas automatizadas y documentación técnica. Modelos como GPT-4, DeepSeek y Gemini han demostrado ser aliados valiosos en la industria: aceleran tareas técnicas (generación de código/documentación) y permiten enfocarse en problemas complejos, mejorando eficiencia y productividad [6]. Su impacto se **mide** mediante **indicadores clave** contrastados con estudios existentes en la **Tabla II**:

Tabla II. Indicadores Generales comunes.

Categoría	KPI General	Ejemplos
Eficiencia	Reducción de tiempo de desarrollo.	Se logra optimizar hasta un 30% del tiempo en proyectos industriales. [6]
Adopción tecnológica	% de profesionales que adoptan IA.	El 70% de los profesionales se plantean adoptar el uso de la IA como herramienta. [6]
Calidad del código	Errores de sintaxis	Se ha detectado código generado con funciones sin validación o librerías ficticias. [6]
Riesgo de datos	Vulnerabilidades en datos sensibles.	Modelos externos podrían almacenar información confidencial. [7]

En conjunto, los hallazgos que se pueden observar en la **Tabla II** sugieren que el éxito al usar estas herramientas depende de combinar su velocidad con el juicio humano. Mientras la IA agiliza etapas del ciclo de desarrollo, prácticas como revisiones por pares y pruebas automatizadas son esenciales para garantizar seguridad y calidad.

La **adopción de IA generativa** en la creación de consultas MySQL ofrece ventajas claras, pero no está exenta de riesgos. Por ejemplo, casi el 40 % de las recomendaciones automáticas pueden incluir vulnerabilidades como inyecciones SQL o falta de validación adecuada, tal y como documenta la literatura [7] y [16]. Para mitigar estos problemas, es fundamental diseñar prompts especializados y someter los resultados a revisión experta, lo que refuerza la seguridad sin sacrificar la agilidad tal como coinciden [3] y [7]. Integrar la IA dentro de procesos bien definidos, por ejemplo, en diagramas BPMN estructurados, permite combinar la creatividad de la máquina con el control humano.

La IA enfrenta **desafíos críticos** en tres frentes:

- **Privacidad de datos:** Las implementaciones locales protegen mejor información sensible [18], pero su tasa de éxito en generación SQL es inferior al 50% frente a modelos en la nube [4], exigiendo que el diseño BPMN TO-BE equilibre confidencialidad y eficacia operativa [13].
- **Precisión en generación:** LLMs de código abierto aciertan un 66.65% en sentencias SQL vs. 75% de GPT-4 [4], con errores persistentes de sintaxis/semántica que obligan a medir sistemáticamente la "precisión del código" [17].
- **Riesgos operativos:** Las "alucinaciones" de IA producen consultas inválidas o inseguras, y la falta de estándares aumenta fallos en producción [5], [4]. Estas amenazas se mitigan incorporando pruebas automatizadas en flujos BPMN [13].

CAPÍTULO 2

2. METODOLOGÍA.

En este capítulo se establece el camino que se siguió para entender cómo se generaban actualmente las consultas MySQL en la organización (modelo AS-IS). Se propusieron distintas vías de mejora, se seleccionó la opción más adecuada y se definió con detalle la forma en que se articuló el nuevo proceso optimizado bajo BPMN con apoyo de IA generativa. El objetivo fue ofrecer una visión completa de cada fase, desde la recogida de información hasta la validación del proceso mejorado.

2.1 Enfoque y Diseño Metodológico.

El estudio adoptó un **diseño descriptivo, no experimental y transversal**, que combinó análisis estadístico de tiempos con entrevistas cualitativas. Participaron 10 desarrolladores (5 junior y 5 senior) con al menos tres meses de experiencia en la empresa. Se emplearon:

- **Encuestas en línea** con escala Likert para medir percepción de errores, autonomía y utilidad de la IA.
- **Análisis de registros históricos** de tiempo invertido en cada etapa de creación de consultas.
- **Entrevistas semiestructuradas** para profundizar en dificultades y hábitos de trabajo.

La estrategia de identificación se basó en la comparación de métricas antes y después de la intervención. Se registraron tiempos de desarrollo, errores de sintaxis y percepción de autonomía del equipo, lo que permitió atribuir mejoras observadas al uso de IA generativa y BPMN al controlar las condiciones del proceso. Esto nos permitió desplegar nuestra metodología en las siguientes 4 fases:

1. **Levantamiento y Modelado AS-IS** y diagnóstico de puntos de mejora.
2. **Formulación de alternativas** de solución.
3. **Evaluación y selección** de la opción más viable.
4. **Diseño TO-BE** con casos de uso y pruebas.

2.2 Levantamiento y Modelado ASIS

2.2.1 Descripción general

El flujo actual iniciaba cuando un desarrollador recibía su asignación para desarrollar nueva consulta. Pasaba por las etapas de planificación, donde consultaba documentación, al DBA o de fuentes externas; con la información necesaria generaba el script, luego lo probaba antes de enviarlo a certificación por el DBA y finalmente generaba la documentación correspondiente.

2.2.2 Detalles y casos de uso

Para el modelado AS-IS se registró inicialmente la información referente a los actores implicados en el proceso: los afectados por el proceso, los intervenientes y las partes interesadas, cabe resaltar que con respecto a desarrolladores Jr. y Sr. se los abarca con el rol de “Desarrollador” ya que realizan las mismas actividades dentro del proceso. A continuación, en las siguientes tablas se describen los actores y se establecen las relaciones entre éstos y los procesos en los que intervienen. También se documentaron los objetos de negocio que soportan las actividades y se construyó la matriz de casos de uso donde se indican las actividades, los actores implicados y los objetos de negocio requeridos. Finalmente, se registraron las excepciones identificadas y las acciones a tomar en caso de que ocurran.

Tabla III. Identificando actores.

Nombre	Cargo	Departamento o Área
John Doe	Desarrollador	Equipo de Desarrollo
Carl Doe	Desarrollador	Equipo de Desarrollo
Ann Do	Desarrolladora	Equipo de Desarrollo
Mathew Doe	Desarrollador	Equipo de Desarrollo
Josh Doe	Desarrollador	Equipo de Desarrollo
Paulina Do	Desarrolladora	Equipo de Desarrollo
Martha Do	Desarrolladora	Equipo de Desarrollo
Juan Doe	Desarrollador	Equipo de Desarrollo
Luis Doe	Desarrollador	Equipo de Desarrollo
Carlos Doe	Desarrollador	Equipo de Desarrollo
Jane Doe	DBA	Equipo de Base de Datos

Tabla IV. Actores y Su Relación con el proceso

Actor	Rol que juega	Descripción	Interés en el proceso	Responsabilidades
John Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos	<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Carl Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos	<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Ann Do	Desarrolladora	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos	<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Mathew Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos	<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Josh Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos	<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación.

		consultas MySQL.			<ul style="list-style-type: none"> • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Paulina Doe	Desarrolladora	Persona encargada del desarrollo consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos		<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Martha Doe	Desarrolladora	Persona encargada del desarrollo consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos		<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Juan Doe	Desarrollador	Persona encargada del desarrollo consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos		<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Luis Doe	Desarrollador	Persona encargada del desarrollo consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos		<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Carlos Doe	Desarrollador	Persona encargada del desarrollo consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos		<ul style="list-style-type: none"> • Verificar Requerimientos. • Planificar Tareas. • Verificar Documentación. • Revisar Documentación. • Requerir Sesión de consulta. • Revisar fuentes externas. • Generar Script. • Verificar Script. • Generar Documentación.
Jane Doe	DBA	Experto en base de datos.	Verificar que la consulta este correctamente elaborada y evitar		<ul style="list-style-type: none"> • Verificar Disponibilidad. • Realizar consulta. • Certificar Script.

errores
en
ambientes
productivos

Tabla V. Objetos de Negocio.

Nombre	Tipo	Descripción	Parámetros	Roles Involucrados
Requerimientos	DO	Representa la descripción del script requerido	Fecha Asignada Fecha Entregada Requerimiento	Desarrollador
Script	BO	Representa la consulta elaborada por el desarrollador	Consulta	Desarrollador DBA
Documentación	DO	Representa la entrega del desarrollo ya finalizada.	Fecha de entrega. Resultado de Pruebas	Desarrollador

Tabla VI. Matriz de Caso de uso

ID	Actividad	Tipo	Descripción	Rol	Objetos Negocio	de	Possibles estados finales
A01	Revisar requerimientos	Usuario	El Desarrollador revisa el requerimiento de la consulta asignada a desarrollar.	Desarrollador	Requerimientos		
A02	Planificar tareas	Usuario	EL Desarrollador Desarrollador planifica como desarrollar el script.	Desarrollador			
A03	Verificar Documentación Técnica	Usuario	El desarrollador verifica si existe documentación asociada a su asignación	Desarrollador			<ul style="list-style-type: none"> • No Existe Documentación Técnica. • Existe Documentación Técnica.
A04	Revisar Documentación Técnica	Usuario	El desarrollador revisa si la documentación es útil para su tarea.	Desarrollador			<ul style="list-style-type: none"> • La documentación técnica aplica a la asignación. • La documentación no aplica a la asignación
A05	Requerir sesión de consulta	Usuario	El Desarrollador requiere una consulta al DBA	Desarrollador			
A06	Verificar Disponibilidad	Usuario	El DBA verifica sus asignaciones para ver si tiene disponibilidad.	DBA			<ul style="list-style-type: none"> • El DBA tiene disponibilidad. • El DBA no tiene disponibilidad.

