**PROYECTO FINAL BASES DE DATOS**

**Escuela Politécnica Nacional**

**Francis Guaman**

**Lenin Taco**

**Misael García**

**05-02-2025**

Tabla de contenido

[Resumen 3](#_Toc189610594)

[Introducción 3](#_Toc189610595)

[Metodología del Proyecto 3](#_Toc189610596)

[Modelado de la Base de Datos 3](#_Toc189610597)

[Seguridad y Control de Acceso 3](#_Toc189610598)

[Respaldo y Recuperación de Datos 3](#_Toc189610599)

[Optimización y Rendimiento de Consultas 3](#_Toc189610600)

[Monitoreo y Pruebas 3](#_Toc189610601)

[Resultados y Análisis 3](#_Toc189610602)

[Conclusiones 3](#_Toc189610603)

[Recomendaciones 3](#_Toc189610604)

[Anexos 3](#_Toc189610605)

## Resumen

El presente proyecto tiene como objetivo el diseño, implementación y optimización de una base de datos eficiente y segura para un sistema de reservas hoteles. A lo largo de las distintas fases, se abordaron actividades clave como el modelado de la base de datos, la definición de restricciones de integridad, la optimización de consultas, y la implementación de mecanismos de seguridad, auditoría y respaldos. Además, se integraron procedimientos almacenados, vistas y triggers para mejorar la eficiencia operativa. Finalmente, se aplicaron técnicas de monitoreo y pruebas de carga para garantizar un alto rendimiento y escalabilidad del sistema, mientras se utilizó Git para el control de versiones y la automatización de pruebas. El proyecto proporciona una solución robusta y escalable para la gestión de datos en un entorno de alta demanda.

## Introducción

El presente proyecto tuvo como finalidad el desarrollo de una base de datos integral para la gestión eficiente de un sistema de reservas de hoteles. A través de un enfoque metodológico, se abarcan aspectos fundamentales como el diseño de un modelo de base de datos bien estructurado, la aplicación de medidas de seguridad y control de acceso, así como la optimización del rendimiento mediante consultas eficientes e índices estratégicos. La implementación de auditoría, respaldos y técnicas avanzadas de recuperación de datos garantiza la continuidad y disponibilidad del sistema.

## Metodología del Proyecto

Este proyecto se desarrolló siguiendo un enfoque estructurado y dividido en varias etapas clave, con el objetivo de diseñar e implementar una base de datos eficiente y escalable para un sistema de reservas de hoteles. A continuación, se describen las etapas del desarrollo.

**Planificación y Análisis de Requerimientos**

* Definir objetivos y requisitos del sistema.
* Identificar entidades clave (Clientes, Vuelos, Reservas, Pagos).
* Determinar requisitos de seguridad, rendimiento y escalabilidad.

**Diseño del Modelo de Base de Datos**

* Crear el modelo ER y definir las relaciones entre las entidades.
* Diseñar el modelo lógico y físico de la base de datos.
* Crear el diccionario de datos, documentando tablas, campos y relaciones.

**Seguridad y Control de Acceso**

* Implementar roles y permisos de usuario para controlar acceso.
* Cifrar datos sensibles y configurar auditoría.
* Establecer políticas de seguridad y acceso.

**Respaldo y Recuperación de Datos**

* Crear respaldos completos e incrementales.
* Configurar respaldos en caliente para no interrumpir el servicio.
* Establecer estrategias de recuperación ante desastres.

**Optimización del Rendimiento**

* Crear índices para mejorar la velocidad de consultas.
* Optimizar consultas SQL y realizar particionamiento de tablas.
* Monitorear y optimizar el uso de recursos.

**Procedimientos, Vistas y Triggers**

* Implementar procedimientos almacenados para centralizar lógica.
* Crear vistas para simplificar consultas complejas.
* Usar triggers para auditoría y control de cambios.

**Monitoreo de Recursos**

* Monitorear el rendimiento de la base de datos y realizar pruebas de carga.
* Optimizar índices y recursos según el tipo de consulta.

**Pruebas y Validación**

* Realizar pruebas funcionales, de carga y de seguridad.
* Validar que todos los procesos funcionen correctamente.

