INFORME DE BASES DE DATOS

Entrega de avance 2° corte

Información

NOMBRE DEL ESPACIO ACADÉMICO: DESARROLLO DE BASE DE DATOS
PROGRAMA: INGENIERÍA SISTEMAS
DOCENTE: BRAYAN ARCOS

YADIR ADRIAN MUÑOZ
yamunozr22@itp.edu.co

Contenido

Resumen Ejecutivo 3

Introducción 4

Metodología 5

Herramientas Utilizadas 5

Procedimientos 5

Desarrollo del Informe 6

Descripción de la Base de Datos 6

Esquema de la Base de Datos: Tablas, relaciones y claves principales 1

Consultas Realizadas 5

Consideraciones de Diseño 10

Nombre de la Base de Datos y Descripción 10

Entidades y Atributos - Base de Datos BarberExpress 10

DIAGRAMA ER 18

CONSULTASCON: CONCAT, LENGTH, UPPER, LOWER, SUBSTRING, ROUND, ABS, COUNT, SUM, AVG, MIN, GROUP BY, HAVING, IN, EXISTS, ANY, Subconsulta en la cláusula WHERE, Subconsulta en la cláusula FROM, Subconsulta en la cláusula SELECT 18

---------- CONCAT ---------- 19

---------- LENGTH ---------- 19

---------- UPPER ---------- 19

---------- LOWER ---------- 20

---------- SUBSTRING ---------- 20

---------- ROUND ---------- 21

---------- ABS ---------- 21

---------- COUNT ---------- 22

---------- SUM ---------- 22

---------- AVG ---------- 23

---------- MIN ---------- 23

---------- GROUP BY con HAVING ---------- 24

---------- IN ---------- 24

---------- EXISTS ---------- 25

---------- ANY ---------- 26

---------- Subconsulta en FROM---------- 26

---------- Subconsulta en SELECT ---------- 27

MANUAL DE MI BASE DE DATOS BARBEREXPRESS 27
Anexo el repositorio de GitHub: 34

Resumen Ejecutivo

Este informe se está realizando con el propósito de facilitar el aprendizaje de los conceptos fundamentales en el diseño y desarrollo de bases de datos. En este caso para una aplicación de barbería llamada BarberExpress, utilizando **MySQL Workbench** como herramienta principal.

A lo largo de esta primera parte del informe, se documentan paso a paso las metodologías necesarias para crear una base de datos eficiente y escalable, con el fin de aplicar estos conocimientos de manera práctica en futuros proyectos.

El objetivo es que, como estudiante, pueda entender cómo estructurar una base de datos a través de un ejemplo real.

También, se explican las consultas SQL utilizadas para insertar, consultar, actualizar y eliminar datos, así como para modificar la estructura de las tablas. Este modelo práctico permite comprender conceptos como la normalización, el uso de claves primarias y foráneas, y la importancia de mantener un historial de modificaciones mediante los campos **createdAt** y **updatedAt.** Estos atributos son recomendados por nuestro docente, ya que estos registros muy importantes en proyectos reales.

El informe concluye con recomendaciones para optimizar y mejorar el rendimiento de la base de datos, lo que no solo consolida el aprendizaje teórico, sino que también ofrece una aplicación tangible de los conocimientos adquiridos.

Introducción

El informe lo estoy desarrollando con el objetivo de que me sirva como una herramienta de aprendizaje práctico para el diseño y manejo de bases de datos, utilizando **MySQL Workbench**. Me permite adquirir y reforzar habilidades esenciales en SQL, incluyendo el diseño de esquemas, la manipulación de datos y la optimización de consultas.

El informe abarca tanto aspectos teóricos como prácticos, cubriendo temas clave como el diseño de tablas, la normalización, y la creación de relaciones entre entidades.

Además, este documento incluye instrucciones detalladas sobre el uso de comandos **SQL** para insertar, consultar, actualizar y eliminar datos, así como la modificación de la estructura de las tablas. Este enfoque práctico proporciona un contexto realista para el aprendizaje, facilitando la asimilación de los conceptos mediante la práctica activa.

El objetivo principal es que, como estudiante, pueda desarrollar un entendimiento integral del proceso de diseño y gestión de bases de datos, adquiriendo habilidades que son fundamentales para proyectos de desarrollo de software reales, donde implica mucha responsabilidad.

Metodología

Herramientas Utilizadas

- ✓ **MySQL Workbench:** Herramienta gráfica para la creación, gestión y diseño de bases de datos.
- ✓ **MySQL Server:** Motor de base de datos donde se almacena toda la información.
- ✓ **SQL:** Lenguaje utilizado para interactuar con la base de datos.

Procedimientos

1. **Creación de la base de datos** y definición del esquema.

2. **Diseño de tablas** con relaciones entre ellas, asegurando la integridad referencial.

3. **Implementación de comandos SQL** para gestionar datos (inserciones, actualizaciones, consultas, eliminaciones).

4. **Documentación** de las consultas SQL y resultados obtenidos.