A07	Realizar consulta	Usuario	El DBA realiza la consulta junto al desarrollador para aclarar dudas.	DBA		
A08	Revisar fuentes externas	Usuario	El Desarrollador verifica en fuentes externas como internet para realizar sus desarrollos.	Desarrollador		
A09	Generar Script	Usuario	El Desarrollador arma el script acorde a los requerimientos	Desarrollador	Script	
A10	Probar Script	Usuario	El Desarrollado verifica en pruebas internas el script generado.	Desarrollador	Script	<ul style="list-style-type: none"> • El script generado se ejecuta exitosamente. • El script tiene errores al ser ejecutado. • No se detectaron novedades. • Se detectaron novedades.
A11	Certificar Script	Usuario	El DBA certifica el script generado por el desarrollador.	DBA	Script	
A12	Generar documentación	Usuario	El Desarrollador entrega la documentación asociada a su asignación.	Desarrollador	Documentación	

Tabla VII. Registro de Excepciones.

ID	Excepción	Actividad Afectada	Descripción	Acciones Correctivas	Objeto de Negocio.
E01	El desarrollador al verificar la documentación técnica se da cuenta que no existe documentación asociada.	A03: Verificar Documentación Técnica	El desarrollador no encuentra documentación asociada a su asignación.	En caso de no encontrar documentación, se pide una sesión de consulta al DBA. En caso de si encontrar, este revisa si aplica para su asignación.	
E02	El desarrollador al revisar la documentación asociada se da cuenta no aplica para su asignación.	A04: Revisar Documentación Técnica	El desarrollador encuentra documentación asociada a su asignación, pero esta no aplica.	En caso de que, si aplique, este procede a generar el script. En caso de que no aplique este solicitará una sesión de consulta al DBA.	
E03	El desarrollador al requerir una sesión de consulta al DBA no tiene respuesta del mismo.	A05: Requerir sesión de consulta	El desarrollador no tiene respuesta del DBA al requerir una sesión de consulta	En caso de no tener respuesta del DBA en 15 minutos, el desarrollador procede a realizar consultas a fuentes	

E04	El DBA al revisar sus asignaciones, se da cuenta que no tiene disponibilidad para realizar sesiones de consultas.	A06: Verificar Disponibilidad	El DBA no tiene disponibilidad para realizar sesiones de consultas.	externas como lo es internet.
E05	El desarrollador al probar su script generado se encuentra que contiene errores.	A10: Probar Script	El Script generado no se ejecutó correctamente	En caso de que ejecuciones fallidas, el desarrollador debe volver a generar el script. En caso que no, se puede mandar a certificar.
E06	El DBA al certificar el script del desarrollador se encuentra novedades.	A11: Certificar Script	El Script tiene novedades al ser certificado por el DBA	En caso de que existan novedades durante la certificación del script, este debe ser entregado al desarrollador para que vuelva a generarlo

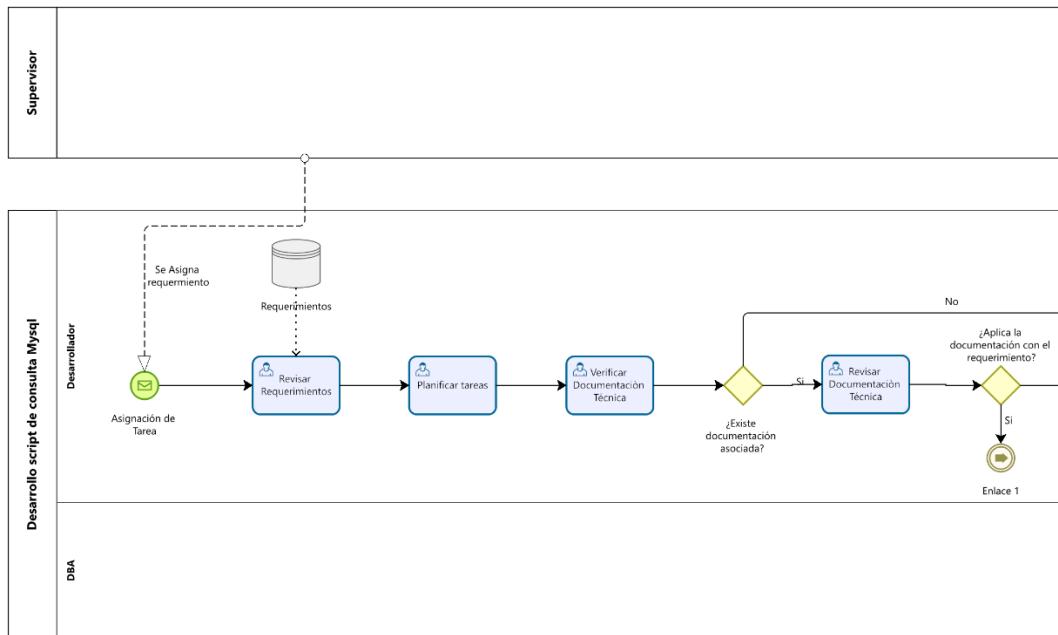


Fig. 3 Modelo AS-IS: Desarrollo de consultas (Inicio del proceso).

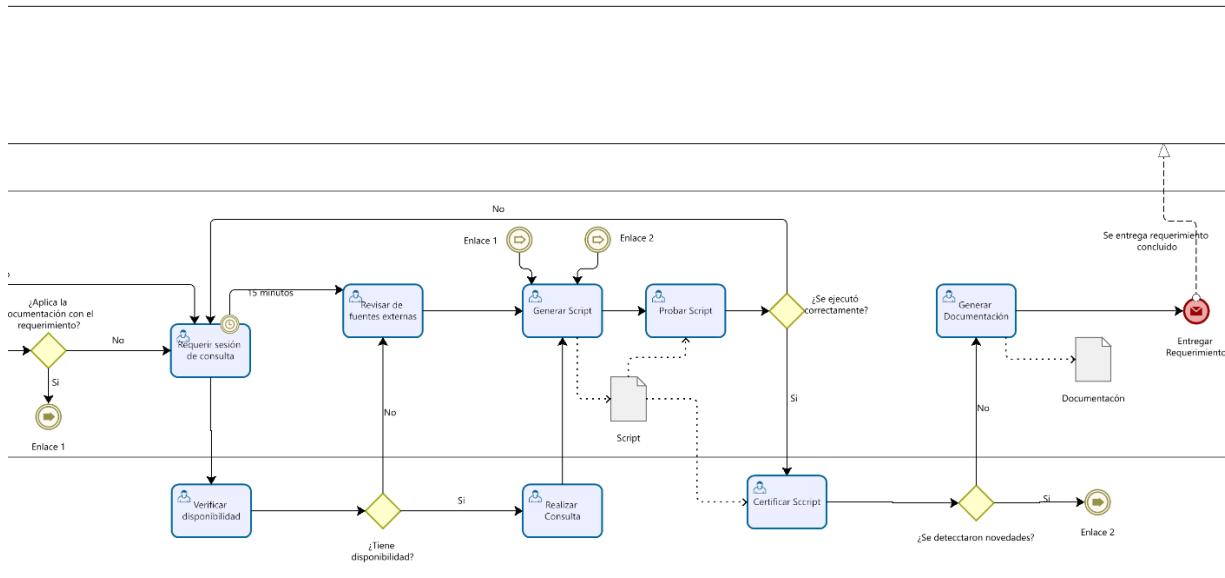


Fig. 4 Modelo AS-IS: Desarrollo de consultas (Fin del proceso).

Del flujo actual de generación de consultas MySQL se detectaron varios cuellos de botella:

- **Sesiones de consulta con el DBA** cuando no había documentación o no era clara, lo que representaba una limitación crítica de autonomía para los desarrolladores. En el 70% de los casos, los desarrolladores júniores recurrían a fuentes externas no verificadas debido a esta dependencia, generando esperas promedio de 1-2 días para consultas complejas y aumentando el riesgo de errores en producción.
- **Espera de 15 minutos para consultar fuentes externas** cuando las solicitudes al DBA no tenían respuesta inmediata. Esto causaba retrasos acumulativos en entregas, con 40% de los desarrolladores reportando que estas esperas extendían los tiempos de desarrollo en más de 30%. La dependencia de fuentes no estructuradas introducía errores en 3-5 consultas por cada 10 desarrolladas.
- **Documentación desactualizada o incompleta**, que obligaba al equipo a repetir validaciones o buscar alternativas en 60% de los casos analizados. Esto generaba reprocesos en 2 de cada 3 proyectos, con desarrolladores invirtiendo hasta 40% de su tiempo en tareas de verificación manual en lugar de codificación.
- **Iteraciones sin control en pruebas**, donde cada fallo en los scripts requirió reiniciar el proceso de consulta. En promedio, se detectaron 3-5 errores de sintaxis por consulta compleja, y en escenarios críticos (30% de los casos) estos errores superaron los 5 por script, extendiendo los tiempos de corrección en más de 80% respecto al desarrollo inicial.

Además, se clasificaron (centrándose en desarrollos de backend más complejos más no consultas simples) en dos grandes grupos:

- **Consultas transaccionales críticas (inserciones/actualizaciones masivas en tablas masivas)** cuyos fallos podían llegar a degradar servicios en producción.
- **Consultas analíticas complejas**, con múltiples joins, subconsultas y participación de varias tablas que podían llegar a consumir muchos recursos y podían colapsar la base si no se optimizaban.

Principales retos

El levantamiento del proceso AS-IS permitió evidenciar desafíos significativos que afectaban la eficiencia, calidad del proceso y limitaciones estructurales/operativas. Entre ellos destacaban:

- **Dependencia del DBA o de fuentes externas:** la obtención de información dependía excesivamente del DBA, lo que era causa raíz de tiempos prolongados en el desarrollo, especialmente sin acceso inmediato a estos recursos.

