

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**FACULTAD FTL**

**Trabajo de Curso de Sistemas de Bases de Datos I**

Autores:  
Lester Cornielt Tamayo   
Moisés Samuel Delgado Docampo

**La Habana, junio 2025**

**“Año 67 de la Revolución”**

# Resumen

El presente trabajo tuvo como objetivo el desarrollo de un sistema de reservas para un hotel, incluyendo el diseño e implementación de una base de datos relacional que garantizara la integridad y eficiencia en la gestión de la información. Se inició con el análisis de soluciones homólogas y la identificación de sus deficiencias, lo que permitió justificar el diseño propuesto. Se elaboró un Diagrama Entidad-Relación cumpliendo los requisitos académicos establecidos, incluyendo la utilización de atributos identificadores, compuestos, multivaluados y derivados, así como relaciones 1:1, 1:N y N:M. Además, se aplicaron extensiones avanzadas del modelo. El modelo fue transformado a un esquema relacional y normalizado. Posteriormente, se implementó la base de datos utilizando SQL, abarcando la creación de tablas, vistas, funciones y consultas de manipulación de datos. Finalmente, se desarrolló una aplicación en Java con interfaz por línea de comandos (CLI) que permitió la interacción con el sistema. Los resultados demostraron un correcto funcionamiento, garantizando la gestión integral de clientes, reservas, habitaciones y reportes administrativos. Se evidenció la importancia del diseño lógico y físico en la creación de sistemas eficientes para el sector turístico.

**Palabras clave**: Base de datos; Sistema de reservas; Hotel; Modelo relacional; Aplicación CLI; SQL

# Índice

1. ***Introducción***
2. ***Capítulo I. Estudio del estado del arte  
   2.1 Análisis de los conceptos asociados al problema  
   2.2 Análisis crítico de soluciones homólogas  
   2.3 Conclusiones parciales***
3. ***Capítulo II. Diseño de la propuesta de solución  
   3.1 Metodología para el diseño de la base de datos  
   3.2 Modelo Entidad-Relación  
   3.3 Normalización de la base de datos  
   3.4 Modelo Relacional  
   3.5 Conclusiones parciales***
4. ***Capítulo III. Implementación de la base de datos  
   4.1 Herramientas seleccionadas  
   4.2 Lenguaje de Definición de Datos  
   4.3 Lenguaje de Manipulación de Datos  
   4.4 Principales funciones implementadas  
   4.5 Implementación de la aplicación  
   4.6 Conclusiones parciales***
5. ***Conclusiones generales***
6. ***Recomendaciones***
7. ***Referencias bibliográficas***
8. ***Anexos***

# Introducción

*El avance de las tecnologías de la información y las comunicaciones ha transformado significativamente la forma en que se gestionan los servicios en la industria hotelera. En este contexto, la automatización de procesos como las reservas, el control de habitaciones y la generación de reportes resulta esencial para lograr una gestión eficiente, oportuna y segura. Sin embargo, en muchos hoteles todavía se emplean métodos manuales o soluciones poco integradas, lo que provoca errores humanos, demoras en la atención al cliente y pérdida de información.*

*El problema identificado radica en la ausencia de un sistema informatizado que permita a un hotel gestionar, de forma centralizada, las reservas realizadas por los clientes, así como la administración de las habitaciones, el personal involucrado y los datos estadísticos para la toma de decisiones. Ante esta situación, se plantea el diseño e implementación de un sistema de reservas para hoteles, basado en una base de datos relacional y una aplicación de gestión que permita cubrir todas las funcionalidades necesarias.*

*Este trabajo tiene como objetivo diseñar e implementar una base de datos relacional y una aplicación funcional para la gestión integral de reservas en un hotel. Para ello, se definieron las siguientes tareas:*

* *Analizar bases de datos homólogas para identificar fortalezas y debilidades de soluciones existentes.*
* *Diseñar un Diagrama Entidad-Relación que cumpla con los requisitos académicos del curso.*
* *Obtener el modelo relacional mediante la transformación del DER y normalización.*
* *Implementar el modelo mediante código SQL que permita crear, poblar y consultar la base de datos.*
* *Desarrollar una aplicación que permita a clientes, empleados y gerentes interactuar con el sistema según sus funciones.*

*La propuesta se desarrolló considerando los requerimientos funcionales del sistema, las mejores prácticas del diseño de bases de datos y el uso de herramientas modernas para su implementación*

