DESARROLLO BASE DE DATOS: RESERVACIONES

YESI ESTEBAN PANTOJA CUELLAR

DOCENTE

BRAYAN IGNACIO ARCOS BURBANO

INSTITUTO TECNOLÓGICO DEL PUTUMAYO

TECNOLOGÍA EN DESARROLLO DE SOFTWARE

QUINTO SEMESTRE

DESARROLLO DE BASE DE DATOS

MOCOA - PUTUMAYO

2023

RESUMEN

El objetivo principal del proyecto es crear una base de datos para optimizar el proceso de reservaciones, mejorando la eficiencia y precisión en la gestión de reservas, clientes, empleados y métodos de pago. Esta base de datos busca resolver problemas identificados en el sistema actual y proporcionar una solución más robusta y organizada.

La base de datos está diseñada con varias tablas clave que incluyen información sobre tipos de documentos, métodos de pago, empleados, clientes, reservas y el historial de reservas. Se han establecido relaciones entre estas tablas mediante claves foráneas para garantizar la integridad de los datos y evitar redundancias.

La implementación incluye la creación de tablas y la aplicación de principios de normalización para asegurar que los datos estén bien organizados y sean fáciles de gestionar. Se ha prestado especial atención a la eficiencia en la consulta de datos y la integridad referencial para garantizar un rendimiento óptimo del sistema.

El diseño de la base de datos tiene como resultado una solución más eficaz para la gestión de reservaciones, que debe ser evaluada en función de su capacidad para resolver los problemas del sistema anterior. Se recomienda continuar con pruebas y ajustes para asegurar que el sistema cumpla con sus objetivos y mejore el proceso de reservas de manera significativa.

INTRODUCCIÓN

Este informe tiene como principal objetivo la **realización y documentación de una base de datos**, la cual se desarrollará bajo los conceptos clave vistos en clase. A lo largo del informe se describirá el proceso de diseño y creación de la base de datos, incluyendo la definición de **tablas**, la estructura de los datos y la manera en que se relacionan entre sí.

Se abordarán aspectos fundamentales como:

- La **creación de tablas** y su estructura adecuada.
- El uso de **Primary Key** y **Foreign Key** para definir las relaciones entre las tablas, asegurando la **integridad referencial**.
- La realización de **consultas** que permitan extraer y manipular datos de manera eficiente.

Además de los temas mencionados, el informe también proporcionará una visión del proceso de normalización, con el fin de reducir redundancias y asegurar la consistencia de los datos.

Por último, se presentarán **conclusiones** y **recomendaciones** basadas en los resultados obtenidos, destacando las posibles mejoras y futuras ampliaciones para la base de datos.

METODOLOGÍA

Herramientas:

- MySQL
- Draw.io

Procedimientos:

A continuación, se detallan los métodos y pasos seguidos para llevar a cabo el análisis y la construcción de la base de datos, asegurando que el proceso se alineara con los conceptos aprendidos en clase.

1. Análisis de Requerimientos

- Objetivo: Identificar las necesidades del sistema y los tipos de datos que se gestionarán en la base de datos.
- Descripción: Se realizó un análisis del caso práctico para definir los tipos de entidades, las relaciones entre ellas y los atributos necesarios para cada una. Este proceso fue fundamental para la creación de un modelo de datos eficiente.

2. Diseño del Modelo Entidad-Relación (E-R)

- Objetivo: Representar gráficamente las entidades y sus relaciones.
- Descripción: A partir del análisis de requerimientos, se diseñó un diagrama E-R que incluía las principales entidades (como clientes, reservaciones, y métodos de pago), así como las relaciones entre ellas. En este paso se identificaron las Primary Key y Foreign Key que se utilizarían posteriormente.

3. Normalización de la Base de Datos

- Objetivo: Eliminar redundancias y asegurar la integridad de los datos.
- Descripción: Se aplicaron las tres primeras formas normales (1NF, 2NF, 3NF) para asegurar que cada tabla contenía únicamente datos relacionados, evitando la duplicación de información y asegurando la consistencia de los datos en la base de datos.