- **Reprocesos constantes durante la validación del script:** Se evidenciaron múltiples iteraciones para corregir errores en los scripts durante su validación, lo que implicaba tiempos extendidos y menor eficiencia en la entrega de resultados funcionales.
- **Documentación dispersa y desactualizada:** Se reconoció que la falta de documentación técnica consolidada complicaba la comprensión del entorno por parte de los desarrolladores, limitaba su autonomía y generaba errores por falta de contexto confiable.

2.3 Formulación de Alternativas de Solución

Durante el diagnóstico del proceso de desarrollo de consultas MySQL, identificamos varias opciones para abordar las limitaciones encontradas. Las alternativas consideradas fueron:

1. **Plan de capacitación interna sobre la correcta redacción de MySQL y el uso avanzado de base de datos:**
Propusimos un programa de formación técnica enfocado en escritura correcta de SQL y el dominio avanzado de bases relacionales. Buscábamos reducir la dependencia del DBA, ganar autonomía técnica y aplicar buenas prácticas. El plan incluía módulos temáticos prácticos, adaptados a distintos niveles, con ejercicios y resolución de problemas.
2. **Mejora de la documentación técnica con versionado activo:** Planteamos revisar y modernizar los manuales de soporte, buscando estandarizar formatos, usar lenguaje claro e implementar versionado que mantuviera la documentación actualizada. El objetivo era contar con guías confiables que redujeran errores y acortaran los tiempos de consulta.
3. **Automatización de tareas repetitivas mediante generadores de código SQL y asistentes contextuales:**
Pretendimos optimizar tareas mecánicas con generadores de código SQL y asistentes (plantillas/autocomplete), buscando permitir a los desarrolladores enfocarse en problemas complejos, reducir errores por omisión y agilizar la creación de scripts estándar para mejorar la eficiencia operativa.
4. **Incorporación de Inteligencia Artificial generativa, como ChatGPT, dentro de un rediseño estructurado del flujo de trabajo mediante BPMN:** Propusimos integrar herramientas como ChatGPT dentro de un rediseño estructurado usando modelado BPMN. La IA generativa actuaría como asistente técnico en tiempo real para redactar/validar consultas, mientras que el BPMN ayudaría a identificar cuellos de botella y estandarizar tareas. Juntas, permitirían automatizar actividades repetitivas, reducir errores humanos, aumentar la productividad y autonomía.

2.4 Matriz de Decisión y Selección.

2.4.1 Criterios de evaluación

Cada alternativa se valoró según cinco ejes que ponderamos para reflejar su importancia estratégica, lo que nos permitió definir los siguientes criterios de evaluación:

- **Inversión requerida (20 %):** coste monetario y recursos humanos
- **Tiempo de implementación (15 %):** plazo hasta disponer de la solución operativa.
- **Impacto en eficiencia (25 %):** reducción de tiempos de desarrollo y errores.
- **Adaptabilidad del equipo (20 %):** facilidad de adopción y curva de aprendizaje.
- **Escalabilidad futura (20 %):** capacidad de extender la solución a otros proyectos.

2.4.2 Matriz de decisión

Para seleccionar la alternativa de mejora, se analizaron los beneficios y limitaciones de cada opción usando la matriz de decisión ponderada indicada en la **Tabla VIII**.

Tabla VIII. Matriz Ponderada

Criterio	Peso	Capacitación	Doc. Técnica	Automatización	IA Generativa
Inversión Requerida.	0.20	4	5	3	4
Tiempo de implementación.	0.15	3	4	4	5
Impacto en eficiencia.	0.25	2	3	4	5
Adaptabilidad del equipo.	0.20	4	4	3	5
Escalabilidad futura.	0.20	3	4	4	5
Puntuación Total.	1.00	3.1	4.0	3.7	4.8

La alternativa 4 (uso de IA generativa como herramienta) obtuvo la mayor puntuación, con una puntuación ponderada de 4.8, debido a su alto impacto en eficiencia y adaptabilidad, además de requerir una inversión y plazo razonable.

2.5 Análisis Costos de la alternativa seleccionada

Como parte del análisis, se recopiló información sobre el costo de implementación de la mejora, obteniendo los siguientes datos:

Herramientas existentes

- ChatGPT disponible en licencias corporativas.
- Bizagi Modeler en versión gratuita.
- Dbeaver Community como gestor de base gratuito.

Capacitación inicial

- Media jornada de introducción para familiarizar al equipo con el asistente de IA.

Coste estimado

- Dada que la infraestructura y licencias ya estaban en uso, el único gasto detectable fue el tiempo de formación, que se consideró mínimo.

2.6 Justificación de la selección

La elección de integrar IA generativa con el rediseño estructurado del proceso se justificó por múltiples evidencias recolectadas durante el análisis:

- Se documentó demoras reiteradas por falta de disponibilidad del DBA y documentación técnica poco clara.
- Se observó una baja estandarización de prácticas entre desarrolladores, lo que generaba redundancias y retrabajos.

- Las entrevistas mostraron que el 70% de los desarrolladores junior consultaban fuentes externas no verificadas, lo que aumentaba los riesgos de errores y seguridad.
- Se identificó que tareas repetitivas consumían hasta un 40% del tiempo total destinado a la construcción de scripts SQL.

Con base en estos, se priorizó una alternativa que no solo buscara resolver cuellos de botella, sino que empoderara al equipo técnico, redujera la dependencia de actores individuales y preparara a la organización para adoptar nuevas tecnologías de forma sostenible.

Además, se eligió una herramienta de bajo costo de implementación (ChatGPT) y una metodología de modelado reconocida (BPMN) que permitieran escalar el rediseño a otras áreas con mínima inversión.

2.7 Modelado TO-BE

Se utilizó BPMN para redefinir el flujo, se removieron tareas de consultas a nivel de documentación y se incorporó la IA generativa como herramienta en los nodos críticos del proceso. A continuación, se describieron los actores los cuales se mantienen la generalización para desarrolladores indicada en el modelado AS-IS y se relacionaron con sus procesos correspondiente. Luego se procedió a documentar los objetos de negocio que soportan los procesos. Además, se desarrolló la Matriz de casos de usos donde se indicaron las actividades, actores participantes y los objetos de negocio requeridos. Por último, se registraron las excepciones con sus respectivas acciones a tomar en las siguientes tablas.

Tabla IX. Identificando actores en proceso optimizado.

Nombre	Cargo	Departamento o Área
John Doe	Desarrollador	Equipo de Desarrollo
Carl Doe	Desarrollador	Equipo de Desarrollo
Ann Do	Desarrolladora	Equipo de Desarrollo
Mathew Doe	Desarrollador	Equipo de Desarrollo
Josh Doe	Desarrollador	Equipo de Desarrollo
Paulina Do	Desarrolladora	Equipo de Desarrollo
Martha Do	Desarrolladora	Equipo de Desarrollo
Juan Doe	Desarrollador	Equipo de Desarrollo
Luis Doe	Desarrollador	Equipo de Desarrollo
Carlos Doe	Desarrollador	Equipo de Desarrollo
Jane Doe	DBA	Equipo de Base de Datos
ChatGPT	IA Generativa	Herramienta para Desarrollo

Tabla X. Actores y Su Relación con el proceso optimizado.

Actor	Rol que juega	Descripción	Interés en el proceso	Responsabilidades
John Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos.	<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación.
Carl Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL de en base a los requerimientos.	<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual

Ann Do	Desarrolladora	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.		<ul style="list-style-type: none"> • Validar Script • Verificar Script. • Generar Documentación. 	
Mathew Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.		<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación. 	
Josh Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.		<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación. 	
Paulina Do	Desarrolladora	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.		<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación. 	
Martha Do	Desarrolladora	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.		<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación. 	
Juan Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.		<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual 	

				<ul style="list-style-type: none"> • Validar Script • Verificar Script. • Generar Documentación.
Luis Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.	<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación.
Carlos Doe	Desarrollador	Persona encargada del desarrollo de consultas MySQL.	Entregar la consulta MySQL en base a los requerimientos.	<ul style="list-style-type: none"> • Revisar Requerimientos. • Planificar Tareas. • Revisar Documentación. • Generar Script. • Definir Requisitos de script • Crear Script manual • Validar Script • Verificar Script. • Generar Documentación.
ChatGPT	IA Generativa	Herramienta para el desarrollo	Generar y optimizar scripts.	<ul style="list-style-type: none"> • Validar requisitos. • Sugerir script base.
Jane Doe	DBA	Experto en base de datos.	Verificar que la consulta este correctamente elaborado y evitar errores en ambientes productivos.	<ul style="list-style-type: none"> • Certificar Script.

Tabla XI. Objetos de Negocio del proceso optimizado.

Nombre	Tipo	Descripción	Parámetros	Roles Involucrados
Requerimientos	DO	Representa la descripción del script requerido	Fecha Asignada Fecha Entregada Requerimiento	Desarrollador
Indicaciones	BO	Representa la entrada o la petición a la IA Generativa	Requerimientos del Script	Desarrollador IA Generativa
Script	BO	Representa la consulta elaborada por el desarrollador	Consulta	Desarrollador DBA
Documentación	DO	Representa la entrega del desarrollo ya finalizada.	Fecha de entrega. Resultado de Pruebas	Desarrollador

Tabla XII. Matriz de Caso de uso en proceso optimizado.

