

## Trabajo Final Integrador (TFI) - Bases de Datos I

### Introducción

Este Trabajo Final Integrador tiene como finalidad consolidar los contenidos de la asignatura **Bases de Datos I** a través de un proyecto único y progresivo, articulado con el desarrollo de Programación (por ejemplo, **Java** con JDBC) **o, alternativamente, con procedimientos SQL en el SGBD**, según la opción elegida en cada etapa.

Se busca simular un entorno real de gestión de datos, se incorporan actividades que simulan el trabajo con bases de datos con volúmenes importantes de datos (miles de registros), consultas optimizadas, seguridad aplicada y manejo de concurrencia, integrando aspectos de diseño, carga de información, consultas, seguridad, integridad y concurrencia.

Se promueve además el uso responsable de herramientas de **Inteligencia Artificial (IA)** como apoyo en el proceso de aprendizaje. Los estudiantes podrán valerse de IA generativa y asistentes inteligentes como **ChatGPT (OpenAI)**, **Gemini (Google DeepMind)**, **Claude (Anthropic)** o **GitHub Copilot (Microsoft)**, entre otras. El rol de estas herramientas será de **tutoría y acompañamiento**, ayudando a formular consultas, corregir razonamientos y explorar alternativas de solución, sin reemplazar el trabajo crítico del estudiante.

Esta guía para el TFI se organiza en **dos partes complementarias**:

1. **Uso pedagógico de la Inteligencia Artificial (IA)**: se presenta una reflexión y guía de trabajo sobre cómo integrar herramientas de IA como recurso de apoyo al aprendizaje, entendiendo su rol actual en el mundo profesional de la programación y su potencial pedagógico en la formación académica.
2. **Desarrollo del Trabajo Final Integrador**: se detallan las consignas que los estudiantes deberán realizar, organizadas en cinco etapas progresivas que abarcan el modelado, la implementación, la carga masiva de datos, las consultas avanzadas, la seguridad y la concurrencia en bases de datos.

### Nota pedagógica.

*Algunos contenidos técnicos de esta etapa pueden no haber sido tratados en profundidad durante el cursado. En estos casos, se espera que los estudiantes utilicen herramientas de inteligencia artificial como apoyo para **comprender, adaptar o generar ejemplos seguros**, citando las interacciones en el anexo correspondiente.*

*Este enfoque busca evaluar la **capacidad de aprendizaje autónomo y uso crítico de la IA** como parte integral del proceso formativo.*

## Uso pedagógico de la Inteligencia Artificial (IA)

### Introducción

Hoy en día, la **Inteligencia Artificial (IA)** está presente de forma ubicua en el trabajo cotidiano de los programadores. Desde asistentes integrados en los entornos de desarrollo hasta sistemas capaces de generar código, la IA se ha convertido en una herramienta habitual de apoyo técnico y creativo.

Frente a este panorama, resulta fundamental integrarla también en el **ámbito pedagógico**, no como un atajo que entregue soluciones finales, sino como un **aliado en el proceso de aprendizaje**. El objetivo es que los estudiantes aprendan a **dialogar con la IA a través de prompts bien diseñados**, obteniendo pistas, explicaciones y guías que los ayuden a construir por sí mismos las soluciones, en lugar de recibirlas terminadas.

De esta manera, se fomenta la autonomía, el pensamiento crítico y la capacidad de utilizar la IA como un recurso formativo en lugar de depender de ella de manera pasiva.

### Objetivos

- Guiar al alumno en la resolución asistida y documentada.
- Evaluar el proceso además del resultado.
- Reducir la carga de corrección de los entregables relacionados con IA (naturalmente extensos) usando IA también para resumir y extraer evidencias.

### Metodología

1. El alumno copia y pega un prompt inicial con la consigna.
2. Trabaja de forma iterativa con la IA, haciendo preguntas, corrigiendo y refinando soluciones mientras hace pruebas en los entornos correspondientes.
3. Entrega como evidencia del uso de IA el chat completo con la misma.