Desarrollo del Informe

Descripción de la Base de Datos

De acuerdo con lo aprendido en clases, para este informe he decidido adentrarme en la parte de crear y gestionar una base de datos para una barbería la cual he llamado BarberExpress, para ello me he estado documentando sobre las tablas necesarias que debe tener.

Mencionare las tablas que debe tener una base de datos de comercio electrónico:

1. countries

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- name: VARCHAR(255), NOT NULL

2. departments

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- name: VARCHAR(100), NOT NULL
- id_country: INT(20), FOREIGN KEY (REFERENCES countries)

3 roles

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- name: VARCHAR(255), NOT NULL

4 users

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- email: VARCHAR(255), NOT NULL
- password: VARCHAR(255), NOT NULL

5 barber_status

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- status: VARCHAR(100), NOT NULL

5 people

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_user: INT(20), FOREIGN KEY (REFERENCES users)
- id_role: INT(20), FOREIGN KEY (REFERENCES roles)
- id_barber_state: INT(20), FOREIGN KEY (REFERENCES barber_status)
- first_name: VARCHAR(255), NOT NULL
- last_name: VARCHAR(255), NOT NULL
- phone: VARCHAR(10), NOT NULL
- profile_image: VARCHAR(255)
- created_at: TIMESTAMP
- updated_at: TIMESTAMP

7 people_location

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_people: INT(20), FOREIGN KEY (REFERENCES people)

- id_country: INT(20), FOREIGN KEY (REFERENCES countries)
- id_department: INT(20), FOREIGN KEY (REFERENCES departments)
- latitude: DECIMAL(10, 7), NOT NULL
- longitude: DECIMAL(10, 7), NOT NULL

8 days_week

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- days: VARCHAR(100), NOT NULL

9 hours

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- hour: VARCHAR(2)
- minutes: VARCHAR(2)
- periodo: VARCHAR(2)

10 schedules

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_day: INT(20), FOREIGN KEY (REFERENCES days_week)
- id_start_hour: INT(20), FOREIGN KEY (REFERENCES hours)
- id_end_hour: INT(20), FOREIGN KEY (REFERENCES hours)

11 states_barbershop

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- name: VARCHAR(255), NOT NULL

12 services_classification

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- name: VARCHAR(255), NOT NULL

13 bank_services

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_classification: INT(20), FOREIGN KEY (REFERENCES services_classification)
- name: VARCHAR(100), NOT NULL
- description: TEXT, NOT NULL
- price: DECIMAL(10, 2), NOT NULL

14 products_classification

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- name: VARCHAR(255), NOT NULL

15 bank_products

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_classification: INT(20), FOREIGN KEY (REFERENCES products_classification)
- name: VARCHAR(100), NOT NULL
- description: TEXT, NOT NULL
- price: DECIMAL(10, 2), NOT NULL
- image: VARCHAR(255), NOT NULL

16 barbershops

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_people: INT(20), FOREIGN KEY (REFERENCES people)

- id_location: INT(20), FOREIGN KEY (REFERENCES people_location)
- id_state_barbershops: INT(20), FOREIGN KEY (REFERENCES states_barbershop)
- id_shedule: INT(20), FOREIGN KEY (REFERENCES schedules)
- name: VARCHAR(255), NOT NULL
- description: TEXT
- image_banner: VARCHAR(255), NOT NULL
- created_at: TIMESTAMP

17 barbershop_products

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_barbershop: INT(20), FOREIGN KEY (REFERENCES barbershops)
- id_classification: INT(20), FOREIGN KEY (REFERENCES products_classification)
- id_product: INT(20), FOREIGN KEY (REFERENCES bank_products)
- created_at: TIMESTAMP

18 services_duration

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- duration: INT(20), NOT NULL

19 barbershop_services

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_barbershop: INT(20), FOREIGN KEY (REFERENCES barbershops)
- id_classification: INT(20), FOREIGN KEY (REFERENCES services_classification)
- id_service: INT(20), FOREIGN KEY (REFERENCES bank_services)
- id_duration: INT(20), FOREIGN KEY (REFERENCES services_duration)
- created_at: TIMESTAMP

20 barbershop_barbers

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_barbershop: INT(20), FOREIGN KEY (REFERENCES barbershops)
- id_people: INT(20), FOREIGN KEY (REFERENCES people)
- created_at: TIMESTAMP

21 payment_status

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- status: VARCHAR(100), NOT NULL

22 commission_value_product

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- value: DECIMAL(10, 2), NOT NULL

23 commission_value_appointment

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- value: DECIMAL(10, 2), NOT NULL

24 product_payments

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT

- id_people: INT(20), FOREIGN KEY (REFERENCES people)
- id_product: INT(20), FOREIGN KEY (REFERENCES bank_products)
- id_payment_status: INT(20), FOREIGN KEY (REFERENCES payment_status)

25 appointment_payments

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_people: INT(20), FOREIGN KEY (REFERENCES people)
- id_service: INT(20), FOREIGN KEY (REFERENCES bank_services)
- id_payment_status: INT(20), FOREIGN KEY (REFERENCES payment_status)