**Implementación y Monitoreo Continuo**

* Desplegar la base de datos en producción.
* Monitorear su desempeño y hacer ajustes según sea necesario.

## Modelado de la Base de Datos

El modelado de la base de datos para el sistema de reservas de hoteles se desarrolló en tres niveles: conceptual, lógico y físico, asegurando un diseño eficiente y escalable. A continuación, se describen cada una de estas etapas, junto con las herramientas utilizadas y la justificación del modelo seleccionado.

**Diseño Conceptual**

Se elaboró un diagrama entidad-relación (ER) para identificar las principales entidades y sus relaciones.

Entidades: Clientes, Habitaciones, Reservas y Pagos.

Relaciones: Un cliente puede hacer múltiples reservas.

Cada reserva está vinculada a una habitación específica.

Los pagos están asociados con reservas realizadas.

**Atributos Clave**

Cliente: ClienteID, nombre, email, teléfono.

Habitación: HabitacionID, tipo, capacidad, precio.

Reserva: ReservaID, fecha reserva, estado.

Pago: PagoID, monto, fecha de pago, método de pago.

**Diseño Lógico**

El modelo lógico tradujo el diseño conceptual a una estructura relacional, definiendo tablas, atributos, claves primarias y relaciones entre tablas.

* Tablas
* Claves primarias
* Claves foráneas

**Diseño Físico**

El diseño físico consistió en la implementación del esquema en el sistema de gestión de bases de datos MySQL.

**Índices**

Índices en las columnas HabitacionID, ClienteID y ReservaID para mejorar el rendimiento de las consultas.

## Seguridad y Control de Acceso

**Implementación de Políticas de Seguridad**

* Creación de usuarios con permisos específicos (admin\_user, empleado\_user, auditor\_user).
* Asignación de privilegios según el rol, restringiendo acceso a tablas sensibles.
* Uso de GRANT para gestionar accesos y WITH GRANT OPTION para delegar permisos.

**Cifrado de Datos Sensibles**

* Implementación de AES\_ENCRYPT para proteger datos como contraseñas.
* Uso de claves de cifrado seguras de 16 bytes para evitar advertencias de MySQL.
* Recuperación de datos mediante AES\_DECRYPT.

**Registro de Auditoría**

* Creación de la tabla Auditoria para registrar cambios en la base de datos.
* Implementación de triggers (AFTER INSERT, AFTER UPDATE) para monitorear cambios en la tabla Reserva.
* Registro automático de usuario y tipo de acción realizada en la base de datos.

## Respaldo y Recuperación de Datos

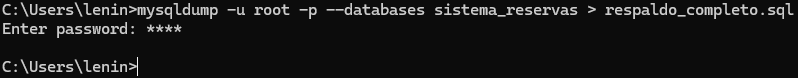
La integridad y disponibilidad de los datos en un sistema de reservas hoteleras son aspectos fundamentales para garantizar su correcto funcionamiento. Para ello, se implementaron diversas estrategias de respaldo y recuperación de datos que permiten minimizar la pérdida de información ante fallos inesperados y optimizar el uso de recursos.

**Creación de Respaldos Completos**

Se realizaron respaldos completos de la base de datos utilizando la herramienta mysqldump. Este proceso permite generar copias exactas de la información almacenada, asegurando que en caso de una falla crítica, toda la base de datos pueda ser restaurada en su totalidad.

**Estrategia aplicada:**

* Se sugiere respaldos completos de la base de datos una vez al día mediante el siguiente comando: 
* Luego se coloca la clave y si no sale algún error, se vería de la siguiente manera.



**Importancia:** Los respaldos completos garantizan la restauración total de los datos, siendo una medida de seguridad esencial en caso de corrupción de archivos o fallos del sistema.

**Configuración de Respaldos Incrementales**

Para optimizar el almacenamiento y reducir el tiempo de respaldo, se configuraron respaldos incrementales. Este tipo de respaldo guarda únicamente los cambios realizados desde el último respaldo completo o incremental, lo que permite una recuperación más eficiente.