#### Ejemplo breve de prompt:

*"Estoy desarrollando un script SQL para generar 100.000 registros ficticios en la tabla ventas. Estoy trabajando en MySQL. Necesito que me guíes paso a paso, sin darme todo el código de golpe."*

*(El prompt completo inicial sugerido figura en el **Anexo I** al final de este documento.)*

## Entregables

- Entrega: incluir un anexo con el texto completo de todos los chats con IA utilizados en este trabajo.
  - Puede ir al final del mismo PDF del TPI o en un PDF complementario titulado: Anexo IA – [Equipo/Apellido(s)].
  - Formato preferido: texto (copiar y pegar el contenido del chat) en un anexo. Se solicita este formato porque es más sencillo de procesar.
  - **Se aceptan capturas si ya fueron realizadas previamente** (equipos que avanzaron antes de este ajuste). En ese caso, agregar debajo de cada imagen fecha y etapa a la que corresponde.
  - Organización mínima: separar por Tema/Etapa del TPI.
  - Criterios de evaluación: legibilidad, organización por tema/etapa, integridad y trazabilidad con las decisiones técnicas del trabajo. La ausencia del anexo o su entrega parcial podrá afectar la calificación.
- Producto final funcional (consultas, script, código Java en caso de elegir esta opción, etc.).

### Nota sobre el uso de IA en las actividades mínimas

Las actividades mínimas de cada etapa incluyen instancias explícitas de interacción con herramientas de Inteligencia Artificial, en coherencia con el *Prompt General* detallado en el Anexo I. Estas interacciones no reemplazan el trabajo del estudiante, sino que funcionan como tutoría pedagógica: la IA debe guiar con pistas, sugerencias y correcciones parciales, mientras el estudiante mantiene un rol activo en el diseño, la prueba y la validación de las soluciones. La entrega final debe incorporar como evidencia los fragmentos de chat (en el anexo correspondiente) que muestren este proceso reflexivo de acompañamiento.

## Beneficios pedagógicos

- Potencia el aprendizaje guiado y fomenta la autonomía.
- Documenta decisiones y correcciones realizadas durante el proceso.
- Facilita la evaluación basada en evidencias reales.
- Prepara al estudiante para un uso profesional y crítico de la IA.

## Desarrollo del Trabajo Final Integrador

### Resumen Etapas

**Etapla 1 – Modelado y Definición de Constraints:** Elaborar el DER y el modelo relacional con PRIMARY KEY, FOREIGN KEY, UNIQUE, CHECK y restricciones de dominio.

**Etapla 2 – Generación y carga de datos masivos con SQL puro :** Generación de datos masivos con SQL puro (entre 10.000 y 500.000 filas distribuidas en 2+ tablas).

**Etapla 3 – Consultas Avanzadas y Reportes:** Desarrollo de consultas complejas con JOIN, GROUP BY, HAVING, subconsultas y vistas.

**Etapla 4 – Seguridad e Integridad:** Creación de usuarios con privilegios mínimos, vistas para ocultar información sensible, validación de restricciones y consultas seguras en Java con PreparedStatement o equivalente en SQL.

**Etapla 5 – Concurrencia y Transacciones:** Simulación de bloqueos y deadlocks; niveles de aislamientos; transacciones en *Java o SQL (procedimientos almacenados)* con *retry* ante deadlock.

### Beneficios pedagógicos

- Integración transversal entre Programación y Bases de Datos.
- Consolidación de SQL avanzado y constraints.
- Exposición a escenarios de seguridad, integridad y concurrencia.
- Desarrollo de competencias críticas en entornos realistas.
- Fomento del uso responsable y reflexivo de IA como tutoría de aprendizaje.

### ETAPAS DETALLADAS

#### Etapla 1 – Modelado y Definición de Constraints

##### Descripción

Partiendo del enunciado del CRUD en Java (materia Programación), se debe realizar el modelo entidad-relación y el correspondiente modelo relacional, incorporando todas las restricciones de integridad necesarias:

- Primary Key (PK)
- Foreign Key (FK)
- UNIQUE
- CHECK
- Restricciones de dominio (tipos de datos, longitudes, obligatoriedad de nulos, rangos de valores).

Esto asegura que la base de datos no solo esté correctamente diseñada en lo conceptual, sino también validada en lo físico antes de comenzar con la **carga masiva de datos**.

### Objetivos

- Comprender el vínculo entre el modelado conceptual y la implementación física.
- Aplicar constraints para garantizar la integridad de los datos.
- Asegurar la consistencia en la base de datos antes de la carga masiva.

### Metodología

1. El alumno parte del modelo de dominio definido en Programación.
2. Identifica las entidades y relaciones principales.
3. Define para cada tabla los atributos y sus restricciones de dominio (tipo, tamaño, obligatoriedad).
4. Incorpora constraints (PK, FK, UNIQUE, CHECK).
5. Prueba la robustez del diseño intentando **inserciones válidas e inválidas**, analizando los mensajes de error.
6. Interactúa con la IA como tutor pedagógico, pidiendo pistas para:
  - Validar que las claves elegidas sean correctas.
  - Revisar si hay redundancias o inconsistencias.
  - Chequear si los dominios de atributos son apropiados.