26 appointments

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_barbershop: INT(20), FOREIGN KEY (REFERENCES barbershops)
- id_service: INT(20), FOREIGN KEY (REFERENCES bank_services)
- id_duration: INT(20), FOREIGN KEY (REFERENCES services_duration)
- appointment_date: DATE, NOT NULL
- created_at: TIMESTAMP

27 barbershop_appointments

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_appointment: INT(20), FOREIGN KEY (REFERENCES appointments)
- id_barber: INT(20), FOREIGN KEY (REFERENCES people)
- created_at: TIMESTAMP

28 barber _schedule

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_barber: INT(20), FOREIGN KEY (REFERENCES people)
- id_day: INT(20), FOREIGN KEY (REFERENCES days_week)
- id_start_hour: INT(20), FOREIGN KEY (REFERENCES hours)
- id_end_hour: INT(20), FOREIGN KEY (REFERENCES hours)

29 appointment_ratings

- id: INT(20), PRIMARY KEY, AUTO_INCREMENT
- id_appointment: INT(20), FOREIGN KEY (REFERENCES appointments)
- rating: DECIMAL(2, 1), NOT NULL
- comment: TEXT

Esquema de la Base de Datos: Tablas, relaciones y claves principales

Las relaciones entre las tablas en son de diferentes tipos y representan conexiones lógicas entre las entidades.

las relaciones y su tipo:

1. Departments y Countries Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada departamento pertenece a un país. **Claves foráneas**:

- id_country -> countries(id)

2. People y Users/Roles/Barber_status Tipo de relación: Muchos a Uno (N:1) **Descripción**: Una persona está asociada a un usuario, un rol y un estado de barbero. **Claves foráneas**:

- id_user -> users(id)
- id_role -> roles(id)
- id_barber_state -> barber_status(id)

3. People_location y People/Countries/Departments Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada ubicación está vinculada a una persona, un país y un departamento. **Claves foráneas**:

- id_people -> people(id)
- id_country -> countries(id)
- id_department -> departments(id)

4. Schedules y Days_week/Hours Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada horario está asociado a un día y horas específicas de inicio y fin. **Claves foráneas**:

- id_day -> days_week(id)
- id_start_hour -> hours(id)
- id_end_hour -> hours(id)

5. Bank_services y Services_classification Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada servicio pertenece a una clasificación específica. **Claves foráneas**:

- id_classification -> services_classification(id)

6. Bank_products y Products_classification Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada producto pertenece a una clasificación específica. **Claves foráneas**:

- id_classification -> products_classification(id)

7. Barbershops y People/Location/States/Schedules Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada barbería está asociada a un propietario, ubicación, estado y horario. **Claves foráneas**:

- id_people -> people(id)
- id_location -> people_location(id)
- id_state_barbershops -> states_barbershop(id)
- id_shedule -> schedules(id)

8. Barbershop_products y Barbershops/Products/Classification Tipo de relación: Muchos a Muchos (N:M) **Descripción**: Relaciona barberías con productos y sus clasificaciones. **Claves foráneas**:

- id_barbershop -> barbershops(id)
- id_product -> bank_products(id)
- id_classification -> products_classification(id)

9. Barbershop_services y Barbershops/Services/Classification/Duration Tipo de relación: Muchos a Muchos (N:M) **Descripción**: Relaciona barberías con servicios, sus clasificaciones y duraciones. **Claves foráneas**:

- id_barbershop -> barbershops(id)
- id_service -> bank_services(id)
- id_classification -> services_classification(id)
- id_duration -> services_duration(id)

10. Barbershop_barbers y Barbershops/People Tipo de relación: Muchos a Muchos (N:M) **Descripción**: Relaciona barberías con los barberos que trabajan en ellas. **Claves foráneas**:

- id_barbershop -> barbershops(id)
- id_people -> people(id)

11. Product_payments y People/Products/Payment_status Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada pago de producto está asociado a una persona, producto y estado de pago. **Claves foráneas**:

- id_people -> people(id)
- id_product -> barbershop_products(id)
- id_payment_status -> payment_status(id)
- id_commission_value_product -> commission_value_product(id)

12. Appointment_payments y People/Appointments/Payment_status Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada pago de cita está asociado a una persona, cita y estado de pago. **Claves foráneas**:

- id_people -> people(id)
- id_appointment -> barbershop_services(id)
- id_payment_status -> payment_status(id)
- id_commission_value_appointment -> commission_value_appointment(id)

13. Posts y Barbershops/People/Classification Tipo de relación: Muchos a Uno (N:1) **Descripción**: Cada publicación está asociada a una barbería, persona y clasificación. **Claves foráneas**:

- id_barbershop -> barbershops(id)
- id_people -> people(id)
- id_classification -> posts_classification(id)

14. Comments y People/Posts Tipo de relación: Muchos a Uno (N:1)
Descripción: Cada comentario está asociado a una persona y una publicación. **Claves foráneas**:

- id_people -> people(id)
- id_post -> posts(id)

15. Reviews y People/Barbershops Tipo de relación: Muchos a Uno (N:1)
Descripción: Cada reseña está asociada a una persona y una barbería.
Claves foráneas:

- id_people -> people(id)
- id_barbershop -> barbershops(id)

Mostrare de forma breve como cree la base de datos:

Primero: se usa el comando **CREATE DATABASE** seguidamente el nombre de la base de datos, en este caso BarberExpress. Así:

```
1 •  CREATE DATABASE BarberExpress;
```

Segundo: usando el comando **USE** seguidamente el nombre de la base de datos previamente definido. Así:

```
1 •  USE BarberExpress;
```

Tercero: Creamos las tablas y sus relaciones de la siguiente manera:

```
5  CREATE TABLE people (
6    id INT(20) AUTO_INCREMENT PRIMARY KEY NOT NULL,
7    first_name VARCHAR(50) NOT NULL,
8    last_name VARCHAR(50) NOT NULL
9  );
```

```sql
CREATE TABLE `role` (
  id INT(20) AUTO_INCREMENT PRIMARY KEY NOT NULL,
  name VARCHAR(100) NOT NULL
);
```

Y a si sucesivamente se crean las demás tablas mencionadas anteriormente.

Consultas Realizadas

Mostrare las consultas que podemos realizar a la base de datos, claro que debemos tener datos o un registro en la base de datos que creamos, este paso lo omití porque es muy fácil de insertar registros a las tablas, con el comando INSERT INTO nombre de la tabla, mención de atributos, ya podemos insertar nuestros registros.

Procedo a mostrar los registros usando consultas vistas en clases:

Consulta 1: Obtener todos los clientes con sus ubicaciones, para ello se hace de la siguiente manera:

```sql
USE BarberExpress;

SELECT p.first_name, p.last_name, c.name AS country, s.name AS state
FROM people p
INNER JOIN people_location pl ON p.id = pl.people_id
INNER JOIN country c ON pl.country_id = c.id
INNER JOIN state s ON pl.state_id = s.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	country	state
Juan	Pérez	Colombia	Cundinamarca
Carlos	Rodríguez	Colombia	Antioquia
Ana	Gómez	Colombia	Valle del Cauca
María	López	Colombia	Atlántico
Luis	Martínez	Colombia	Bolívar

Consulta 2: Listar todos los métodos de pago utilizados en las órdenes, para ello se hace de la siguiente manera:

```
use ecommerce_db;

SELECT Ordenes.id, Metodos_Pago.nombre, Pagos.monto, Pagos.estado_pago
FROM Pagos
JOIN Ordenes ON Pagos.orden_id = Ordenes.id
JOIN Metodos_Pago ON Pagos.metodo_pago_id = Metodos_Pago.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

id	nombre	monto	estado_pago
1	Tarjeta de Crédito	500.00	Pendiente
2	PayPal	300.00	Completado
3	Transferencia Bancaria	100.00	Completado
4	Crédito en Tienda	200.00	Fallido
5	Bitcoin	150.00	Pendiente
6	Tarjeta de Débito	75.00	Completado
7	Pago contra entrega	400.00	Completado
8	Google Pay	350.00	Pendiente
9	Apple Pay	600.00	Completado
10	PSE	250.00	Pendiente

Consulta 3: Consultar todas las barbershops con sus dueños, para ello se hace de la siguiente manera:

```
USE BarberExpress;

SELECT b.name AS barbershop_name, p.first_name, p.last_name
FROM barbershop b
INNER JOIN people p ON b.people_id = p.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

barbershop_name	first_name	last_name
Carlos Barbershop	Carlos	Rodríguez
Luis Cuts	Luis	Martínez
Juan´s Hair Salon	Juan	Pérez
Ana Styles	Ana	Gómez
María´s Barber	María	López

Consulta 4: Mostrar todas las citas con información del cliente y servicio

```sql
USE BarberExpress;

SELECT p.first_name, p.last_name, s.name AS service_name, a.date
FROM appointments a
INNER JOIN people p ON a.people_id = p.id
INNER JOIN services s ON a.service_id = s.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	service_name	date
Juan	Pérez	Basic Haircut	2024-10-06 10:00:00
Carlos	Rodríguez	Beard Trim	2024-10-07 11:00:00
Ana	Gómez	Hair Color	2024-10-08 12:00:00
María	López	Hair Wash	2024-10-09 13:00:00
Luis	Martínez	Deluxe Haircut	2024-10-10 14:00:00

Consulta 5: Obtener todos los barberos con sus barbershops

```sql
USE BarberExpress;

SELECT br.name AS barber_name, b.name AS barbershop_name
FROM barbers br
INNER JOIN barbershop b ON br.barbershop_id = b.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

barber_name	barbershop_name
Pedro Torres	Carlos Barbershop
Miguel Sánchez	Luis Cuts
Jorge Ramírez	Juan´s Hair Salon
Oscar Díaz	Ana Styles
Fernando Mendoza	María´s Barber

Consulta 7: Buscar registros donde el nombre del usuario contiene la letra 'e'

```sql
use ecommerce_db;