4. Creación de Tablas en SQL

- Objetivo: Implementar el modelo E-R en un sistema de gestión de bases de datos.
- Descripción: Usando el lenguaje SQL, se crearon las tablas necesarias, especificando sus Primary Key y Foreign Key correspondientes. Además, se definieron los tipos de datos y las restricciones de cada columna.

5. Población de Datos

- Objetivo: Ingresar datos reales o de prueba en las tablas.
- Descripción: Una vez creadas las tablas, se insertaron datos de prueba utilizando sentencias INSERT INTO, para verificar que el modelo funcionaba correctamente.

6. Consultas SQL

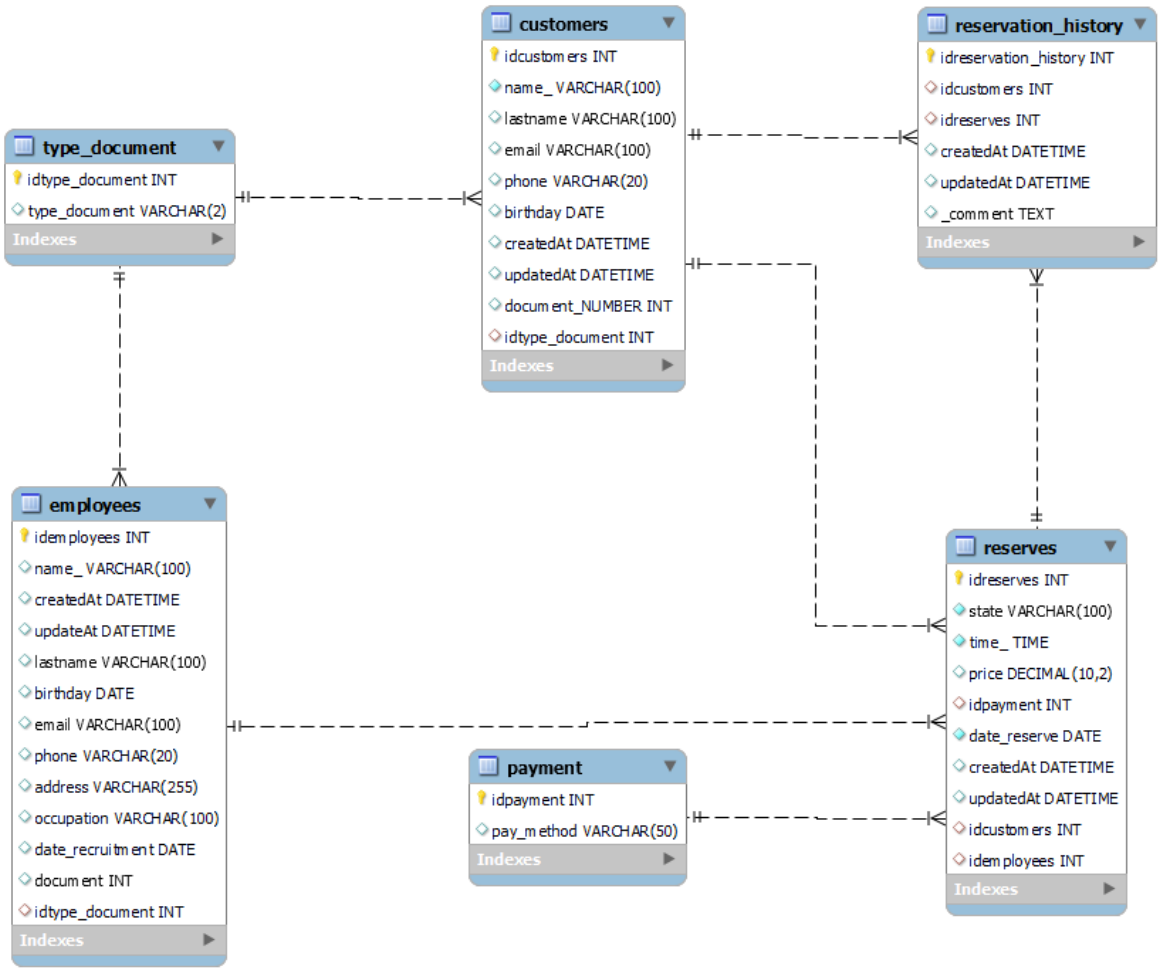
- Objetivo: Extraer información relevante para el análisis y pruebas del sistema.
- Descripción: Se implementaron consultas básicas y avanzadas para validar que la base de datos respondiera correctamente a las preguntas planteadas. Entre las consultas realizadas se incluyeron:
  - Búsquedas por cliente.
  - Reservaciones en un rango de fechas.
  - Análisis de métodos de pago utilizados.