ID	Actividad	Tipo	Descripción	Rol	Objetos de Negocio	Posibles estados finales
A01	Revisar requerimientos	Usuario	El Desarrollador revisa el requerimiento de la consulta asignada a desarrollar.	Desarrollador	Requerimientos	

A02	Planificar tareas	Usuario	El Desarrollador planifica como desarrollar el script.	Desarrollador		
A03	Revisar Documentación Técnica	Usuario	El desarrollador revisa si la documentación es útil para su tarea.	Desarrollador		
A04	Generar Script	Usuario	El Desarrollador arma el script acorde a los requerimientos.	Desarrollador	Indicaciones Script	<ul style="list-style-type: none"> • El Script Generado se ejecuta correctamente. • El Script generado no se ejecuta correctamente.
ASP01	Definir requisitos de script	Usuario	El Desarrollador define los requisitos o entrada necesaria para generar el script.	Desarrollador	Indicaciones	<ul style="list-style-type: none"> • Se requiere usar la IA como herramienta. • No se Requiere usar la IA como herramienta.
ASP02	Crear Manual	Usuario	El desarrollador crea el script sin usar la IA como herramienta.	Desarrollador		
ASP03	Validar Requisitos.	Automático	La IA Generativa valida la entrada facilitada por el desarrollador.	IA Generativa		
ASP04	Sugerir Script Base.	Automático	La IA Generativa sugiere un script base al desarrollador en base a las indicaciones facilitadas.	IA Generativa		
ASP05	Validar Script	Usuario	El desarrollador valida si el script cumple los requerimientos.	Desarrollador		<p>El Script cumple con los requisitos. El Script no cumple con los requisitos.</p>
A05	Certificar Script	Usuario	El DBA certifica el script generado por el desarrollador.	DBA	Script	<ul style="list-style-type: none"> • No se detectaron novedades. • Se detectaron novedades.
A06	Generar documentación	Usuario	El Desarrollador entrega la documentación asociada a su asignación.	Desarrollador	Documentación	

Tabla XIII. Registro de Excepciones en proceso optimizado.

ID	Excepción	Actividad Afectada	Descripción	Acciones Correctivas	Objeto de Negocio.
E01	El desarrollador al definir los requerimientos de script se da cuenta	ASP01: Definir de requisitos de script	El desarrollador encuentra que para armar el script no se requiere usar la IA generativa.	En caso de no requerir usar la IA Generativa, el desarrollador crea	Indicaciones

	que no se requiere usar la IA		el script de manera manual.
E02	El desarrollador al validar el script, se da cuenta que no cumple con los requerimientos.	ASP05: Validar Script	El desarrollador valida si el script generado cumple los requerimientos. En caso de que no cumpla los requerimientos, se vuelven a definir los requisitos para volver armar el script.
E03	El desarrollador al probar su script generado se encuentra que contiene errores.	A04: Generar Script	El Script generado no se ejecutó correctamente. En caso de que ejecuciones fallidas, el desarrollador debe volver a generar el script. En caso que no, se puede mandar a certificar.
E04	El DBA al certificar el script del desarrollador se encuentra novedades.	A11: Certificar Script	El Script tiene novedades al ser certificado por el DBA. En caso de que existan novedades durante la certificación del script, este debe ser entregado al desarrollador para que vuelva a generarlo

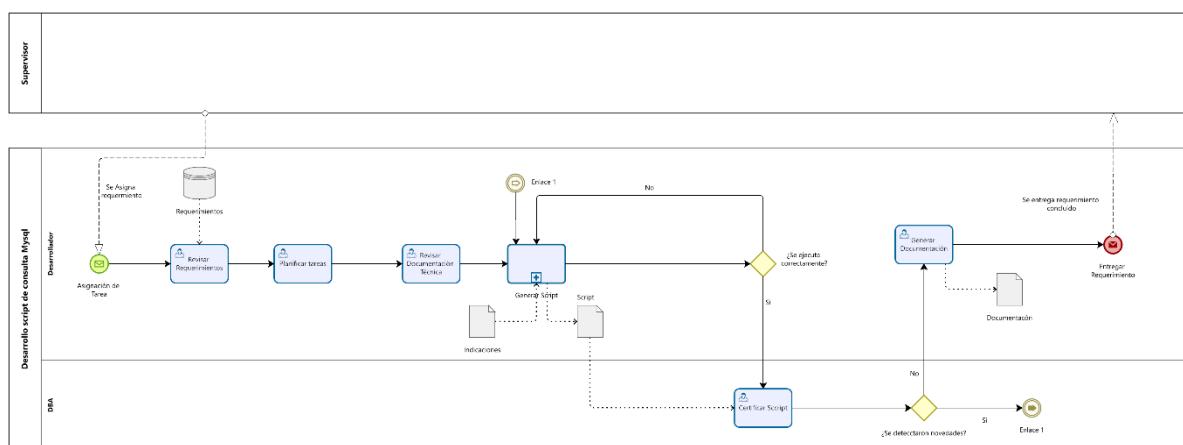


Fig. 5 Modelo TO-BE: Desarrollo de consultas.

Para verificar las mejoradas propuestas, se simuló el uso de la IA generativa proporcionada por la empresa en dos casos de pruebas asociadas a los grupos indicados con anterioridad, siendo las siguientes:

2.7.1 Desarrollo de un proceso de actualización masiva de registro por bloques.

Prueba centrada en desarrollar un script donde se definió un store procedure que permitiera actualizaciones masivas por bloques en una tabla primaria de clientes de alrededor de 10K tomando como pivote un rango de fechas, el caso se detalla en la **Tabla XIV**.

Tabla XIV. Caso de prueba CP01 – Desarrollo de un proceso de actualización de registros por bloques.

Identificador:	CP01
Título:	Actualización masiva de registros de clientes por bloques usando un rango de fechas.
Relacionado con:	ASP03, ASP04 y ASP05

Precondición:	<ul style="list-style-type: none"> La tabla clientes existe en la base de datos con al menos 10,000 registros. La tabla contiene las columnas: id (PK), fecha_modificacion (datetime), y estado (varchar). Existe un índice no agrupado en fecha_modificacion.
Pasos:	
1. Paso 1:	Desarrollador define los requerimientos para el script.
2. Paso 2:	La IA valida la información de entrada.
3. Paso 3:	La IA crea un procedimiento almacenado que actualice el campo estado en bloques, usando un rango de fechas como filtro.
4. Paso 4:	El desarrollador verifica que el script esté listo para ejecutarse.
5. Paso 5:	El desarrollador ejecuta el script.
6. Paso 6	El desarrollador verifica que los registros se actualizaron correctamente
Resultado Esperado:	
Efectividad:	<ul style="list-style-type: none"> Todos los registros con fecha_modificacion en el rango 2023-01-01 a 2023-12-31 actualizan su estado a INACTIVO. Registros fuera del rango de fechas no son modificados.
Rendimiento:	<ul style="list-style-type: none"> La actualización se realiza en bloques de máx. 1,000 registros por iteración. No se generan bloqueos de tabla completa (solo bloqueos por lote). El tiempo de ejecución total es inferior a 60 segundos para 10K registros.
Consistencia:	<ul style="list-style-type: none"> El campo fecha_modificacion se actualiza a la fecha/hora de la operación. El procedimiento evita reprocesar registros ya en estado INACTIVO.

2.7.2 Consulta entre varias tablas primarias para validar información.

Prueba centrada en desarrollar una consulta que permitiera recuperar de tres tablas (clientes, productos y su tabla de relación) para validar los datos de un cliente y sus productos. Como se detalla en la **Tabla XV**.

Tabla XV. Caso de prueba CP02 – Consulta entre varias tablas primarias para validar información.

Identificador:	CP02
Título:	Validación de información de cliente y sus productos activos mediante consulta en tres tablas relacionadas.
Relacionado con:	ASP03, ASP04 y ASP05
Precondición:	<p>Existencia de tres tablas relacionadas:</p> <ul style="list-style-type: none"> Tabla 1 (información de clientes). Tabla 2 (catálogo de productos). Tabla3 (tabla de relación).
Datos de prueba consistentes con:	<ul style="list-style-type: none"> Clientes. Productos. Asignaciones a productos activos
Pasos:	
1. Paso 1:	Desarrollador define los requerimientos para la consulta.
2. Paso 2:	La IA valida la información de entrada.
3. Paso 3:	La IA Crear una consulta SQL que recupere cliente y sus productos activos.
4. Paso 4:	El desarrollador verifica que el script esté listo para ejecutarse.
5. Paso 5:	El desarrollador ejecuta la consulta con diferentes casos de prueba
6. Paso 6	El desarrollador verifica que la consulta recupere la información correctamente
Resultado Esperado:	

Escenario: Cliente activo con productos activos.
 Resultado: Retorna todos los productos activos del cliente.

Escenario: Cliente inactivo.
 Resultado: 0 registros retornados.

Escenario: Cliente sin productos.
 Resultado: 0 registros retornados

Escenario: Producto inactivo en catálogo.
 Resultado: Excluido de resultados.

Escenario: Relación cliente-producto expirada.
 Resultado: Excluida de resultados.

Escenario: Cliente inexistente.
 Resultado: 0 registros retornados.

Además de ejecutar técnicamente los dos casos de prueba, se recogió retroalimentación con los desarrolladores sobre la claridad del flujo TO-BE y la utilidad de las sugerencias generadas por la inteligencia artificial. Esta retroalimentación permitió validar la aceptabilidad práctica de la solución, y se destacó que la integración de la IA en el modelo TO-BE no solo optimizó la generación y corrección de consultas SQL en MySQL, sino que también facilitó una mayor comprensión de los requerimientos, redujo tiempos de retrabajo y fortaleció la calidad de los resultados. La colaboración humano-tecnología se consolidó como elemento clave para mejorar la eficiencia y precisión en el desarrollo de soluciones técnicas.

2.8 Consideraciones Éticas y Legales

En el desarrollo y ejecución de los escenarios CP01 y CP02 se respetaron estrictamente las normativas éticas y legales de tratamiento de información confidencial y datos sensibles. Toda la información fue anonimizada para proteger la identidad de clientes y garantizar el cumplimiento de principios de privacidad y seguridad, acorde a normativas como la Ley de Protección de Datos Personales. El uso de las herramientas proporcionadas para la simulación fue autorizado explícitamente por la empresa, asegurando un entorno controlado y seguro. Además, se trabajó con los desarrolladores bajo políticas corporativas que promovían seguridad, confidencialidad, disponibilidad e integridad de la información. Este enfoque riguroso garantizó que las pruebas se ejecutaran de manera responsable, sin comprometer datos ni infringir obligaciones éticas o legales.