SELECT *
FROM Usuarios
WHERE nombre LIKE '%e%';
```

Result Grid

id	nombre	email	contraseña	telefono	direccion	createdAt	updatedAt
1	Juan Pérez	juan@example.com	pass1234	555-1234	Calle Falsa 123	2024-09-11 12:53:14	2024-09-11 12:53:14
2	Ana López	ana@example.com	pass5678	555-5678	Av. Siempre Viva 742	2024-09-11 12:53:14	2024-09-11 12:53:14
4	Luisa Martínez	luisa@example.com	pass3456	555-9876	Av. Principal 10	2024-09-11 12:53:14	2024-09-11 12:53:14
5	Miguel Rodríguez	miguel@example.com	pass7890	555-5432	Calle La Estrella 20	2024-09-11 12:53:14	2024-09-11 12:53:14
6	Sofía González	sofia@example.com	pass1235	555-6543	Calle El Sol 88	2024-09-11 12:53:14	2024-09-11 12:53:14
7	Pedro Torres	pedro@example.com	pass4567	555-7654	Av. El Bosque 25	2024-09-11 12:53:14	2024-09-11 12:53:14
8	María Fernández	maria@example.com	pass8901	555-8767	Calle Los Álamos 100	2024-09-11 12:53:14	2024-09-11 12:53:14
9	Jorge Morales	jorge@example.com	pass2345	555-9878	Av. Los Olivos 70	2024-09-11 12:53:14	2024-09-11 12:53:14
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Consulta 8: Listar todos los pagos con información del cliente y la barbería

```sql
USE BarberExpress;

SELECT p.first_name, p.last_name, b.name AS barbershop, py.amount
FROM payments py
INNER JOIN people p ON py.people_id = p.id
INNER JOIN barbershop b ON py.barbershop_id = b.id;
```

Result Grid

first_name	last_name	barbershop	amount
Juan	Pérez	Carlos Barbershop	20.00
Carlos	Rodríguez	Luis Cuts	15.00
Ana	Gómez	Juan´s Hair Salon	40.00
María	López	Ana Styles	10.00
Luis	Martínez	María´s Barber	30.00

Consulta 9: Mostrar todas las reseñas con el nombre del cliente y la barbería.

```sql
USE BarberExpress;

SELECT p.first_name, p.last_name, b.name AS barbershop, r.rating, r.comment
FROM reviews r
INNER JOIN people p ON r.people_id = p.id
INNER JOIN barbershop b ON r.barbershop_id = b.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	barbershop	rating	comment
Juan	Pérez	Carlos Barbershop	5	Great service!
Carlos	Rodríguez	Luis Cuts	4	Excellent beard trim.
Ana	Gómez	Juan´s Hair Salon	5	Loved the hair coloring.
María	López	Ana Styles	3	Cancelled, but still good.
Luis	Martínez	María´s Barber	5	Best haircut ever!

Consulta 10: Obtener todos los posts con información del barbero y la barbería.

```sql
USE BarberExpress;

SELECT b.name AS barbershop, br.name AS barber, p.title
FROM posts p
INNER JOIN barbershop b ON p.barbershop_id = b.id
INNER JOIN barbers br ON p.barber_id = br.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

barbershop	barber	title
Carlos Barbershop	Pedro Torres	New Haircut Styles
Luis Cuts	Miguel Sánchez	Beard Care Tips
Juan´s Hair Salon	Jorge Ramírez	Hair Coloring Trends
Ana Styles	Oscar Díaz	Special Offers
María´s Barber	Fernando Mendoza	Deluxe Services

Consulta 11: Listar los horarios de cada barbería.

```sql
USE BarberExpress;

SELECT b.name AS barbershop, s.day, s.start_time, s.end_time
FROM schedule s
INNER JOIN barbershop b ON s.barbershop_id = b.id;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

barbershop	day	start_time	end_time
Carlos Barbershop	Monday	08:00:00	18:00:00
Luis Cuts	Tuesday	09:00:00	19:00:00
Juan ´s Hair Salon	Wednesday	08:00:00	17:00:00
Ana Styles	Thursday	10:00:00	20:00:00
María ´s Barber	Friday	08:30:00	18:30:00

Consideraciones de Diseño

Nombre de la Base de Datos y Descripción

Nombre de la Base de Datos: BarberExpress

Descripción: La base de datos BarberExpress ha sido diseñada para servir como fundamento sólido en la gestión de información relacionada con servicios de barbería. Esta estructura de datos permite el almacenamiento y la organización eficiente de información crítica sobre clientes, barberos, servicios y transacciones.