# Capítulo I. Estudio del estado del arte

## Introducción

*Este capítulo se enfoca en la revisión de conceptos fundamentales y el análisis crítico de soluciones existentes relacionadas con sistemas de reservas para hoteles. Se identifican los elementos comunes, carencias y oportunidades de mejora que justifican el desarrollo de la propuesta presentada en este informe.*

# Análisis de los conceptos asociados a su problema.

*Un sistema de reservas hoteleras es una solución informática que permite gestionar el proceso de reserva de habitaciones, registro de clientes, disponibilidad de espacios, pagos y reportes administrativos. Estos sistemas suelen involucrar múltiples actores (clientes, empleados, gerentes) y se basan en estructuras de datos sólidas que garanticen la integridad y disponibilidad de la información.*

*Conceptos como* ***modelo entidad-relación****,* ***normalización****,* ***base de datos relacional****,* ***sistema de gestión de bases de datos (SGBD)****,* ***modelo cliente-servidor*** *y* ***aplicaciones web responsivas*** *resultan esenciales para el entendimiento y desarrollo del sistema. Asimismo, se deben manejar principios como la atomicidad, integridad referencial, y seguridad de la información, pilares fundamentales para la correcta implementación de este tipo de sistemas.*

# Análisis crítico de soluciones homólogas

*Existen en el mercado diversas soluciones comerciales y de código abierto orientadas a la gestión hotelera, como* Hotelogix*,* Cloudbeds *o* Reservio*. Estas plataformas ofrecen funcionalidades completas, desde la reserva en línea hasta el control de ingresos y reportes gerenciales. No obstante, muchas de estas herramientas están en idioma inglés, son de pago o no permiten su adaptación libre a las particularidades de cada hotel.*

*Por otro lado, algunas bases de datos académicas analizadas presentaban estructuras deficientes en cuanto a normalización o carencia de relaciones avanzadas como agregaciones o especializaciones. También se identificó que no todas las soluciones permiten realizar reservas múltiples, gestionar roles diferenciados (cliente, empleado, gerente) o producir reportes personalizados.*

*A partir de este análisis, se concluyó que una solución diseñada desde cero y adaptada al contexto nacional, puede cubrir las brechas detectadas, al tiempo que permite aplicar los conocimientos adquiridos durante la asignatura.*

## Conclusiones parciales

***Se identificaron debilidades en soluciones existentes como limitaciones en la personalización, carencia de control detallado por roles y falta de flexibilidad para adaptarse a necesidades específicas. Estas observaciones sustentan la necesidad y pertinencia del desarrollo de un sistema propio, fundamentado en principios sólidos de diseño de bases de datos y lógica de negocio clara.***

# Capítulo II. Diseño de la propuesta de solución

## Introducción

Este capítulo detalla el proceso de diseño de la base de datos, desde la metodología empleada, hasta la obtención del modelo relacional. Se incluye además la descripción del Diagrama Entidad-Relación (DER), su normalización y transformación al Modelo Relacional (MR), todos ellos alineados a los requerimientos establecidos.