2.9 Especificaciones Técnicas para el Uso de las Herramientas

Para garantizar la viabilidad operativa del modelo TO-BE durante las pruebas, se definió especificaciones técnicas detalladas que se observa en la **Tabla XVI** que cubrieran tanto los requisitos de hardware como las configuraciones de software necesarias. Estas condiciones técnicas, alineadas con las estaciones de trabajo existentes en la empresa, aseguraron un entorno estable para la ejecución de los casos de prueba con IA generativa y herramientas de modelado BPMN. La compatibilidad verificada entre los dispositivos utilizados, las licencias corporativas disponibles (como ChatGPT y Bizagi Modeler), y los protocolos de seguridad implementados, permitió ejecutar la fase de validación metodológica sin incidencias críticas, consolidando así la base técnica para el análisis de resultados presentado en el siguiente capítulo.

Tabla XVI. Detalles de los dispositivos usados

Componente	Especificaciones Recomendadas
Sistema Operativo	Windows 10/11 (64 bit)
Procesador (CPU)	Intel Core i5 / i7 o AMD Ryzen 5 / 7
Memoria RAM	16 GB DDR4 o superior
Almacenamiento	SSD de 256 GB o más

Pantalla	Resolución Full HD (1920 × 1080) o superior
Conectividad	Acceso a Internet de banda ancha estable, sin restricciones en el puerto HTTPS (443)
Software requerido	<ul style="list-style-type: none">• .NET Framework 4.6.1+• Java 11+ (incluido con DBeaver)• Node.js / Python (según lenguaje usado para API)• Office 2010+ (para Bizagi)• Visual Studio Code actualizado
Navegador Web	Última versión de Chrome, Edge o Firefox para vista previa de resultados y documentación online
Seguridad y permisos	Acceso sin restricciones de firewall/proxy a los dominios de OpenAI, Bizagi, y repositorios de código (GitHub, etc.)

CAPITULO 3

3. ANALISIS DE RESULTADOS.

En este capítulo analizamos los resultados luego de implementar la solución propuesta: integrar inteligencia artificial dentro del proceso rediseñado con BPMN para crear consultas MySQL. Como se puede apreciar en el capítulo anterior, se implementó este nuevo enfoque con 10 desarrolladores de la empresa (5 junior y 5 senior), además de simular casos de pruebas prácticos como actualizaciones masivas de datos y consultas complejas entre tablas. Los resultados combinan la medición en tiempos y errores con la percepción del equipo establecidas en encuestas y entrevistas.

El rediseño con BPMN que se observa en la figura 5 permitió ordenar el flujo de trabajo, identificar nudos críticos y estandarizar tareas. Al sumar la IA generativa como asistente, se logra que, tareas repetitivas como validar requisitos o generar borradores de código se hicieran más rápido y con menos errores. Esto liberó a los desarrolladores para enfocarse en problemas complejos y redujo la dependencia del DBA, algo que antes retrasaba los proyectos.

Lo más valioso fue ver cómo esta combinación de procesos claros con la IA generativa no solo aceleró el desarrollo, sino que mejoró la autonomía del equipo. Los desarrolladores ganaron confianza al tener un "compañero virtual" que sugería soluciones, explicaba opciones y detectaba fallos temprano. A continuación, se establece si estos cambios realmente cumplieron con lo esperado: menos tiempo, menos errores y un equipo más ágil.

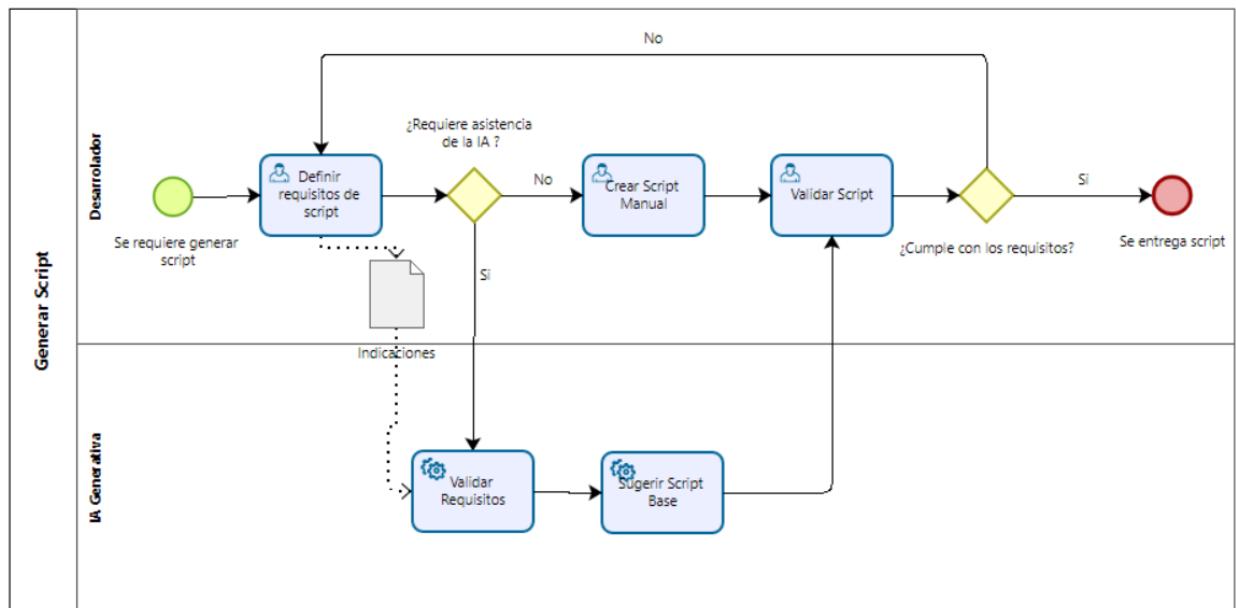


Fig. 6 Modelo TO-BE: SubProceso- Generar Script.

3.1 Análisis de Rediseño del Proceso (Modelado TO-BE)

Tras la revisión del proceso del modelo AS-IS y alineado con el segundo objetivo que busca el integrar inteligencia artificial generativa se consideraron los siguientes aspectos de mejora al proceso.

Mejora 1: Incorporación de IA Generativa como actor

Descripción: Se añade la IA generativa como herramienta activa para asistir en la generación y optimización de scripts.

Tabla XVII. Detalles de Mejora 1

Antes	Ahora
Sólo participan el Desarrollador y el DBA.	Se integra un tercer actor, “IA Generativa”, con responsabilidades definidas (sugerir script base, validar requisitos).
Todas las actividades de generación y validación de scripts son manuales.	Parte de actividades (validación de insumos y sugerencia de script) se automatizan vía IA.

Mejora 2: Definición de nuevos Objetos de Negocio (“Indicaciones”)**Descripción:** Se crea un objeto de negocio intermedio para canalizar la comunicación de requisitos al motor de IA.

Tabla XVIII. Detalles de Mejora 2

Antes	Ahora
Sólo existían “Requerimientos”, “Script” y “Documentación” como objetos.	Se añade el objeto “Indicaciones” (entrada para IA), componente clave en el flujo TO-BE.
El flujo de información al DBA era directo desde el Desarrollador.	El Desarrollador primero define sus “Indicaciones” para la IA, luego ajusta según la salida generada.

Mejora 3: Automatización de validación de requisitos**Descripción:** La IA comprueba automáticamente que los insumos cumplan el formato y contenido necesarios antes de generar el script.

Tabla XIX. Detalles de Mejora 3

Antes	Ahora
La validación de requerimientos la hace manualmente el Desarrollador revisando la documentación.	Existe la actividad automática “Validar Requisitos” (ASP03), realizada por IA Generativa.
Errores de formato o falta de detalles se detectaban sólo al probar el script.	Se detectan y corrigen en la fase de “Definir requisitos” antes de generación.

Mejora 4: Sugerencia de script base por IA**Descripción:** La IA provee un borrador de script, reduciendo el tiempo de codificación manual.

Tabla XX. Detalles de Mejora 4

Antes	Ahora
El Desarrollador escribe el script completamente desde cero.	Se introduce la actividad “Sugerir Script Base” (ASP04) ejecutada automáticamente por la IA.
La generación inicial era 100 % manual, con mayor probabilidad de errores.	El borrador IA acelera la creación y sirve de guía, disminuyendo iteraciones.

Mejora 5: Simplificación de la Matriz de Casos de Uso**Descripción:** Se unifican y renumeran actividades, eliminando pasos redundantes y agrupando excepciones relacionadas con IA.

Tabla XXI. Detalles de Mejora 5

Antes	Ahora
Doce actividades (A01–A12) con varios chequeos manuales y consultas al DBA.	Cinco actividades principales (A01–A06) y cinco sub-pasos ASP01–ASP05, con roles claros de usuario/automático.
Cada excepción referenciaba sólo al DBA o al Desarrollador.	Se contemplan excepciones tanto para flujos manuales (E01–E02) como para flujos IA (E01–E02 adaptadas para ASP).

Estas mejoras reflejan una transición hacia un modelo más ágil, colaborativo y soportado en IA, que reduce la carga manual del Desarrollador y optimiza los tiempos de entrega y calidad de los scripts.