### Actividades mínimas

- El DER deberá tener 2 entidades (las del TFI de PROG II) o más si el equipo incorpora una tercera entidad auxiliar coherente con el dominio.
- Construir el modelo relacional en SQL con PK, FK, UNIQUE, CHECK y restricciones de dominio.
- Validación práctica: ejecutar 2 inserciones correctas y 2 erróneas mostrando mensajes de error distintos (ej. violación de UNIQUE y CHECK o FK).
- Interacción con IA: consultar a la IA sobre la elección de claves y restricciones (ej. “¿mi elección de PK es correcta o convendría otra?”) y adjuntar el texto de la interacción como evidencia en el anexo correspondiente.

### Entregables

- DER en PDF o imagen.
- Script SQL con creación de tablas y constraints.
- Evidencia de IA: chat completo en el anexo correspondiente.

### Beneficios pedagógicos

- Permite reflexionar sobre cada decisión de modelado y su impacto en la integridad.
- Refuerza la capacidad de abstraer un problema del dominio y llevarlo a un modelo entidad-relación.
- Desarrolla la habilidad de definir claves y restricciones correctas, esenciales en cualquier proyecto de bases de datos.
- Ayuda a comprender el impacto de las restricciones en la consistencia y robustez de un sistema real.
- Fomenta el hábito de validar modelos con ejemplos prácticos, una práctica profesional crítica.

## Etapa 2. Generación y carga de datos masivos con SQL puro

### Descripción

Generar datos ficticios en **gran volumen** usando **exclusivamente mecanismos de SQL**, de modo que las etapas posteriores (consultas, seguridad, concurrencia y performance) se ejecuten sobre un tamaño realista.

Para pruebas funcionales y validaciones iniciales del modelo, se acepta una cantidad aproximada de 10.000 registros totales, distribuidos entre las tablas principales del esquema.

No obstante, para realizar mediciones de rendimiento realmente útiles —como comparar los tiempos de ejecución con y sin índices o evaluar la eficiencia de distintas consultas— se recomienda escalar la base de datos a un volumen de entre 200.000 y 500.000 registros totales.

Este rango no impacta directamente en la calificación del trabajo, pero permite obtener resultados de medición más precisos y con mayor valor analítico, lo cual enriquece la interpretación de la performance del sistema.

**Propósito del volumen (justificación):** este tamaño permite observar diferencias de desempeño entre consultas equivalentes con y sin índice, reproducir escenarios de contención realistas en etapas posteriores y obtener resultados estables (no ruidosos) al repetir consultas. No se busca optimizar aquí, sino **disponer de un conjunto de datos de gran volumen y coherente** para las siguientes etapas.

### Objetivo de aprendizaje

Comprender, **en términos conceptuales**, cómo se puede **generar un conjunto de datos grande y consistente** respetando el dominio (cardinalidades, integridad, distribuciones), sin exigir el entendimiento detallado de cada técnica de generación.

### Condiciones de llenado

- **Integridad referencial:** todas las FKs deben apuntar a PKs válidas.
- **Cardinalidades del dominio** respetadas.
- **Reproducibilidad conceptual:** que el proceso pueda repetirse con resultados comparables.

*Nota de acompañamiento didáctico: En esta etapa, el foco está en **comprender la lógica general de generación de datos**, no en dominar cada instrucción SQL avanzada. Se recomienda utilizar fuertemente herramientas de IA como apoyo para **explorar estrategias posibles y entender los mecanismos**. Lo que se evaluará es la **coherencia del conjunto de datos y la comprensión conceptual**, no la autoría literal del script.*