**Estrategia aplicada:**

* Se recomienda realizar respaldos incrementales cada cuatro horas utilizando binlog en MySQL con el siguiente procedimiento:

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente

* Para crear el respaldo incremental, se debe colocar la información obtenida con show master status de la siguiente manera:



**Importancia:** Los respaldos incrementales optimizan el uso de recursos y reducen los tiempos de recuperación, permitiendo restaurar solo los datos modificados en lugar de toda la base de datos.

**Implementación de Respaldos en Caliente (Hot Backups)**

Los respaldos en caliente buscan garantizar la disponibilidad continua del sistema y la integridad de los datos sin interrumpir la operación del negocio, realizando copias de seguridad de la base de datos mientras sigue en uso, asegurando que las reservas y transacciones en curso no se vean interrumpidas.

**Estrategia aplicada:**

* Se recomienda realizar respaldos en caliente diarios en horarios de menor actividad para reducir el impacto en el rendimiento.
* Para realizar el respaldo en caliente, se utilizó **mysqldump** con la opción --single-transaction, lo que garantiza la consistencia del respaldo sin interrumpir las operaciones de la base de datos.
* El comando utilizado para realizar el respaldo en caliente fue el siguiente:



**Importancia:** Los respaldos en caliente son esenciales para bases de datos de producción donde la interrupción del servicio no es viable. Permiten una recuperación rápida y eficiente sin afectar la disponibilidad del sistema.

## Optimización y Rendimiento de Consultas

La eficiencia en la recuperación de datos es crucial para un sistema de reservas de hotel, dado el manejo continuo de operaciones como la consulta de disponibilidad de habitaciones, las reservas activas y el historial de pagos. Para lograr un sistema rápido y escalable, se implementaron diversas técnicas de optimización.

Se realizaron pruebas de extracción de información con y sin índices lo que mostro como se puede reducir el tiempo de consulta cuando usamos índices.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig 1 Consulta datos sin índices**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig 2 Consulta datos con indices**

Estos índices trajeron beneficios importantes, como una notable aceleración en la búsqueda de registros específicos, así como una mejora significativa en el rendimiento de las consultas entre las tablas, contribuyendo al rendimiento general del sistema.

**Optimización de Consultas SQL**

La optimización de consultas fue fundamental para garantizar el rendimiento eficiente del sistema de reservas. Se mejoraron las consultas más críticas, como la verificación de disponibilidad de habitaciones y la obtención del historial de reservas de los clientes. Para determinar las mejores estrategias, se utilizó la instrucción EXPLAIN, que permitió identificar cuellos de botella y ajustar el plan de ejecución de las consultas.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Fig 3 Consulta con índices**

Captura de pantalla de computadora

Descripción generada automáticamente

**Fig 4 Consulta con y sin índices**

**Estrategias de Particionamiento**

Para manejar eficientemente el creciente volumen de datos en el sistema de reservas, se evaluó el particionamiento de la tabla Reservas por año. Esta estrategia permite que las consultas se ejecuten sobre subconjuntos de datos, mejorando el tiempo de respuesta al limitar el escaneo de registros innecesarios.

Texto, Tabla

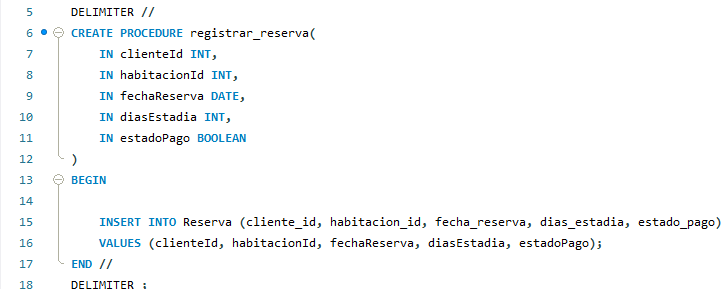
Descripción generada automáticamente

**Fig. 5 Partición de tabla reserva**

## Automatización y Procedimientos

**Procedimientos Almacenados, Vistas y Trigger**

Se implementaron procedimientos almacenados y triggers para automatizar tareas clave en el sistema de reservas. Uno de los procedimientos principales registrar nueva reserva, pidiendo todos los datos necesarios como servicios especiales solicitados por los clientes. Esto permitió centralizar la lógica de negocio y reducir la carga del lado del cliente.