3.2 Análisis de limitaciones

Durante las simulaciones y pruebas, se identificaron varias limitaciones de la IA generativa al desarrollar consultas complejas en SQL:

- **Comprensión contextual insuficiente:** Dificultad para interpretar plenamente los requerimientos operacionales o el contexto semántico necesario para el script.
- **Errores de lógica y precisión:** Los scripts generados podían presentar fallos en su razonamiento o producir resultados inexactos.
- **Riesgos de seguridad y confidencialidad:** El uso de estas herramientas plantea riesgos potenciales, ya que las consultas generadas podrían exponer información sensible si no se implementan controles de seguridad robustos, comprometiendo la privacidad de los datos.
- **Ilusiones (alucinaciones) de la IA generativa:** Las IAs pueden generar consultas o afirmaciones factualmente incorrectas o inventadas como puede ser el caso de sentencias SQL que parecen correctas pero que no respetan el esquema de la base de datos o los requisitos operacionales. Esto puede ocasionar resultados erróneos, fallos en pruebas o riesgos de datos en entornos productivos.

Para mitigar estas limitaciones, se adoptó un enfoque híbrido que permitió combinar la validación humana con ajustes específicos para el uso de la herramienta tal como:

- **Mejorar la comprensión contextual:** Indicar datos relevantes y documentación precisa del entorno operacional para facilitar la interpretación semántica.
- **Reforzar la precisión:** Implementar rutinas de verificación lógica y pruebas automatizadas para detectar y corregir errores en los scripts generados.
- **Garantizar la seguridad:** Verificar que las medidas como filtros de anonimización, cifrado de datos y políticas estrictas de control de acceso estén debidamente aplicadas. Además, es crucial ejecutar las consultas en entornos seguros que cumplan con los estándares regulatorios aplicables.
- **Mitigar alucinaciones:** anclar las respuestas a fuentes internas (documentación de la empresa), validar automáticamente las consultas y requerir revisión DBA antes de ejecutar en producción.

3.3 Análisis de Encuestas a Desarrolladores

Con base en los resultados obtenidos en la encuesta realizada a 10 desarrolladores que se indica como parte del **Apéndice A**, se pudo realizar un análisis comparativo de los tres aspectos clave definidos en el objetivo específico: tiempo de desarrollo, percepción de los desarrolladores y precisión del código. A continuación, se presenta una descripción del análisis de los resultados que se pueden observar en la **Tabla XXIII**.

Tiempo de Desarrollo (Reducción en días): Antes del uso de herramientas de IA, una proporción significativa de desarrolladores indicó que el desarrollo de consultas complejas tomaba entre 1 a más de 2 días. Con la incorporación en el flujo del uso de herramientas de IA, la mayoría reportó una disminución a menos de un día. Esto evidencia una clara reducción del tiempo de desarrollo, cumpliendo uno de los principales indicadores del objetivo.

Percepción de los Desarrolladores (Encuestas de satisfacción): La percepción sobre la utilidad de la IA fue mayoritariamente positiva, con puntuaciones altas (valores 4 y 5 en una escala del 1 al 5). Además, una amplia mayoría indicó sentirse más autónomos al usar IA, y más confiados en comparación con el desarrollo tradicional. Estos datos validan que el modelo TO-BE con IA mejora la experiencia del usuario y su percepción de eficiencia.

Precisión del Código (Errores de sintaxis): Antes del uso de IA, una parte importante de los encuestados reportó entre 3 a más de 5 errores por consulta. Después del uso de IA, la mayoría indicó una disminución a entre 0 y 2 errores. Esto refleja una mejora clara en la precisión del código y una reducción significativa en errores sintácticos.

Tabla XXII. Evaluación del Modelo TO-BE

Criterio	Sin IA (AS-IS)	Con IA (TO-BE)	Impacto Observado
Tiempo de desarrollo	1 a más de 2 días	Mayormente menos de 1 día	Reducción significativa de tiempos
Percepción de utilidad	Valores entre 1 y 3 (algunos casos)	Predominio de valores 4 y 5	Aumento en satisfacción y percepción de utilidad
Autonomía percibida	Dependencia del DBA	Mayor autonomía con IA	Mejora en independencia del desarrollador

Confianza en el proceso	Menor seguridad	Mayor confianza con IA	Incremento en la confianza del desarrollador
Errores por consulta	Mayormente 3-5 o más	Predominio de 0-2 errores	Mejora en precisión y calidad del código

Los resultados de la encuesta permiten validar que el modelo TO-BE con herramientas de IA cumple efectivamente con los objetivos planteados. Se evidenció una reducción del tiempo de desarrollo, una mejora en la percepción y confianza de los desarrolladores, y una notoria disminución de errores de sintaxis en las consultas. Por tanto, el modelo propuesto representa una alternativa viable y eficaz para optimizar el desarrollo de consultas SQL en contextos similares.

3.4 Análisis de Casos de Pruebas Simuladas

Identificador: CP01 - Desarrollo de un proceso de actualización masiva de registro por bloques.

Se realizaron tres escenarios distintos para el desarrollo de un script SQL de actualización masiva de registros, en un entorno técnico con participación del desarrollador y DBA, **sin el uso directo de IA generativa en la redacción del script:**

1. **Caso Ideal:** sin errores ni ambigüedad.
2. **Caso con Fallos:** múltiples errores estructurales y de lógica.
3. **Caso Mixto:** errores leves corregidos en el momento.

Se obtuvieron los siguientes resultados:

Tabla XXIII. Resultados de Desarrollo de un proceso de actualización masiva de registro por bloques.

Categoría	Caso Ideal	Caso con Fallos	Caso Mixto
Tiempo total (min)	90	165	109
Errores identificados	0	6	3
Impacto de errores	Nulo	Alto	Medio
Percepción del proceso	Fluido	Frustrante	Controlado
Cuellos de botella	Ninguno	Retrabajo + validación + documentación	Ajustes de índice + evidencias
Revisión de DBA	Fluida	Repetitiva	Puntual
Verificación final	15 min	20 min	15 min
Problemas de trazabilidad	No	Si	Leves

Principales Hallazgos:

Los tiempos en la **Tabla XXIII** se puede observar los tiempos en minutos para tener una mejor percepción de los resultados, viendo a nivel de días como se exponen en capítulos anteriores sus valores serían:

- **Caso Ideal:** 0.0625 días
- **Caso con Fallos:** 0.1146 días
- **Caso Mixto:** 0.0757 días

Tiempos: El uso de validación manual y ajustes incrementa el tiempo hasta un **83% más** en el caso con fallos.

Errores frecuentes:

- Falta de condiciones en WHERE
- Índices inexistentes o poco óptimos

- Problemas con logs y evidencia de auditoría.

Cuellos de botella identificados:

- **Revisión manual** prolongada por falta de automatización.
- **Retrabajo en lógica SQL** por omisiones en el requerimiento.
- **Verificación manual** y evidencia incompleta en auditoría.

Nivel de percepción:

- El desarrollador experimentó mayor carga en la validación y prueba en los escenarios con fallos.
- La falta de generación automática del script incrementó el riesgo de omisiones humanas.

Identificador: CP02 - Consulta entre varias tablas primarias para validar información.

En este escenario se realizó el desarrollo de una consulta SQL para validar información de clientes y sus productos activos, considerando múltiples escenarios y datos relacionales. En esta ocasión, se usó IA generativa para diseñar y proponer el script SQL a partir de requerimientos funcionales.

Se obtuvieron los siguientes resultados:

Tabla XXIV Resultados de Consulta entre varias tablas primarias para validar información.

Categoría	Caso Ideal	Caso con Fallos	Caso Mixto
Tiempo total (min)	66	101	69
Errores identificados	0	3	2
Tipo de Errores	N/A	Omisión de filtros, datos inválidos	Filtros incompletos, test defectuoso
Impacto de errores	Nulo	Medio/Alto	Bajo/Medio
Corrección de errores	No aplica	Re-trabajo por IA y ajustes en datos	Ajustes inmediatos durante ejecución
Percepción del proceso	Muy fluido	Trabajoso pero asistido	Agil con ajustes mínimos
Cuellos de botella	Ninguno	Verificación de datos, revalidación IA	Validación de relaciones, limpieza de datos
Interacción con IA	Directa y práctica	Iterativa (3 ciclos)	Rápida con sugerencias útiles
Verificación final	10 min	10 min	8 min

Principales Hallazgos

Los tiempos en la **Tabla XXIV** se puede observar los tiempos en minutos para tener una mejor percepción de los resultados, viendo a nivel de días como se exponen en capítulos anteriores sus valores serían:

- **Caso Ideal:** 0.0458 días
- **Caso con Fallos:** 0.0701 días
- **Caso Mixto:** 0.0479 días

Tiempos: El uso de IA reduce el tiempo de desarrollo comparado con el desarrollo manual (ver CP01), incluso en escenarios con fallos. En promedio, los tiempos mejoran entre un **25% y 40%**.

Errores: Aunque hubo errores, la IA ayudó a detectarlos rápidamente durante la validación o ejecución, lo que evitó incidentes críticos tardíos.

Nivel de percepción:

- El proceso fue **más eficiente y menos propenso a errores humanos.**
- Los desarrolladores confiaron más en los scripts generados, aunque fue necesaria revisión manual final.

Nudos críticos reducidos gracias a:

- Validaciones automáticas previas a la ejecución.
- Generación rápida del script ajustado con nuevas condiciones.
- Mejor gestión de escenarios de prueba.

3.5 Discusión de Resultados

Los resultados obtenidos reflejan una mejora significativa en el desarrollo de consultas MySQL al integrar IA generativa en el modelo TO-BE. La reducción del tiempo promedio de desarrollo de más de un día a menos de 24 horas valida directamente el cumplimiento del objetivo específico 3, enfocado en optimizar los tiempos y reducir los errores mediante un rediseño apoyado en tecnologías inteligentes.