### Mecanismos posibles:

- **Tablas “semilla” y producto cartesiano controlado** para combinar valores (CREATE TABLE, INSERT ... SELECT, CROSS JOIN, JOIN, WHERE, LIMIT).
- **Series/secuencias** para enumerar y derivar atributos técnicos (WITH RECURSIVE, UNION ALL, ROW\_NUMBER() OVER (), INSERT ... SELECT, CONCAT, LPAD, MOD).
- **Tablas calendario** para generar fechas/horas de forma sistemática (WITH RECURSIVE, UNION ALL, DATE\_ADD/ADDDATE, INTERVAL, DATE\_FORMAT).
- **Catálogos/maestras** con cuotas por categoría/estado/rol (CREATE TABLE, FOREIGN KEY, UNIQUE, JOIN, ROW\_NUMBER OVER , MOD, CASE WHEN).
- **Pseudoaleatoriedad con sesgos** (para evitar distribuciones uniformes) (RAND(), CASE WHEN, FLOOR/ROUND, BETWEEN, INSERT ... SELECT).
- **Expansiones iterativas o recursivas** (si el motor lo permite) para aumentar el volumen por niveles (WITH RECURSIVE, UNION ALL, INSERT ... SELECT, SELF JOIN).

### Actividades mínimas

1. Producir un **poblado de tablas masivo y consistente** que cumpla las **Condiciones de llenado** y alcance el **volumen elegido**.
2. Entregar una **descripción conceptual (1 página)** del mecanismo elegido:
  - Tablas semilla/maestras usadas.
  - Cómo se garantizó integridad y cardinalidades.



3. **Pruebas de consistencia:** 3–5 verificaciones simples (conteos, FK huérfanas = 0, cardinalidades del dominio respetadas, porcentaje por tablas respetado según lo planificado, rango de valores de dominio respetados)
4. **Interacción con IA:** adjuntar evidencia según lo indicado en el **entregable de IA** (ver sección *Entregables – Evidencia de uso de IA* al inicio de este documento).

### Entregables

- **Descripción conceptual** del llenado.
- **Cuadro de verificaciones** (conteos).
- **Evidencia de interacción con la IA** (en el anexo correspondiente).

## Etapa 3. Consultas complejas y útiles a partir del CRUD inicial

### Descripción

Partiendo del CRUD desarrollado en Programación (**por ejemplo, en Java/JDBC**), en Bases de Datos I deberán diseñar consultas SQL complejas y útiles que agreguen valor al sistema. Por ejemplo: reportes agrupados, análisis temporales, filtros avanzados o combinaciones de tablas mediante JOIN.

### Objetivos

- Pasar de consultas simples a consultas multi-tabla y agregadas.
- Diseñar consultas que tengan utilidad práctica en el sistema.
- Entender el impacto de la estructura de datos y los índices en consultas complejas.

### Metodología

1. Identificar áreas del CRUD donde un reporte agregue valor.
2. Diseñar consultas con JOIN, GROUP BY, HAVING, subconsultas y vistas.
3. Probar con grandes volúmenes de datos.
4. **Medir tiempos con y sin índices** en al menos 3 consultas representativas:
  - Igualdad (p. ej., WHERE col = ? o IN (...)),
  - Rango (BETWEEN, >=, <=, LIKE 'pref%'),
  - JOIN (p. ej., A.fk = B.pk).

Ejecutar cada consulta 3 veces y reportar la mediana de los tiempos. Incluir EXPLAIN (y EXPLAIN ANALYZE si está disponible). Para comparar sin índice, usar DROP INDEX o IGNORE INDEX; para asegurar el camino con índice, usar FORCE INDEX si fuese necesario.

### Actividades mínimas

- Diseñar 4 consultas: al menos 2 con JOIN (entre A↔B y/o con tablas auxiliares del dominio), 1 con GROUP BY + HAVING y 1 con subconsulta.
- Crear al menos 1 vista útil en el contexto del sistema.



- Documentar brevemente (3–5 líneas) la utilidad práctica de cada consulta.
- **Medición comparativa con/sin índice** en 3 consultas (igualdad, rango, JOIN) con breve conclusión (5–10 líneas).
- Interacción con IA: plantear una consulta compleja o a optimizar y mostrar cómo la IA ayudó a mejorarla o a validarla.

### Entregables

- Scripts SQL de las consultas.
- Capturas de resultados.
- **Tabla de tiempos** (con/sin índice; 3 corridas → mediana) + EXPLAIN.
- Breve explicación de utilidad y lógica de cada consulta.

### Beneficios pedagógicos

- Enfoque en resolución de problemas reales.
- Mejora de la capacidad de pensar en términos de datos.
- Consolidación de conocimientos de SQL intermedio/avanzado.

## Etapa 4. Seguridad e integridad

### Descripción

Implementar medidas de seguridad en el acceso a datos y garantizar la integridad de la base. Esto implica aplicar mínimos privilegios, usar vistas para exponer datos, evitar SQL injection mediante consultas parametrizadas (en Java con PreparedStatement o en SQL con procedimientos almacenados sin SQL dinámico) y demostrar que las restricciones de integridad se cumplen.