Entidades y Atributos - Base de Datos BarberExpress

Entidades:

1. countries (Entidad Fuerte)

- id: INT

- name: VARCHAR

2. **departments** (Entidad fuerte)

- id: INT
- name: VARCHAR
- id_country: INT

3. **roles** (Entidad Fuerte)

- id: INT
- name: VARCHAR

4. **users** (Entidad Fuerte)

- id: INT
- email: VARCHAR
- password: VARCHAR

5. **barber_status** (Entidad Fuerte)

- id: INT
- status: VARCHAR

6. **people** (Entidad fuerte)

- id: INT
- id_user: INT
- id_role: INT
- id_barber_state: INT
- first_name: VARCHAR
- last_name: VARCHAR
- phone: VARCHAR

- profile_image: VARCHAR
- created_at: TIMESTAMP
- updated_at: TIMESTAMP

7. **people_location** (Entidad fuerte)

- id: INT
- id_people: INT
- id_country: INT
- id_department: INT
- latitude: DECIMAL
- longitude: DECIMAL

8. **days_week** (Entidad Fuerte)

- id: INT
- days: VARCHAR

9. **hours** (Entidad Fuerte)

- id: INT
- hour: VARCHAR
- minutes: VARCHAR
- periodo: VARCHAR

10. **schedules** (Entidad fuerte)

- id: INT
- id_day: INT
- id_start_hour: INT
- id_end_hour: INT

11. **states_barbershop** (Entidad Fuerte)

- id: INT
- name: VARCHAR

12. **services_classification** (Entidad Fuerte)

- id: INT
- name: VARCHAR

13. **bank_services** (Entidad fuerte)

- id: INT
- id_classification: INT
- name: VARCHAR
- description: TEXT
- price: DECIMAL

14. **products_classification** (Entidad Fuerte)

- id: INT
- name: VARCHAR

15. **bank_products** (Entidad fuerte)

- id: INT
- id_classification: INT
- name: VARCHAR
- description: TEXT
- price: DECIMAL
- image: VARCHAR

16. **barbershops** (Entidad fuerte)

- id: INT
- id_people: INT
- id_location: INT
- id_state_barbershops: INT
- id_shedule: INT
- name: VARCHAR
- description: TEXT
- image_banner: VARCHAR
- created_at: TIMESTAMP

17. **barbershop_products** (Entidad fuerte)

- id: INT
- id_barbershop: INT
- id_classification: INT
- id_product: INT
- created_at: TIMESTAMP

18. **services_duration** (Entidad Fuerte)

- id: INT
- duration: INT

19. **barbershop_services** (Entidad fuerte)

- id: INT
- id_barbershop: INT
- id_classification: INT
- id_service: INT

- id_duration: INT
- created_at: TIMESTAMP

20. **barbershop_barbers** (Entidad fuerte)

- id: INT
- id_barbershop: INT
- id_people: INT
- created_at: TIMESTAMP

21. **payment_status** (Entidad Fuerte)

- id: INT
- status: VARCHAR

22. **commission_value_product** (Entidad Fuerte)

- id: INT
- value: DECIMAL

23. **commission_value_appointment** (Entidad Fuerte)

- id: INT
- value: DECIMAL

24. **product_payments** (Entidad fuerte)

- id: INT
- id_people: INT
- id_product: INT
- id_payment_status: INT
- quantity: INT
- total: DECIMAL

- total_receive_barbershop: DECIMAL
- id_commission_value_product: INT
- created_at: TIMESTAMP

25. appointment_payments (Entidad fuerte)

- id: INT
- id_people: INT
- id_appointment: INT
- id_payment_status: INT
- total: DECIMAL
- total_receive_barbershop: DECIMAL
- id_commission_value_appointment: INT
- created_at: TIMESTAMP

26. posts_classification (Entidad Fuerte)

- id: INT
- name: VARCHAR

27. posts (Entidad fuerte)

- id: INT
- id_barbershop: INT
- id_people: INT
- id_classification: INT
- title: VARCHAR
- content: TEXT
- media: VARCHAR

- created_at: TIMESTAMP

28. comments (fuerte)

- id: INT
- id_people: INT
- id_post: INT
- comment: TEXT
- created_at: TIMESTAMP

29. reviews (Entidad fuerte)

- id: INT
- id_people: INT
- id_barbershop: INT
- comment: TEXT
- rating: INT
- created_at: TIMESTAMP