DESARROLLO DE LA BASE DE DATOS

Esta base de datos está diseñada para resolver un problema que he identificado en el proceso de reservaciones, el cual presenta varias dificultades en su versión actual. El sistema actual es lento, con problemas para manejar correctamente la información de las reservaciones, lo que genera errores y confusiones entre los clientes y el personal encargado.

El objetivo principal de esta base de datos es:

- **Mejorar la eficiencia** al realizar una reservación, reduciendo los tiempos de respuesta y facilitando el acceso a la información.
- **Centralizar la información** para evitar duplicados y errores en los datos de clientes, reservaciones y métodos de pago.
- **Automatizar procesos** que anteriormente eran manuales, como la verificación de la disponibilidad en fechas y la asignación de recursos.

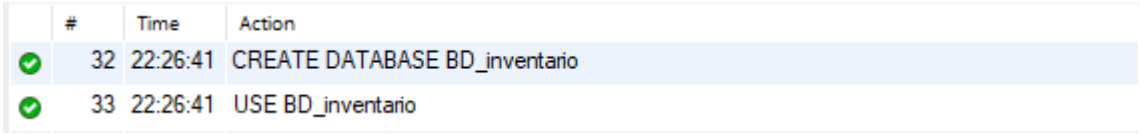


1. **CREACIÓN DE LA BASE DE DATOS:**

- Primero, comenzamos creando y seleccionando la base de datos que utilizaremos.

```
#Creacion de la base de datos
#Paso 1

CREATE DATABASE BD_inventario;

#Paso 2

USE BD_inventario;

#Creacion de las tablas
#Paso 3
```

- Asegurándonos de no recibir errores, la base de datos está lista para su uso.



#	Time	Action
32	22:26:41	CREATE DATABASE BD_inventario
33	22:26:41	USE BD_inventario

CREACIÓN DE TABLAS:

- A continuación, procedemos a crear la tabla type_document, con la cuál identificaremos el tipo de documento de cada persona ya sea cédula de ciudadanía, cédula de extranjería entre otros, usamos un varchar 2 que nos permite agregar la abreviación de dichos documentos

```
CREATE TABLE type_document(
idtype_document INT PRIMARY KEY AUTO_INCREMENT,
type_document VARCHAR(2)
);
```

- Luego tenemos la tabla payment, esta se enfoca en los tipos de métodos de pagos, ya sea, en efectivo o transferencia, entre otras muchas más

```sql
CREATE TABLE payment(
    idpayment INT AUTO_INCREMENT UNIQUE,
    pay_method VARCHAR(50),
    PRIMARY KEY(idpayment)
);
```

- De ahí seguimos con la tabla employees, dónde se guardará la información de los empleados y usaremos la primera foreing key que será el tipo de documento.

```sql
CREATE TABLE employees (
idemployees INT AUTO_INCREMENT UNIQUE,
name_ VARCHAR(100),
lastname VARCHAR(100),
birthday DATE,
email VARCHAR(100),
phone VARCHAR(20),
address VARCHAR(255),
occupation VARCHAR(100),
date_recruitment DATE,
document_number INT UNIQUE,
idtype_document INT,
createdAt DATETIME,
updateAt DATETIME,
PRIMARY KEY(idemployees),
FOREIGN KEY (idtype_document) references type_document(idtype_document)
);
```