### Objetivos

- Aplicar la regla de usuario con mínimos privilegios.
- Usar vistas para filtrar y simplificar acceso.
- Respetar integridad referencial y unicidad.
- Aplicar seguridad en el código Java o SQL, según la opción elegida.

### Metodología

1. Creación de un usuario con permisos acotados.
2. Vistas que oculten información sensible.
3. Pruebas de restricciones PK, FK, UNIQUE, CHECK.
4. En Java, uso de PreparedStatement y validaciones o equivalente en SQL.

### Actividades mínimas

- Crear un usuario con privilegios mínimos y mostrar pruebas de acceso restringido.
- Diseñar 2 vistas que oculten información sensible.

- Ejecutar al menos 2 pruebas de integridad (ej. duplicación de PK, inserción fuera de rango, violación de FK).
- Implementar una consulta segura, ya sea A) en Java con PreparedStatement, o B) en SQL mediante procedimiento almacenado sin SQL dinámico; en ambos casos, adjuntar prueba anti-inyección documentada (intento malicioso bloqueado + breve explicación).

### Entregables

- Script SQL con creación de usuario, permisos y vistas.
- Pruebas de integridad.
- Código cliente (Java) seguro o procedimiento SQL equivalente (sin SQL dinámico), según opción elegida. **Prueba anti-inyección documentada** (entrada maliciosa utilizada, resultado esperado y breve explicación de por qué quedó neutralizada).
- Evidencia del proceso de interacción con la IA (en el anexo correspondiente)

### Beneficios pedagógicos

- Comprensión práctica de la seguridad en bases de datos.
- Aprendizaje del uso real de vistas.
- Reforzamiento de buenas prácticas en desarrollo seguro.

## Etapas 5. Concurrencia y transacciones

### Descripción

Diseñar y ejecutar pruebas que muestren el comportamiento de la base ante accesos simultáneos, implementando transacciones y evaluando bloqueos, deadlocks y niveles de aislamiento.

**Nota de alcance.** Aquí se observa **cómo los índices afectan la contención** (bloqueos/esperas) en **acceso simultáneo**; la **comparación de tiempos crudos aproximados con/sin índice** ya se realizó en **Etapas 3**.

### Objetivos

- Implementar transacciones en operaciones críticas.
- Comparar niveles de aislamiento en MySQL.
- Diagnosticar y manejar bloqueos y deadlocks.
- Medir impacto de índices en entornos concurrentes.

### Metodología

1. Simulación con dos sesiones MySQL para generar bloqueos y deadlocks.
2. Comparación de READ COMMITTED y REPEATABLE READ.

3. Uso **opcional** de mysqlslap para carga concurrente.
4. Implementación de retry ante deadlock en Java o SQL (procedimiento almacenado).

#### Actividades mínimas

- Simular al menos 1 deadlock en dos sesiones y documentar el error generado.
- Implementar **en Java o en SQL (procedimiento almacenado)** un ejemplo de transacciones con START TRANSACTION/COMMIT/ROLLBACK y logging de errores. Si hay deadlock (1213/SQLSTATE 40001), aplicar **retry** hasta 2 reintentos con backoff breve.
- Comparar en la práctica 2 niveles de aislamiento (ej. READ COMMITTED y REPEATABLE READ) mostrando diferencias con ejemplos simples (basta con un ejemplo breve que muestre una diferencia observable; no se requiere análisis de phantoms ni next-key locks).
- Elaborar un breve informe (5–10 líneas) con las observaciones sobre concurrencia y transacciones.
- Interacción con IA: usarla como apoyo para interpretar los resultados de las pruebas de concurrencia o sugerir pasos adicionales, dejando evidencia del intercambio.

#### Entregables

- Guion de pruebas SQL.
- Resultados de mediciones.
- Código cliente (Java) o procedimiento SQL con transacciones y manejo de errores (incluyendo estrategia de retry si corresponde).
- Evidencia del proceso.

#### Beneficios pedagógicos

- Vivencia directa de problemas de concurrencia.
- Comprensión de la importancia de índices y diseño.
- Mejora en el manejo de transacciones y recuperación de errores.