**Fig. 6 Creación de un Procedimiento Almacenado**

Los triggers fueron utilizados para mantener la integridad de los datos y realizar auditoría.

Imagen que contiene Texto

Descripción generada automáticamente

**Fig. 7 Creación de un Trigger**

Se crearon vistas para simplificar y optimizar el acceso a datos complejos en el sistema de reservas.

Texto

Descripción generada automáticamente

**Fig. 8 Creación de una Vista**

## Monitoreo y Pruebas

**Monitoreo del Rendimiento de Consultas**

* Uso de SHOW PROCESSLIST para detectar consultas lentas.
* Aplicación de EXPLAIN para analizar el plan de ejecución de consultas.
* Optimización de consultas mediante índices y ajustes en JOIN y subconsultas.
* Pruebas de Carga y Rendimiento
* Uso de herramientas como Apache JMeter para simular múltiples usuarios concurrentes.
* Evaluación del tiempo de respuesta de la base de datos bajo alta carga.
* Ajustes en índices y consultas según los resultados obtenidos.

**Optimización de Índices y Recursos**

* Identificación y eliminación de índices no utilizados.
* Creación de índices en columnas clave para mejorar el rendimiento.
* Evaluación del impacto de los índices en operaciones de lectura y escritura.

## Resultados y Análisis

El desarrollo del sistema de reservas para el hotel resultó en una solución robusta, eficiente y escalable, capaz de cubrir las necesidades operativas clave. A lo largo del proyecto, se obtuvieron importantes mejoras en múltiples áreas, desde el diseño de la base de datos hasta la seguridad y el rendimiento.

La creación de un modelo de base de datos bien estructurado permitió una gestión eficiente de entidades como clientes, habitaciones, reservas y pagos, garantizando la integridad y consistencia de los datos. El uso de restricciones referenciales y la optimización de relaciones entre tablas facilitaron consultas rápidas y confiables.

Se aplicaron técnicas avanzadas de seguridad, como el cifrado de datos sensibles y la implementación de roles y permisos, lo que contribuyó a proteger la información confidencial del sistema. La habilitación de auditorías permitió registrar y monitorear actividades críticas, mejorando el control del sistema.

En el ámbito de la optimización del rendimiento, la gestión de índices y la automatización de consultas mediante procedimientos almacenados demostraron ser fundamentales para acelerar operaciones frecuentes. El particionamiento de tablas ayudó a manejar grandes volúmenes de datos, asegurando la escalabilidad del sistema.

La integración de estrategias de respaldo y recuperación de datos garantizó la disponibilidad continua del sistema ante posibles fallos.

Este proyecto consolidó una solución efectiva para la gestión de reservas de hotel, brindando una base sólida para futuras mejoras y expansiones.

## Conclusiones

* El desarrollo de un sistema de reservas hoteleras bien estructurado demostró la importancia de un modelado de base de datos eficiente, que garantiza la integridad y disponibilidad de la información, permitiendo una gestión óptima de clientes, reservas y pagos.
* Las estrategias de optimización, como la creación de índices, procedimientos almacenados y particionamiento de tablas mejoraron significativamente el rendimiento del sistema, permitiendo consultas rápidas y manejo eficiente de grandes volúmenes de datos.
* La implementación de medidas de seguridad, auditoría y respaldo de datos contribuyó a garantizar la protección de información sensible y la continuidad del servicio, factores esenciales para un sistema de alta disponibilidad.

## Recomendaciones

* Implementar mejoras futuras como la integración de inteligencia de negocio para analizar patrones de reservas y optimizar la gestión hotelera.
* Continuar monitoreando el rendimiento del sistema mediante herramientas avanzadas de monitoreo y auditoría para detectar y resolver cuellos de botella de manera proactiva.
* Expandir las funcionalidades del sistema incorporando integraciones con plataformas de pago externas y motores de reservas en línea para brindar una experiencia más completa y competitiva.

## Anexos