- Continuando tenemos la tabla customers, en la que se guardaran los datos de los clientes y también usamos una foreign key para el tipo de documento

```sql
CREATE TABLE customers (
    idcustomers INT AUTO_INCREMENT UNIQUE,
    name_ VARCHAR(100) NOT NULL,
    lastname VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20),
    birthday DATE,
    createdAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    updatedAt DATETIME ON UPDATE CURRENT_TIMESTAMP,
    document_NUMBER INT UNIQUE,
    idtype_document INT,
    PRIMARY KEY(idcustomers),
    FOREIGN KEY (idtype_document) references type_document(idtype_document)
);
```

- En la siguiente tabla, reserves, crearemos las reservas en la cuál se tendrá relación con la tabla payment, employees y customers.

```sql
CREATE TABLE reserves (
    idreserves INT AUTO_INCREMENT UNIQUE,
    state VARCHAR(100) NOT NULL,
    time_ TIME NOT NULL,
    price DECIMAL (10, 2) ,
    idpayment INT,
    date_reserve DATE NOT NULL,
    createdAt DATETIME DEFAULT CURRENT_TIMESTAMP,
    updatedAt DATETIME ON UPDATE CURRENT_TIMESTAMP,
    idcustomers INT,
    idemployees INT,
    PRIMARY KEY(idreserves),
    FOREIGN KEY (idcustomers) REFERENCES customers(idcustomers),
    FOREIGN KEY (idpayment) REFERENCES payment(idpayment),
    FOREIGN KEY (idemployees) REFERENCES employees(idemployees)
);
```

- Y ya para finaliza tenemos la tabla reservation_history, en la cuál se guardarán los clientes y su cantidad total de reservas hechas

```sql
CREATE TABLE reservation_history (
idreservation_history INT AUTO_INCREMENT,
idcustomers INT,
idreserves INT,
createdAt DATETIME DEFAULT CURRENT_TIMESTAMP,
updatedAt DATETIME ON UPDATE CURRENT_TIMESTAMP,
_comment TEXT,
PRIMARY KEY (idreservation_history),
FOREIGN KEY (idcustomers) REFERENCES customers(idcustomers),
FOREIGN KEY (idreserves) REFERENCES reserves(idreserves),
UNIQUE (idcustomers, idreserves)
);
```

- Una vez creadas, confirmamos que todas las tablas fueron generadas correctamente y están listas para almacenar datos.

Consultas en la Base de Datos

Primero damos orden de que database vamos a usar, y para eso hacemos el comando “USE” junto al nombre de la database de la siguiente manera:

```
USE BD_inventario;
```

#1 consulta: Una consulta simple dónde pedimos los siguientes atributos = “address”,”email”, “name_” de la tabla “employees”

```
SELECT address, email, name_ FROM employees;
```

Respuesta de la consulta:

address	email	name_
789 Pine St	michael.brown@example.com	Michael
321 Oak St	sarah.taylor@example.com	Sarah

#2 consulta: Consultar todos los registros de “customer”

```
SELECT * FROM customers;
SELECT * FROM payment;
```

Respuesta de la consulta:

idcustomers	name_	lastname	email	phone	birthday	createdAt	updatedAt	document	idtype_document
1	Alice	Johnson	alice.johnson@example.com	+123456789	1988-02-14	2024-09-05 23:37:44	NULL	123456	1
2	Bob	Williams	bob.williams@example.com	+234567890	1992-05-21	2024-09-05 23:37:44	NULL	234567	2
3	Carol	Brown	carol.brown@example.com	+345678901	1985-08-30	2024-09-05 23:37:44	NULL	345678	3
4	David	Jones	david.jones@example.com	+456789012	1990-11-05	2024-09-05 23:37:44	NULL	456789	1
5	Eve	Davis	eve.davis@example.com	+567890123	1995-12-22	2024-09-05 23:37:44	NULL	567890	2
6	Frank	Miller	frank.miller@example.com	+963852741	1987-11-25	2024-09-05 23:37:44	NULL	963852	1
7	Grace	Wilson	grace.wilson@example.com	+852741963	1991-03-03	2024-09-05 23:37:44	NULL	852741	1
8	Henry	Moore	henry.moore@example.com	+741852963	1992-12-08	2024-09-05 23:37:44	NULL	741852	1
9	Ivy	Taylor	ivy.taylor@example.com	+369852147	1986-05-20	2024-09-05 23:37:44	NULL	369852	2
10	Jack	Anderson	jack.anderson@example.com	+258963147	1989-08-15	2024-09-05 23:37:44	NULL	258963	1
11	Liam	Martinez	liam.martinez@example.com	+123987456	1988-03-18	2024-09-05 23:37:44	NULL	987123	1
12	Sophia	Garcia	sophia.garcia@example.com	+234876543	1993-04-29	2024-09-05 23:37:44	NULL	876234	2

#3 consulta: Consultar todos los pagos realizados con un método específico en este caso “Cash”

```
SELECT reserves.*, payment.pay_method FROM reserves
INNER JOIN payment ON  reserves.idpayment = payment.idpayment
WHERE payment.pay_method = 'Cash';
```

Respuesta de la consulta:

idreserves	state	time_	price	idpayment	date_reserve	createdAt	updatedAt	idcustomers	idemployees	pay_method
6	Confirmed	01:00 PM	180.00	3	2024-09-15	2024-09-06 21:32:21	NULL	1	1	Cash
9	Confirmed	07:00 PM	170.00	3	2024-09-18	2024-09-06 21:32:21	NULL	4	2	Cash
16	Confirmed	11:30 AM	170.00	3	2024-09-25	2024-09-06 21:32:21	NULL	6	1	Cash
19	Confirmed	05:30 PM	190.00	3	2024-09-28	2024-09-06 21:32:21	NULL	9	2	Cash

#4 consulta: Consultar las reservas que tengan un “Price” entre 170 y 200 y además que su “state” sea “Confirmed”:

```
SELECT idreserves, state, price FROM reserves
where reserves.price BETWEEN 170 AND 200 AND reserves.state = "Confirmed";
```

Respuesta de la consulta:

idreserves	state	price
6	Confirmed	180.00
9	Confirmed	170.00
16	Confirmed	170.00
19	Confirmed	190.00
NULL	NULL	NULL

#5 consulta: Consultar el “name” y “lastname” de “customers” dónde “birthday” sea mayor a “1995-01-01” o “name” contenga la letra “R” o que su “idtype_document” sea 1

```
SELECT name_, lastname FROM customers
WHERE customers.birthday > '1995-01-01' or customers.name_ Like '$R$' or customers.idtype_document = 1;
```

Respuesta de la consulta:

name_	lastname
Alice	Johnson
David	Jones
Eve	Davis
Frank	Miller
Grace	Wilson
Henry	Moore
Jack	Anderson
Liam	Martinez
Isabella	Ramirez

DISEÑO DE BASE DE DATOS

1. Modelo de Datos

- **Diagramas Entidad-Relación (ERD)**:
    - **Entidad type_document**: Almacena los tipos de documentos (por ejemplo, DNI, Pasaporte).
    - **Entidad payment**: Guarda la información sobre los métodos de pago (por ejemplo, Tarjeta de Crédito, Efectivo).
    - **Entidad employees**: Contiene datos de los empleados, como nombre, apellido, y detalles de contacto.
    - **Entidad customers**: Incluye información sobre los clientes, como nombre, apellido y correo electrónico.
    - **Entidad reserves**: Registra las reservas hechas, con información sobre el estado, precio y fecha.
    - **Entidad reservation_history**: Mantiene un historial de todas las reservas realizadas por cada cliente.