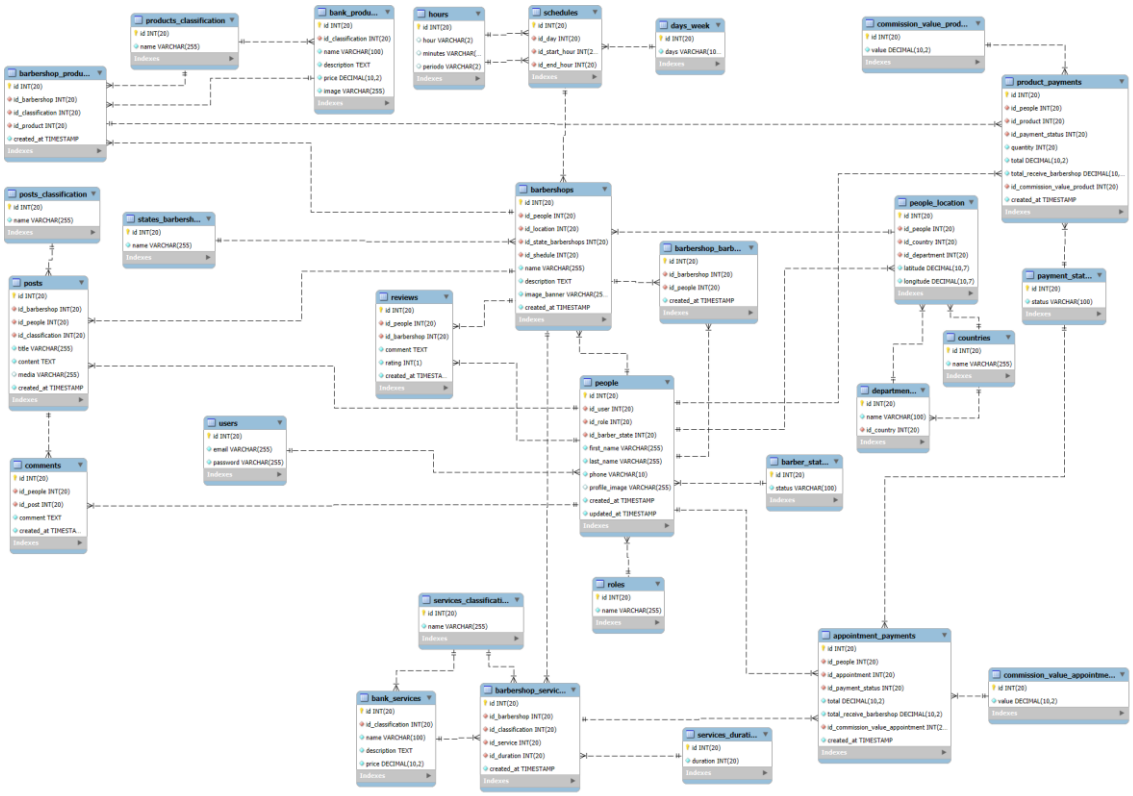

DIAGRAMA ER

https://www.mermaidchart.com/app/projects/b3a185f3-14bb-4163-a323-cf40906d2115/diagrams/c896bf8b-1628-464a-a9d3-a26dfc8055c0/version/v0.1/edit

https://www.mermaidchart.com/raw/c896bf8b-1628-464a-a9d3-a26dfc8055c0?theme=light&version=v0.1&format=svg

CONSULTASCON: CONCAT, LENGTH, UPPER, LOWER, SUBSTRING, ROUND, ABS, COUNT, SUM, AVG, MIN, GROUP BY, HAVING, IN, EXISTS, ANY, Subconsulta en la cláusula WHERE, Subconsulta en la cláusula FROM, Subconsulta en la cláusula SELECT

---------- CONCAT ----------

```
4   -- Ejemplo 1: Combinar nombre y apellido de las personas
5   SELECT CONCAT(first_name, ' ',last_name) AS nombre_completo
6   FROM people;
7   /* Explicación: CONCAT une el first_name con un espacio y last_name para crear un nombre completo */
8
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre_completo
Juan Pérez
Maria González
Carlos Rodriguez

---------- LENGTH ----------

```
15  -- Ejemplo 1: Encontrar emails con más de 20 caracteres
16  SELECT email, LENGTH(email) AS longitud_email
17  FROM users
18  WHERE LENGTH(email) > 20;
19  /* Explicación: LENGTH cuenta el número de caracteres en el email y filtra los que tienen más de 20 */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

email	longitud_email
YADIRADRIAN@GMAIL.COM	21

---------- UPPER ----------

```
36  -- Ejemplo 1: Mostrar nombres de barberías en mayúsculas
37  SELECT UPPER(name) AS nombre_mayusculas
38  FROM barbershops;
39  /* Explicación: Convierte el nombre de cada barbería a mayúsculas */
40
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre_mayusculas
BARBERSHOP ELITE
CORTES PREMIUM
STYLE BARBER

```sql
-- Ejemplo 2: Nombres y apellidos en mayúsculas
SELECT UPPER(first_name) AS nombre, UPPER(last_name) AS apellido
FROM people;
/* Explicación: Convierte tanto el nombre como el apellido a mayúsculas */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	apellido
JUAN	PÉREZ
MARIA	GONZÁLEZ
CARLOS	RODRIGUEZ

---------- LOWER ----------

```sql
-- Ejemplo 1: Emails en minúsculas
SELECT LOWER(email) AS email_minusculas
FROM users;
/* Explicación: Convierte todos los emails a minúsculas */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

email_minusculas
juan@email.com
maria@email.com
carlos@email.com
juan@email.com
maria@email.com
yadiradrian@gmail.com

---------- SUBSTRING ----------

```sql
-- Ejemplo 1: Primeros 3 caracteres del nombre
SELECT first_name, SUBSTRING(first_name, 1, 3) AS iniciales
FROM people;
/* Explicación: Extrae los primeros 3 caracteres del nombre, comenzando en posición 1 */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	iniciales
Juan	Jua
Maria	Mar
Carlos	Car

```sql
-- Ejemplo 2: Últimos 4 dígitos del teléfono
SELECT phone, SUBSTRING(phone, -4) AS ultimos_digitos
FROM people;
/* Explicación: Extrae los últimos 4 dígitos del número telefónico */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

phone	ultimos_digitos
3001234567	4567
3009876543	6543
3004567890	7890

---------- ROUND ----------