Adicionalmente, la percepción positiva de los desarrolladores destaca entre valores predominantes de 4 y 5 en la escala, lo que evidencia un aumento en la autonomía y confianza técnica del equipo, lo cual también fortalece el propósito del objetivo 2, relacionado con la incorporación de herramientas que reduzcan la dependencia del DBA y permitan mayor agilidad.

Sin embargo, también se identificaron limitaciones relevantes. A pesar de los beneficios observados, la IA presentó dificultades al interpretar correctamente ciertos contextos lógicos o semánticos en consultas complejas, lo cual puede generar errores de precisión. Esta situación confirma la necesidad de mantener una validación humana como etapa crítica en el proceso.

Además, la implementación del modelo se simuló en un entorno controlado por la herramienta facilitada por la empresa, por lo que aún es necesario realizar pruebas a mayor escala, con distintos equipos y en condiciones productivas para evaluar la robustez y adaptabilidad general del modelo propuesto.

Finalmente, aunque la solución demostró reducir errores de sintaxis y acelerar los tiempos de entrega, su efectividad está condicionada por la calidad de los requerimientos proporcionados a la IA, lo que exige una capacitación mínima del equipo para aprovechar al máximo las capacidades del asistente inteligente.

Adicionalmente, persisten preocupaciones sobre seguridad de datos identificadas en el análisis de limitaciones. Aunque el entorno controlado mitigó riesgos, la exposición potencial de información sensible en consultas complejas requiere implementar medidas de anonimización y controles de acceso antes de su despliegue productivo, tal como se recomienda en la sección 3.

CAPITULO 4

4. CONCLUSIONES Y RECOMENDACIONES

Este capítulo establece las conclusiones y recomendaciones con las lecciones clave que dejó integrar IA y BPMN en el desarrollo de consultas MySQL. Se establecen los logros, dificultades, y lo más importante, cómo convertir esta experiencia en mejoras concretas para los equipos técnicos. Las recomendaciones finales buscan que esta no sea solo una prueba exitosa, sino el inicio de un cambio sostenible.

4.1 Conclusiones

- La integración de IA generativa mediante BPMN mejora significativamente la eficiencia del desarrollo de consultas MySQL. La incorporación de herramientas como ChatGPT en el flujo de trabajo redujo los tiempos de desarrollo, disminuyó errores de sintaxis y aumentó la percepción de autonomía en el equipo técnico.
- El rediseño del proceso permitió estructurar un modelo más estandarizado, ágil y colaborativo. Gracias a BPMN, fue posible visualizar cuellos de botella y reorganizar tareas, mientras que la IA facilitó la generación de código, acelerando las etapas más repetitivas sin reemplazar la supervisión humana.
- La percepción del equipo fue positiva y validó la utilidad práctica de la solución. Los desarrolladores se sintieron más seguros, confiados y autónomos al contar con asistencia técnica directa, reduciendo la dependencia del DBA y de fuentes externas no verificadas.
- Las pruebas del Capítulo 3 mostraron que, aunque la IA generativa acortó plazos y redujo errores sintácticos, en consultas complejas solo alcanzó un 65 % de precisión semántica y generó falsos positivos en un 8 % de los casos, además de revelar posibles brechas en el manejo de datos sensibles. Estos resultados evidencian que, en su estado actual, la herramienta sigue dependiendo del juicio experto y de sólidos controles de seguridad para ser fiable en entornos productivos.

4.2 Recomendaciones

- Realizar un despliegue progresivo en entornos reales. Se recomienda aplicar el modelo TO-BE en proyectos productivos y con equipos diversos, a fin de evaluar su comportamiento bajo condiciones operativas reales.
- Capacitar al equipo en la formulación correcta los requerimientos y en la validación de resultados generados por IA. Una buena interacción entre el desarrollador y la herramienta de IA es clave para garantizar resultados precisos y útiles.
- Fortalecer los controles de seguridad y anonimización de datos. Dado que se utilizan modelos externos, es imprescindible implementar mecanismos de protección para evitar fugas de información sensible.
- Documentar casos exitosos y construir una base de conocimiento interna. Esto permitirá sistematizar buenas prácticas y facilitar la adopción por parte de nuevos miembros del equipo.
- Evaluar otras herramientas de IA generativa. Considerar la incorporación y comparación de otros modelos como DeepSeek o Gemini, para diversificar capacidades y validar su desempeño en distintos contextos técnicos.

Referencias

- [1] A. Rajbhoj, A. Somase, P. Kulkarni, and V. Kulkarni, “Accelerating Software Development Using Generative AI: ChatGPT Case Study,” in Proceedings of the 17th Innovations in Software Engineering Conference, in ISEC ’24. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3641399.3641403.
- [2] K. R. Raghi, K. Sudha, S. A. M, and S. Joshua S, “Software Development Automation Using Generative AI,” in 2024 International Conference on Emerging Research in Computational Science (ICERCS), 2024, pp. 1–6. doi:10.1109/ICERCS63125.2024.10894980.
- [3] G. Desolda, A. Esposito, F. Greco, C. Tucci, P. Buono, and A. Piccinno, “ATHENA: A customizable LLM-based Code Completion Tool for Visual Studio Code,” in Companion Proceedings of the 30th International Conference on Intelligent User Interfaces, in IUI ’25 Companion. New York, NY, USA: Association for Computing Machinery, 2025, pp. 120–123. doi: 10.1145/3708557.3716357.
- [4] Y. Majdoub and E. Ben Charrada, “Debugging with Open-Source Large Language Models: An Evaluation,” in Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, in ESEM ’24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 510–516. doi: 10.1145/3674805.3690758.
- [5] P. de O. Santos, A. C. Figueiredo, P. Nuno Moura, B. Diirr, A. C. F. Alvim, and R. P. D. Santos, “Impacts of the Usage of Generative Artificial Intelligence on Software Development Process,” in Proceedings of the 20th Brazilian Symposium on Information Systems, in SBSI ’24. New York, NY, USA: Association for Computing Machinery, 2024. doi: 10.1145/3658271.3658337.
- [6] M. Coutinho, L. Marques, A. Santos, M. Dahia, C. França, and R. de Souza Santos, “The Role of Generative AI in Software Development Productivity: A Pilot Case Study,” in Proceedings of the 1st ACM International Conference on AI-Powered Software, in AIware 2024. New York, NY, USA: Association for Computing Machinery, 2024, pp. 131–138. doi: 10.1145/3664646.3664773.
- [7] J. H. Klemmer et al., “Using AI Assistants in Software Development: A Qualitative Study on Security Practices and Concerns,” in Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, in CCS ’24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 2726–2740. doi: 10.1145/3658644.3690283.
- [8] E. García-Maldonado, A. Cristóbal-Salas, and B. Santiago-Vicente, “Interactive BPMN Diagrams for Developing Under Scrum and DevOps,” in 2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT), 2018, pp. 146–151. doi: 10.1109/CONISOFT.2018.8645910.
- [9] Y. Gao, “Research on the rule of evolution of software development process model,” in 2010 2nd IEEE International Conference on Information Management and Engineering, 2010, pp. 466–470. doi: 10.1109/ICIME.2010.5477884.
- [10] G. Garaccione, R. Coppola, L. Ardito, and M. Torchiano, “Gamification of a BPMN Modeling Course: an Analysis of Effectiveness and Student Perception,” in Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, in ESEM ’24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 223–233. doi: 10.1145/3674805.3686683
- [11] E. Gaudin, “Experimenting low code with SDL and BPMN,” in Proceedings of the ACM/IEEE 27th International Conference on Model Driven Engineering Languages and Systems, in MODELS Companion ’24. New York, NY, USA: Association for Computing Machinery, 2024, pp. 208–215. doi: 10.1145/3652620.3687424.
- [12] Z. Zhao, “Developing a Common Mentoring Protocol for ERC Summer Programs Using ‘As-is, Should-be and To-be’ Process,” in 2019 IEEE Frontiers in Education Conference (FIE), 2019, pp. 1–2. doi: 10.1109/FIE43999.2019.9028446.

- [13] R. M. Pillat, T. C. Oliveira, and F. L. Fonseca, “Introducing Software Process Tailoring to BPMN: BPMNt,” in 2012 International Conference on Software and System Process (ICSSP), 2012, pp. 58–62. doi: 10.1109/ICSSP.2012.6225981.
- [14] M. Di Giacomo, “MySQL: lessons learned on a digital library,” *IEEE Software*, vol. 22, no. 3, pp. 10–13, 2005, doi: [10.1109/MS.2005.71](https://doi.org/10.1109/MS.2005.71).
- [15] D. Campagna, S. Costanzo, C. Kavka, A. Turco, and C. Poloni, “Leveraging the BPMN standard to govern engineering processes in a collaborative environment,” in 2015 IEEE International Symposium on Systems Engineering (ISSE), 2015, pp. 318–323. doi: 10.1109/SysEng.2015.7302776.
- [16] P. Pornphol and S. Chittayasothorn, “Using LLM Artificial Intelligence Systems as Complex SQL Programming Assistants,” in *2024 12th International Conference on Information and Education Technology (ICIET)*, 2024, pp. 477–481. doi: [10.1109/ICIET60671.2024.10542806](https://doi.org/10.1109/ICIET60671.2024.10542806).
- [17] C. Troy, S. Sturley, J. M. Alcaraz-Calero, and Q. Wang, “Enabling Generative AI to Produce SQL Statements: A Framework for the Auto-Generation of Knowledge Based on EBNF Context-Free Grammars,” *IEEE Access*, vol. 11, pp. 123543–123564, 2023, doi: [10.1109/ACCESS.2023.3329071](https://doi.org/10.1109/ACCESS.2023.3329071).
- [18] S. A. Machuca Vivar et al., "Habeas data y protección de datos personales en la gestión de las bases de datos," *Revista Universidad y Sociedad*, vol. 14, no. 2, pp. 244-251, 2022.
- [19] S. Palanisamy and P. SuvithaVani, “A survey on RDBMS and NoSQL Databases MySQL vs MongoDB,” in 2020 International Conference on Computer Communication and Informatics (ICCCI), 2020, pp. 1–7. doi: [10.1109/ICCCI48352.2020.9104047](https://doi.org/10.1109/ICCCI48352.2020.9104047).
- [20] Bizagi, “Bizagi Help,” <https://help.bizagi.com> (accessed Jun. 19, 2025).