## RUBRICA DE EVALUACIÓN

Etapa	Peso	Criterios de evaluación
<b>Etapa 1: Modelado y Constraints</b>	20%	<ul style="list-style-type: none"> <li>- DER con al menos 2 entidades, atributos y relaciones bien cardinalizadas.</li> <li>- Modelo relacional en SQL con PK, FK, UNIQUE, CHECK y dominios.</li> <li>- Validación con 2 inserciones correctas y 2 erróneas que disparen errores distintos.</li> <li>- Evidencia de interacción con IA sobre claves y restricciones.</li> <li>- <b>IA (5%)</b>: uso reflexivo, no copia literal.</li> </ul>
<b>Etapa 2: Implementación y Carga Masiva</b>	20%	<ul style="list-style-type: none"> <li>- Generación de datos con SQL puro (<b>mínimo 10 000 registros</b>, preferentemente escalable a <b>200 000–500 000</b> para mediciones de performance útiles).</li> <li>- Verificaciones de consistencia (FK válidas, cardinalidades) correctamente documentadas.</li> <li>- Evidencia de interacción con IA sobre estrategia de generación de datos.</li> <li>- <b>IA (5%)</b>: aplicación práctica de la sugerencia recibida.</li> </ul>
<b>Etapa 3: Consultas Avanzadas y Reportes</b>	25%	<ul style="list-style-type: none"> <li>- 4 consultas: 2 JOIN, 1 GROUP BY/HAVING, 1 subconsulta.</li> <li>- Medición comparativa con/sin índice + conclusión.</li> <li>- Creación de al menos 1 vista útil.</li> <li>- Documentación breve de la utilidad de cada consulta.</li> <li>- Evidencia de interacción con IA para optimización o diseño de consulta.</li> <li>- <b>IA (5%)</b>: calidad de la reflexión, no mera copia.</li> </ul>
<b>Etapa 4: Seguridad e Integridad</b>	15%	<ul style="list-style-type: none"> <li>- Usuario con privilegios mínimos y prueba de restricciones.</li> <li>- Creación de 2 vistas para ocultar datos sensibles.</li> <li>- 2 pruebas de integridad documentadas.</li> <li>- Implementación de consulta parametrizada segura: A) Java con PreparedStatement o B) SQL con procedimiento almacenado sin SQL dinámico.</li> <li>- Evidencia de interacción con IA sobre seguridad y buenas prácticas.</li> <li>- <b>IA (5%)</b>: integración real de la sugerencia.</li> </ul>
<b>Etapa 5: Concurrencia y Transacciones</b>	20%	<ul style="list-style-type: none"> <li>- Simulación de 1 deadlock con documentación.</li> <li>- Ejemplo en Java o en SQL (procedimiento almacenado) con BEGIN/COMMIT/ROLLBACK, manejo de errores y retry ante deadlock si aplica.</li> <li>- Comparación práctica de 2 niveles de aislamiento.</li> <li>- Informe breve (5–10 líneas) con observaciones.</li> <li>- Evidencia de interacción con IA para interpretar resultados de concurrencia.</li> <li>- <b>IA (5%)</b>: uso para interpretación, no para “responder por el alumno”.</li> </ul>

## Anexo I – Prompt completo para uso pedagógico de IA

### PROMPT PARA IA

#### Contexto

*Estás interactuando con un estudiante de la materia **Base de Datos I** de una Tecnicatura Superior. El estudiante está resolviendo un Trabajo Práctico sobre modelos entidad-relación y modelo relacional.*

#### Instrucciones para la IA

1. *No des la solución completa de manera directa.*
2. *Ofrece **pistas graduales** que guíen al estudiante hacia la respuesta correcta.*
3. *Señala los errores conceptuales o de sintaxis de forma clara y respetuosa.*
4. *Explica los **motivos de cada corrección** y, cuando corresponda, da ejemplos similares que ayuden a entender mejor.*
5. *Si el estudiante se queda bloqueado, sugiere pasos intermedios o preguntas disparadoras.*
6. *Utiliza un lenguaje accesible, evitando tecnicismos innecesarios.*

#### Ejemplo de interacción esperada

- *Si el estudiante pregunta: “¿Está bien si pongo la clave primaria en esta tabla así?”*
  - *La IA responde: “Revisemos: la clave primaria debe identificar unívocamente a cada fila. En tu caso, ¿qué pasa si hay dos estudiantes con el mismo apellido? Tal vez convenga otra opción. ¿Cuál columna pensás que nunca se repite?”*

#### Recordatorio

*El objetivo es que el estudiante **razone y corrija su propio trabajo**, no que copie una respuesta ya hecha.*