2. Consideraciones de Diseño

- **Elección de Claves Primarias:**

Cada tabla tiene una clave primaria (AUTO_INCREMENT) que asegura que cada registro tenga un identificador único. Esto es crucial para mantener la integridad de los datos y facilitar las búsquedas.

- **Relaciones entre Tablas:**
    - type_document se relaciona con employees y customers a través de la clave foránea idtype_document.
    - payment se relaciona con reserves a través de idpayment.
    - employees y customers están relacionados con reserves mediante idemployees y idcustomers.
    - reservation_history se conecta con customers y reserves para llevar un registro completo del historial de reservas.
- **Nombre de la Base de Datos y Descripción:**

Nombre: BD_inventario

Descripción: Esta base de datos gestiona un sistema de reservas. Incluye tablas para tipos de documentos, métodos de pago, empleados, clientes, reservas y un historial de

reservas. La base de datos está diseñada para mejorar la eficiencia y precisión en el proceso de reservaciones.

- **Entidades y Atributos:**

  - **type_document:** (Entidad Débil)
    - idtype_document, (Llave primaria)
    - type_document.

  - **payment:** (Entidad Débil)
    - idpayment, (Llave primaria)
    - pay_method.

  - **employees:** (Entidad Fuerte)
    - idemployees, (Llave primaria)
    - name_,
    - lastname,
    - birthday,
    - email,
    - phone,
    - address,
    - occupation,
    - date_recruitment,
    - document_number,
    - idtype_document, (Atributo multivalorado)

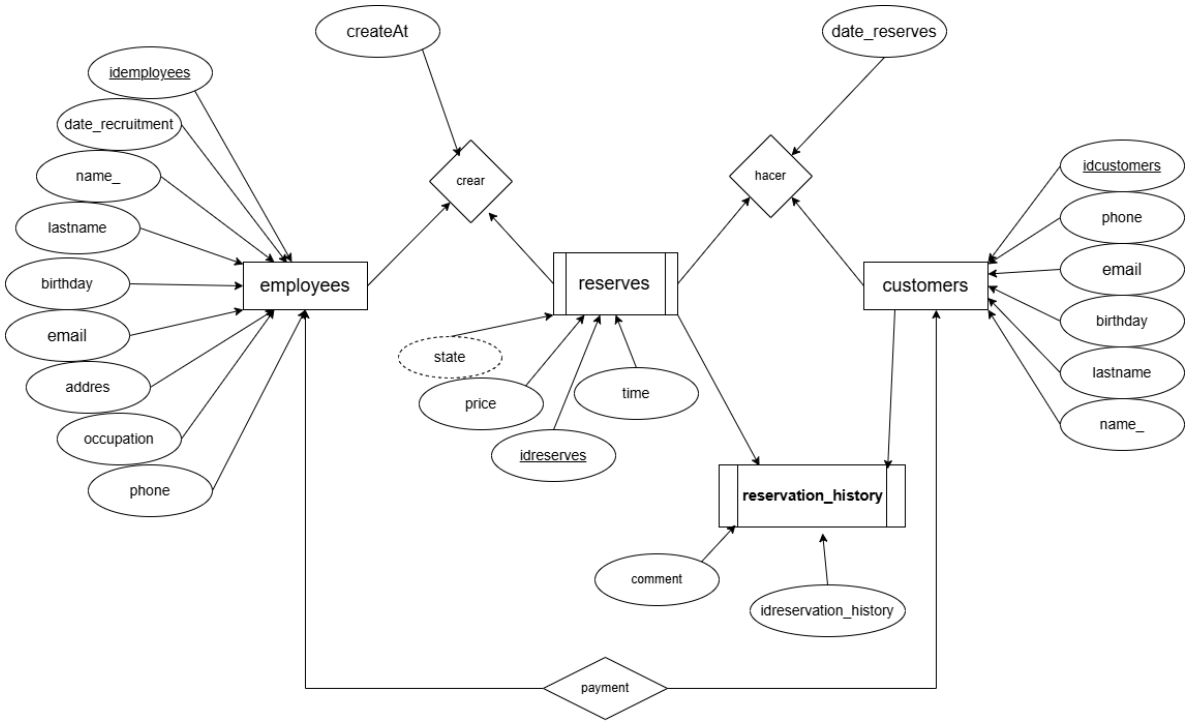
  - **customers:** (Entidad Fuerte)
    - idcustomers, (Llave primaria)
    - name_,
    - lastname,
    - email,
    - phone,
    - birthday,
    - document_NUMBER,
    - idtype_document. (Atributo multivalorado)