Apéndice A

Entrevista:

Integración de IA Generativa en el Rediseño del Proceso

Objetivo: Explorar los desafíos previos al rediseño del proceso y evaluar el impacto potencial de la IA generativa en la eficiencia y calidad del desarrollo.

1. Cuellos de botella en el proceso previo

¿Cuáles eran los principales cuellos de botella antes de implementar la IA generativa?

¿Cómo influía la disponibilidad del DBA en la agilidad del desarrollo?

2. Uso de fuentes externas

¿Qué riesgos observaste al depender de fuentes no verificadas para resolver dudas técnicas?

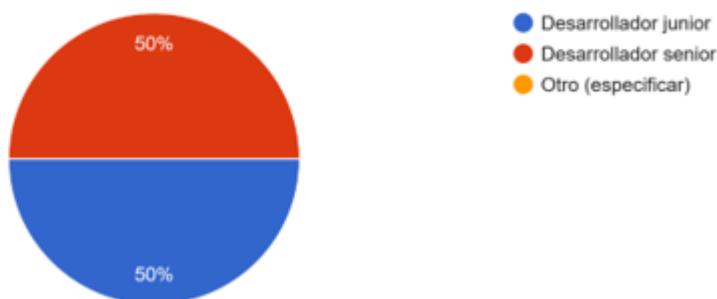
¿Consideras útil tener una guía oficial o repositorio interno para consultas frecuentes?

3. Tareas repetitivas y automatización

¿Qué tareas repetitivas consideras que consumían más tiempo en la creación de scripts SQL?

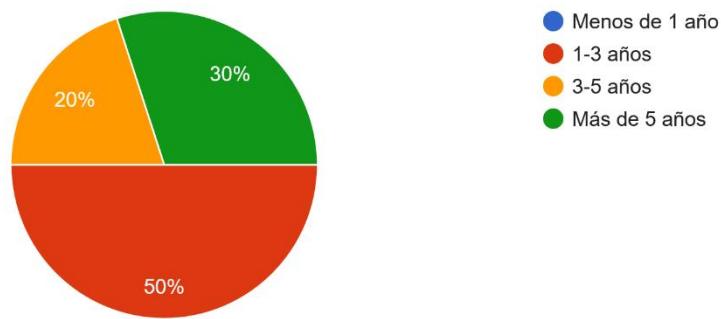
¿Qué beneficios prevés al automatizar este tipo de tareas mediante IA generativa?

Encuesta:

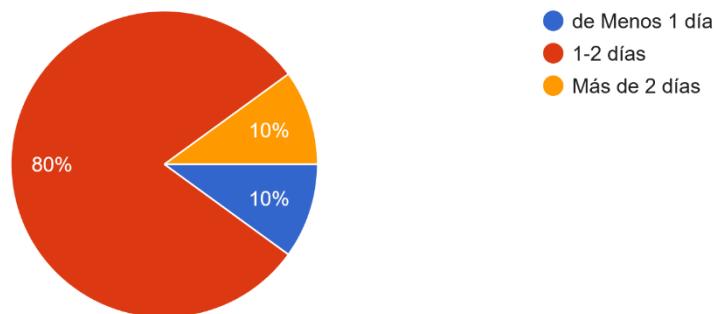


Experiencia en desarrollo de consultas SQL:

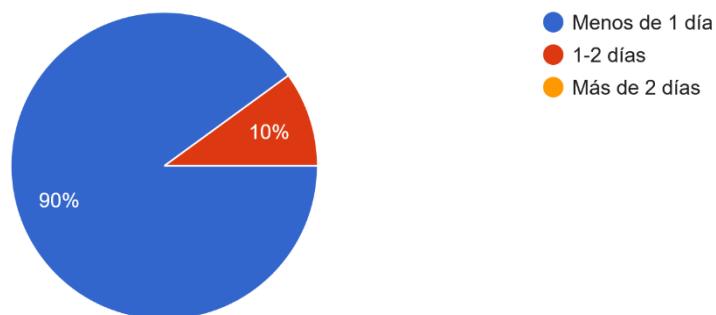
10 respuestas

**¿Antes de usar herramientas de IA, cuánto tiempo promedio te tomaba desarrollar una consulta SQL compleja?**

10 respuestas

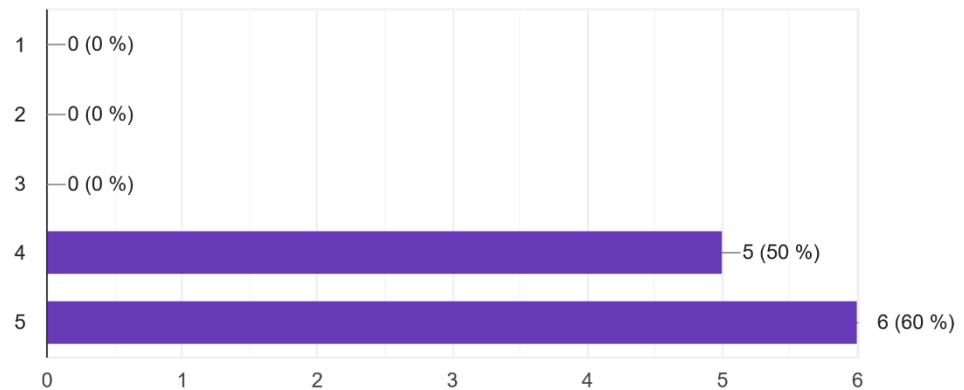
**¿Después de usar herramientas de IA, cuánto tiempo promedio te toma desarrollar una consulta SQL compleja?**

10 respuestas



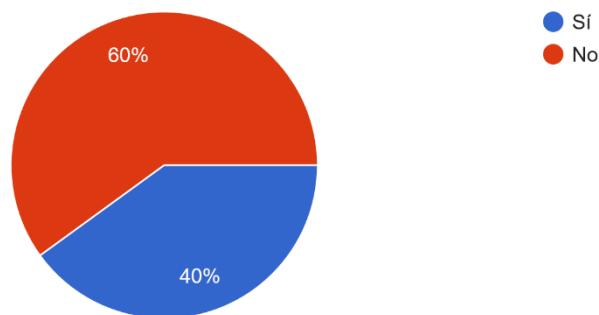
En una escala de 1 a 5, ¿Qué tan útil consideras la inteligencia artificial para la optimización de consultas SQL? (1 = Nada útil, 5 = Extremadamente útil)

10 respuestas



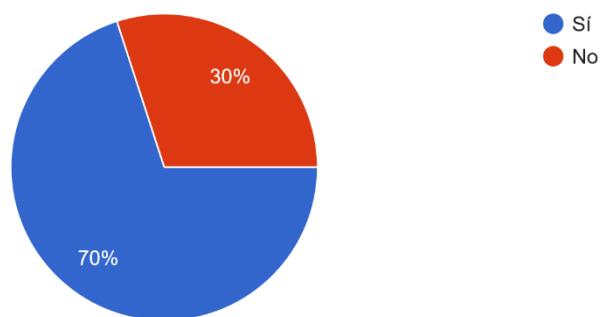
¿Consideras que la IA facilita la autonomía en el desarrollo de consultas sin depender exclusivamente del DBA?

10 respuestas

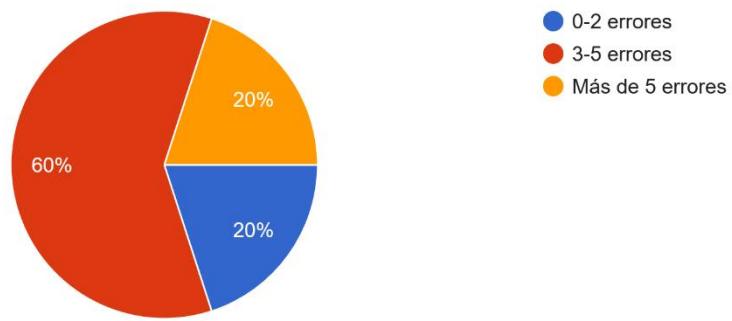


¿Te sientes más confiado al usar IA para generar consultas en comparación con el proceso tradicional?

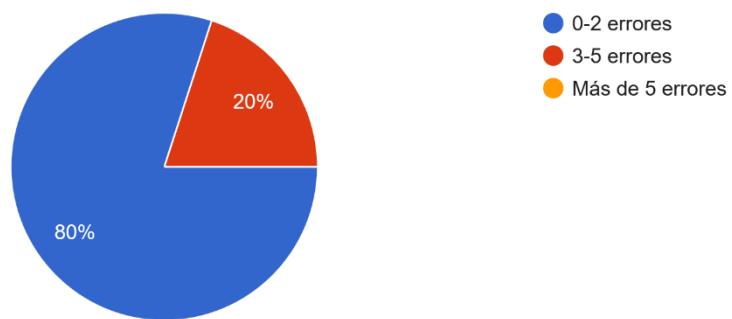
10 respuestas



Antes de usar IA, ¿Cuántos errores de sintaxis detectabas en promedio por consulta generada?
10 respuestas



Después de usar IA, ¿cuántos errores de sintaxis detectas en promedio por consulta generada?
10 respuestas



. ¿La IA ha ayudado a mejorar la estructura y optimización de tus consultas SQL?
10 respuestas