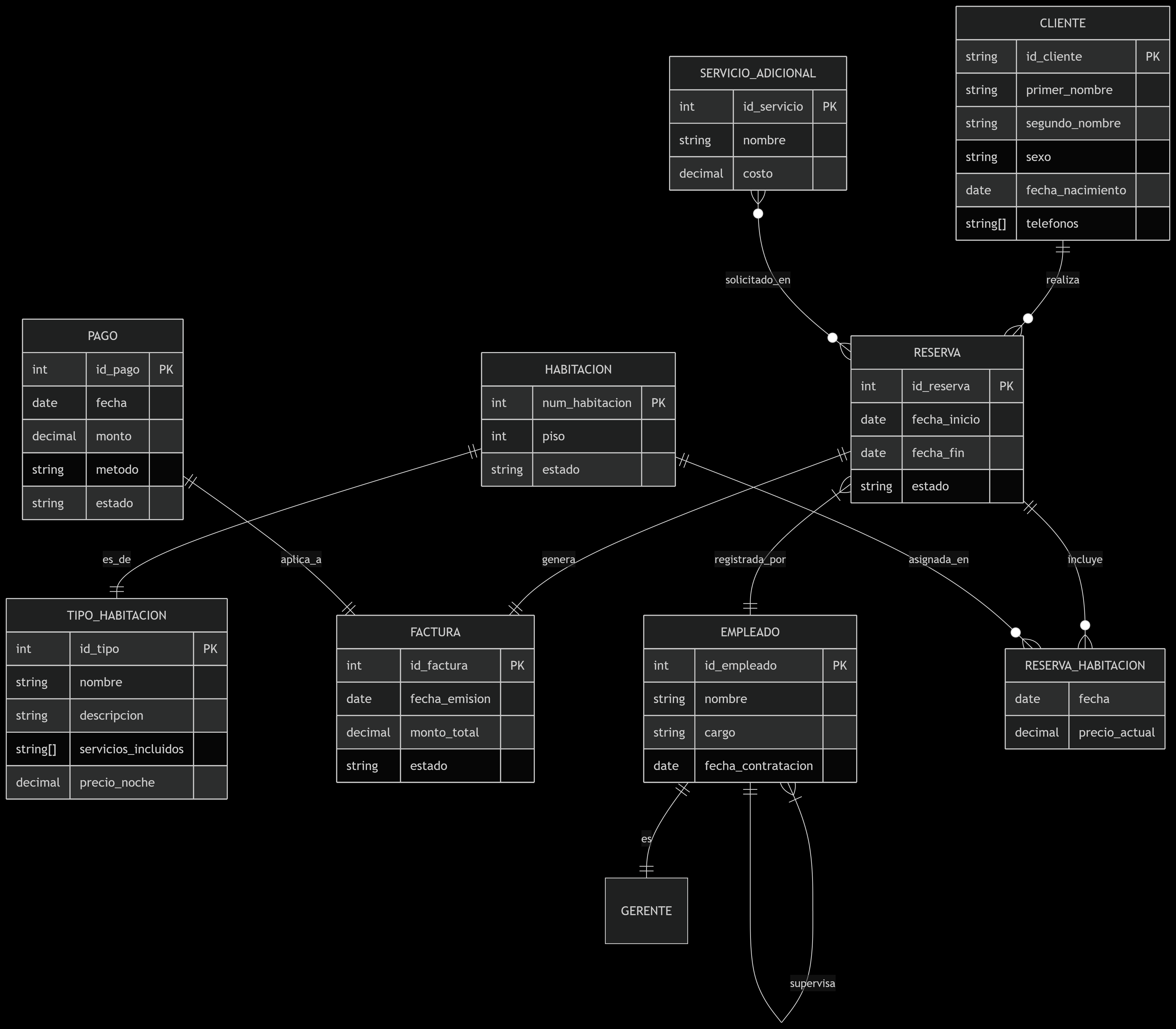
## 2.1 Descripción de la metodología utilizada para el diseño de la base de datos

*Para el diseño de la base de datos se utilizó una metodología en fases, consistente en:*

1. ***Recolección de requisitos:*** *Se definieron los roles (cliente, empleado, gerente) y las funcionalidades del sistema (registro, reserva, gestión de habitaciones, reportes).*
2. ***Identificación de entidades y relaciones:*** *Se determinaron las entidades principales: Cliente, Empleado, Gerente, Habitación, Reserva, entre otras.*
3. ***Modelado conceptual (DER):*** *Se elaboró el DER cumpliendo con las especificaciones académicas: mínimo ocho entidades, atributos variados, relaciones con atributos, relaciones 1:1, 1:N y N:M, y al menos tres extensiones del modelo (entidad débil, especialización, relación recursiva).*
4. ***Normalización:*** *Se aplicaron las formas normales hasta la 3FN, eliminando redundancias y asegurando la integridad.*
5. ***Transformación al modelo lógico:*** *Se obtuvo el modelo relacional aplicando reglas formales desde el DER.*

## 2.2 Modelo Entidad Relación

El Diagrama Entidad-Relación (DER) que se presenta fue construido cumpliendo con los requisitos académicos del curso. Incluye un mínimo de ocho entidades, relaciones con atributos, claves primarias y foráneas, así como extensiones como entidades débiles, especialización y relaciones recursivas. Este diagrama representa de forma conceptual la estructura del sistema de reservas hoteleras y es la base para la posterior normalización y obtención del modelo relacional.