- **reserves:** (Entidad Débil)
  - idreserves, (Llave primaria)
  - state, (Atributo Derivado)
  - time_,
  - price,
  - idpayment, (Atributo multivalorado)
  - date_reserve,
  - idcustomers, (Atributo multivalorado)
  - idemployees. (Atributo multivalorado)

- **reservation_history:** (Entidad Débil)
  - idreservation_history, (Llave primaria)
  - idcustomers, (Atributo multivalorado)
  - idreserves, (Atributo multivalorado)
  - _comment.

- **Relaciones:**

  - Un cliente puede tener muchas reservas (relación uno a muchos).
  - Una reserva puede estar asociada con un solo método de pago y un solo empleado (relación uno a muchos).
  - Un método de pago puede ser usado en muchas reservas (relación uno a muchos).
  - Un empleado puede manejar muchas reservas (relación uno a muchos).


createAt
date_reserves
idemployees
date_recruitment
name_
lastname
birthday
email
addres
occupation
phone
crear
hacer
employees
reserves
customers
idcustomers
phone
email
birthday
lastname
name_
state
price
time
idreserves
reservation_history
comment
idreservation_history
payment

INTERPRETACIÓN DE RESULTADOS

1. **Objetivos de la Base de Datos:**
   - Optimización del Proceso de Reservaciones: El objetivo principal es mejorar la eficiencia y precisión en la gestión de reservas, abordando las falencias del sistema actual.

2. **Evaluación del Rendimiento:**
   - **Consultas:** Se debe evaluar si las consultas a la base de datos se ejecutan de manera eficiente. Las claves primarias y foráneas, así como los índices apropiados, deben mejorar el rendimiento de las consultas relacionadas con reservas, clientes, y empleados.
   - **Integridad de Datos:** La integridad referencial se mantiene mediante las claves foráneas. Las restricciones y validaciones implementadas aseguran que los datos se mantengan consistentes y precisos.

3. **Observaciones y Recomendaciones:**
   - **Observaciones:** Basado en el análisis de los resultados, se deben observar los puntos fuertes y las áreas que pueden requerir mejoras adicionales. Esto podría incluir ajustes en el diseño de la base de datos, optimización de consultas o adición de nuevas funcionalidades.
   - **Recomendaciones:** Basado en las observaciones, hacer recomendaciones para futuras mejoras. Esto podría incluir la revisión de índices, ajustes en la normalización, o la implementación de nuevos requisitos funcionales.

CONCLUSIÓN

La nueva base de datos aborda las falencias del sistema actual al ofrecer una solución estructurada y eficiente para la gestión de reservas. Se recomienda realizar pruebas continuas para evaluar el rendimiento y hacer ajustes según sea necesario para mantener la eficacia del sistema. La correcta implementación y ajuste continuo del diseño garantizarán el cumplimiento de los objetivos del proyecto y la mejora del proceso de reservaciones.

La interpretación de los resultados debe centrarse en cómo el diseño y la implementación de la base de datos cumplen con los objetivos establecidos. Analizar cómo cada componente de la base de datos contribuye a la mejora del proceso de reservaciones y cómo se alinean con los problemas identificados en el sistema actual proporciona una visión clara del éxito del proyecto y de las áreas para mejorar.

Referencias:

- Draw.io
- Moodle ITP
- Link repositorio Github: https://github.com/EstP19/Desarrollo_Base_De_Datos