**

## 2.3 Normalización de la base de datos

*Se aplicó el proceso de normalización desde la* ***Primera Forma Normal (1FN)*** *hasta la* ***Tercera Forma Normal (3FN)****, con el objetivo de eliminar redundancias, evitar anomalías de inserción, modificación y eliminación, y garantizar la integridad de los datos en el sistema.*

*A continuación, se describen algunas decisiones representativas:*

* ***1FN****: Se eliminaron atributos multivaluados como telefonos[] del cliente y servicios\_incluidos[] del tipo de habitación. Se crearon las entidades auxiliares TELEFONO\_CLIENTE y SERVICIO\_INCLUIDO para manejar estas relaciones 1:N de manera independiente.*
* ***2FN****: Se identificaron relaciones con claves primarias compuestas, como RESERVA\_HABITACION, y se aseguraron que todos los atributos dependieran completamente de la clave. Aquellos que dependían parcialmente de uno de los atributos fueron trasladados a nuevas entidades.*
* ***3FN****: Se removieron dependencias transitivas, garantizando que todos los atributos no clave dependieran únicamente de la clave primaria. Por ejemplo, en la tabla CLIENTE, la edad fue eliminada como atributo almacenado y tratada como derivado, calculado a partir de la fecha de nacimiento.*

*Este proceso permitió mejorar la estructura del sistema, facilitar su mantenibilidad, y asegurar la integridad de la información almacenada.*

## 2.4 Modelo Relacional CLIENTE(id\_cliente PK, primer\_nombre, segundo\_nombre, sexo, fecha\_nacimiento, tipo\_cliente) TELEFONO\_CLIENTE(id\_cliente FK, telefono) EMAIL\_CLIENTE(id\_cliente FK, email) HABITACION(num\_habitacion PK, piso, estado, disponible, id\_tipo FK) TIPO\_HABITACION(id\_tipo PK, nombre, descripcion) SERVICIOS\_TIPO(id\_tipo FK, servicio) RESERVA(id\_reserva PK, fecha\_inicio, fecha\_fin, total\_estadia, id\_cliente FK, id\_empleado FK) RESERVA\_HABITACION(id\_reserva FK, num\_habitacion FK, fecha\_especifica, estado) FACTURA(id\_factura PK, fecha\_emision, monto\_total, id\_reserva FK) PAGO(id\_pago PK, fecha, monto, metodo, id\_factura FK) SERVICIO\_ADICIONAL(id\_servicio PK, nombre, costo) RESERVA\_SERVICIO(id\_reserva FK, id\_servicio FK) EMPLEADO(id\_empleado PK, nombre, cargo, supervisor\_id FK) GERENTE(id\_gerente PK, area\_responsable)

*El modelo relacional fue construido a partir del Diagrama Entidad-Relación (DER) elaborado con la herramienta* ***dbdiagram.io****, que permitió modelar gráficamente las entidades, relaciones y atributos necesarios para el sistema de reservas hoteleras.*

*Para la transformación del* ***modelo conceptual (MER)*** *al* ***modelo lógico (MR)*** *se siguieron las siguientes consideraciones:*

* ***Cada entidad fuerte*** *del DER se convirtió en una tabla relacional. Por ejemplo, la entidad CLIENTE pasó a ser la tabla CLIENTE, manteniendo sus atributos principales.*
* ***Las relaciones N:M*** *fueron transformadas en tablas intermedias con claves foráneas. Un ejemplo es la tabla RESERVA\_HABITACION, que vincula las entidades RESERVA y HABITACION.*
* ***Las relaciones con atributos*** *fueron modeladas como tablas independientes. RESERVA\_HABITACION contiene el atributo fecha\_especifica que forma parte de la relación.*
* ***Atributos derivados*** *como edad o disponible no se almacenan físicamente, sino que se calculan dinámicamente.*
* ***Relaciones recursivas*** *como la supervisión entre empleados (EMPLEADO supervisa EMPLEADO) se representaron mediante claves foráneas a la misma tabla.*
* ***Especialización/Generalización*** *fue implementada entre GERENTE y EMPLEADO, manteniendo la integridad entre ambas con una relación 1:1.*
* ***Entidades débiles*** *como SERVICIO\_INCLUIDO dependen completamente de su entidad fuerte (TIPO\_HABITACION) y así se representó mediante claves foráneas y restricciones ON DELETE CASCADE.*

*La transformación fue guiada por las reglas formales de diseño y normalización, cumpliendo con las mejores prácticas para el desarrollo de bases de datos robustas y escalables.*

## Conclusiones parciales

***El diseño de la base de datos cumplió con los requisitos académicos y funcionales. El modelo entidad-relación propuesto es robusto, flexible y escalable, y fue transformado exitosamente a un modelo relacional normalizado, listo para su implementación.***

# Capítulo III. Implementación de la base de datos

## Introducción

Este capítulo describe el proceso de implementación técnica de la base de datos diseñada para el sistema de reservas hoteleras. Incluye una descripción de las herramientas seleccionadas, los comandos del lenguaje de definición y manipulación de datos, las funciones implementadas y la aplicación construida para la interacción con el sistema, garantizando una gestión eficiente y segura de la información.

## 3.1 Herramientas seleccionadas

Para la implementación del sistema se utilizaron las siguientes herramientas:

* **PostgreSQL 17.5**: sistema de gestión de bases de datos relacional, robusto y de código abierto.
* **pgAdmin 4**: interfaz gráfica para la administración y visualización de bases de datos PostgreSQL.
* **Visual Studio Code**: editor de código versátil y multiplataforma utilizado para programar la aplicación.
* **Aplicación en Java**: se desarrolló una interfaz por línea de comandos (CLI) para interactuar con la base de datos de forma segura, realizando operaciones CRUD, consultas y ejecución de funciones.

## 3.2 Lenguaje de Definición de Datos

El modelo relacional fue implementado mediante comandos SQL que crearon las distintas tablas y restricciones necesarias. A continuación, se resumen algunos de los objetos creados:  
- Tablas: CLIENTE, TELEFONO\_CLIENTE, HABITACION, TIPO\_HABITACION, RESERVA, EMPLEADO, USUARIO, etc.  
- Relaciones y restricciones: claves primarias, claves foráneas, restricciones CHECK, UNIQUE, ENUM y not null.  
- Tipos derivados calculados como edad del cliente, estado de la habitación, etc.  
- Vistas: vista\_clientes\_por\_pais, vista\_disponibilidad\_detallada.  
Todos estos objetos fueron generados con herramientas CASE y afinados manualmente para ajustarse a las reglas de normalización y lógica de negocio.

## 3.3 Lenguaje de Manipulación de Datos

### 3.3.1 Consideraciones para poblar la base de datos

La base de datos fue poblada utilizando scripts SQL automatizados combinados con datos sintéticos generados con la herramienta Mockaroo. Se insertaron aproximadamente 50,000 tuplas por tabla, de las cuales al menos 3,000 fueron diseñadas con coherencia lógica para simular el comportamiento real del sistema. Estas tuplas representan reservas válidas, usuarios con roles definidos, habitaciones asignadas correctamente, y pagos relacionados con facturación.

### 3.3.2 Principales consultas que responden a la lógica del negocio.

A continuación se presentan algunas de las principales consultas implementadas:  
1. Insertar nueva reserva: `INSERT INTO RESERVA ...`  
2. Consultar disponibilidad de habitaciones: `SELECT \* FROM vista\_disponibilidad\_detallada`  
3. Actualizar estado de una reserva: `UPDATE RESERVA SET estado = 'Cancelada' WHERE id\_reserva = ?`  
4. Eliminar un cliente inactivo: `DELETE FROM CLIENTE WHERE id\_cliente = ?`  
5. Buscar clientes frecuentes: `SELECT fn\_obtener\_huesped\_frecuente()`  
6. Consulta para panel de gerente: `SELECT COUNT(\*), SUM(p.monto) ...`

## 3.4 Principales funciones implementadas

*Se desarrollaron funciones específicas que responden a los reportes y necesidades del sistema:  
1. `fn\_ingresos\_por\_tipo` – calcula ingresos generados por cada tipo de habitación.  
2. `fn\_obtener\_huesped\_frecuente` – identifica al cliente con más estancias.  
3. `fn\_calcular\_ocupacion` – evalúa el porcentaje de ocupación diaria.  
4. `fn\_servicios\_mas\_solicitados` – devuelve los servicios adicionales más utilizados.*

## 3.5 Implementación de la aplicación

*La aplicación fue implementada como una* ***interfaz de línea de comandos (CLI)*** *utilizando el lenguaje de programación Java.  
Las principales funcionalidades incluyen:*

* *Autenticación de usuarios por roles (Cliente, Empleado, Gerente).*
* *Flujos de menú según el rol autenticado.*
* *Operaciones CRUD sobre reservas, habitaciones, clientes y pagos.*
* *Consulta de disponibilidad y generación de reportes administrativos.*

*La comunicación con la base de datos se realiza directamente mediante* ***JDBC****, garantizando integridad y seguridad en la manipulación de los datos.*

***Ejemplos de funcionalidades implementadas en la CLI****:*

1. *Crear nueva reserva*
2. *Consultar habitaciones disponibles por fecha*
3. *Cancelar reserva existente*
4. *Registrar nuevo cliente*
5. *Ver ingresos por tipo de habitación*

# Conclusiones parciales

*La implementación técnica del sistema permitió validar la efectividad del modelo relacional propuesto, garantizando su operatividad bajo distintos escenarios del negocio hotelero. Las herramientas seleccionadas demostraron ser idóneas para el desarrollo completo. Las funciones y consultas implementadas cumplen con los requerimientos del negocio, y la aplicación desarrollada permite una gestión completa de clientes, habitaciones, reservas y reportes administrativos.*

# Conclusiones generales

1. Se logró diseñar e implementar un sistema de reservas hoteleras eficiente, cumpliendo con los objetivos funcionales y académicos propuestos.

2. El modelo de base de datos relacional implementado permite una gestión estructurada, con integridad de datos y bajo nivel de redundancia.

3. Las herramientas utilizadas resultaron apropiadas para el desarrollo del sistema, facilitando tanto la creación como la gestión de la base de datos.

4. El desarrollo de funciones, vistas y consultas específicas aportó valor al análisis del negocio y la toma de decisiones.

*5. La aplicación final ofrece una interfaz intuitiva y funcional que permite la interacción por parte de los distintos usuarios del sistema (clientes, empleados y gerentes).*

# Recomendaciones

***Se recomienda incorporar un sistema de autenticación multifactor (MFA) para mejorar la seguridad del acceso de los usuarios, especialmente los perfiles administrativos (gerentes y empleados).***

* ***Sería útil desarrollar un módulo de estadísticas avanzadas, que integre visualizaciones gráficas más detalladas sobre ocupación, ingresos y comportamiento de los clientes, utilizando herramientas como Chart.js o Recharts.***
* ***Se sugiere implementar un historial de auditoría más completo, extendiendo la tabla AUDITORIA para registrar cambios en entidades críticas como CLIENTE, HABITACION y PAGO, permitiendo mayor trazabilidad y cumplimiento normativo.***
* ***A mediano plazo, se puede considerar la integración con pasarelas de pago (como PayPal o Stripe), lo que permitiría a los clientes realizar pagos electrónicos directamente desde la aplicación.***
* ***Se propone desarrollar una versión móvil de la aplicación, optimizada para Android/iOS, que permita a los usuarios realizar reservas, consultas y gestiones desde sus dispositivos personales.***
* ***También sería recomendable incorporar un módulo de retroalimentación del cliente, mediante encuestas posteriores a su estancia, lo cual ayudaría a mejorar la calidad del servicio ofrecido por el hotel.***

# Bibliografía Consultada

***ELMASRI, R.; NAVATHE, S. B.* Fundamentals of Database Systems*. 7th ed. Boston: Pearson, 2016. 1272 p.***

* ***DATE, C. J.* Database Design and Relational Theory: Normal Forms and All That Jazz*. 2nd ed. Sebastopol: O'Reilly Media, 2019. 496 p.***
* ***SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S.* Database System Concepts*. 7th ed. New York: McGraw-Hill Education, 2020. 1376 p.***
* ***CONNOLLY, T.; BEGG, C.* Database Systems: A Practical Approach to Design, Implementation, and Management*. 6th ed. London: Pearson, 2015. 1408 p.***
* ***CULNAN, M. J. An analysis of the information usage patterns of academics and practitioners in computer field: A citation analysis of a national conference proceedings.* Information Processing and Management*, 1978, 14(6): p. 395–404.***
* ***YUAN, A.; JEANNETTE, J., et al. Characterizing and Mining the Citation Graph of the Computer Science Literature. [En línea].* Knowledge and Information Systems*, 2004, vol. 6. [Consultado el: 12 de enero de 2010] 664–678 p. Disponible en:*** [***http://dx.doi.org/10.1007/s10115-003-0128-3***](http://dx.doi.org/10.1007/s10115-003-0128-3)
* ***HELLERSTEIN, J. M.; STONEBRAKER, M.; HAMILTON, J. Architecture of a Database System.* Foundations and Trends in Databases*, 2007, 1(2): p. 141–259.***
* ***RAGHAVAN, P.; RAJARAMAN, A.; ULLMAN, J. D.* Mining of Massive Datasets*. 3rd ed. Cambridge: Cambridge University Press, 2020. 496 p.***
* ***POSTGRESQL GLOBAL DEVELOPMENT GROUP.* PostgreSQL Documentation*. [En línea]. 2024. [Consultado: 10 de junio de 2025]. Disponible en:*** [***https://www.postgresql.org/docs/***](https://www.postgresql.org/docs/)