```sql
-- Ejemplo 1: Redondear precios de servicios
SELECT name, price, ROUND(price, -3) AS precio_redondeado
FROM bank_services;
/* Explicación: Redondea el precio al millar más cercano *
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	price	precio_redondeado
Corte Clásico	25000.00	25000
Afeitado Premium	35000.00	35000
Peinado Ejecutivo	20000.00	20000

```sql
-- Ejemplo 2: Redondear comisiones
SELECT total, ROUND(total * 0.15, 2) AS comision
FROM appointment_payments;
/* Explicación: Calcula el 15% del total y redondea a 2 decimales */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total	comision
25000.00	3750.00
35000.00	5250.00
20000.00	3000.00

---------- ABS ----------

```sql
-- Ejemplo 1: Diferencia absoluta entre precio y precio recibido
SELECT total, total_receive_barbershop,
       ABS(total - total_receive_barbershop) AS diferencia
FROM appointment_payments;
/* Explicación: Muestra la diferencia absoluta entre el total y lo recibido */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total	total_receive_barbershop	diferencia
25000.00	22500.00	2500.00
35000.00	31500.00	3500.00
20000.00	18000.00	2000.00

```sql
-- Ejemplo 2: Diferencia absoluta de latitudes
SELECT ABS(p1.latitude - p2.latitude) AS diff_latitud
FROM people_location p1, people_location p2
WHERE p1.id != p2.id
LIMIT 3;
/* Explicación: Calcula la diferencia absoluta entre latitudes de diferentes ubicaciones */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

diff_latitud
14.7216000
39.3147000
14.7216000

---------- COUNT ----------

```
-- Ejemplo 1: Contar barberos por barbería
SELECT b.name, COUNT(bb.id_people) AS numero_barberos
FROM barbershops b
LEFT JOIN barbershop_barbers bb ON b.id = bb.id_barbershop
GROUP BY b.name;
/* Explicación: Cuenta cuántos barberos hay en cada barbería */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	numero_barberos
BarberShop Elite	1
Cortes Premium	1
Style Barber	1

```
-- Ejemplo 2: Contar servicios por clasificación
SELECT sc.name, COUNT(bs.id) AS numero_servicios
FROM services_classification sc
LEFT JOIN bank_services bs ON sc.id = bs.id_classification
GROUP BY sc.name;
/* Explicación: Cuenta cuántos servicios hay en cada clasificación */
```

---------- SUM ----------

```
-- Ejemplo 1: Total de ventas por barbería
SELECT b.name, SUM(ap.total) AS ventas_totales
FROM barbershops b
INNER JOIN barbershop_services bs ON b.id = bs.id_barbershop
INNER JOIN appointment_payments ap ON bs.id = ap.id_appointment
GROUP BY b.name;
/* Explicación: Suma todos los pagos recibidos por cada barbería */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	ventas_totales
BarberShop Elite	25000.00
Cortes Premium	35000.00
Style Barber	20000.00

---------- AVG ----------

```
-- Ejemplo 1: Promedio de calificaciones por barbería
SELECT b.name, AVG(r.rating) AS calificacion_promedio
FROM barbershops b
INNER JOIN reviews r ON b.id = r.id_barbershop
GROUP BY b.name;
/* Explicación: Calcula el promedio de las calificaciones para cada barbería */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	calificacion_promedio
BarberShop Elite	5.0000
Cortes Premium	5.0000
Style Barber	4.0000

```
-- Ejemplo 2: Promedio de precios por clasificación de servicios
SELECT sc.name, AVG(bs.price) AS precio_promedio
FROM services_classification sc
INNER JOIN bank_services bs ON sc.id = bs.id_classification
GROUP BY sc.name;
/* Explicación: Calcula el precio promedio de los servicios por clasificación */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	precio_promedio
Afeitado	35000.000000
Corte de cabello	25000.000000
Tinte	20000.000000

---------- MIN ----------

```
-- Ejemplo 1: Servicio más económico por clasificación
SELECT sc.name, MIN(bs.price) AS precio_minimo
FROM services_classification sc
INNER JOIN bank_services bs ON sc.id = bs.id_classification
GROUP BY sc.name;
/* Explicación: Encuentra el precio más bajo de los servicios en cada clasificación */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	precio_minimo
Afeitado	35000.00
Corte de cabello	25000.00
Tinte	20000.00

```sql
-- Ejemplo 2: Primera cita del día por barbería
SELECT b.name, MIN(ap.created_at) AS primera_cita
FROM barbershops b
INNER JOIN barbershop_services bs ON b.id = bs.id_barbershop
INNER JOIN appointment_payments ap ON bs.id = ap.id_appointment
GROUP BY b.name;
/* Explicación: Encuentra la hora de la primera cita registrada para cada barbería */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	primera_cita
BarberShop Elite	2024-10-24 22:35:03
Cortes Premium	2024-10-24 22:35:03
Style Barber	2024-10-24 22:35:03

---------- GROUP BY con HAVING ----------

```sql
-- Ejemplo 1: Barberías con más de 3 barberos
SELECT b.name, COUNT(bb.id_people) AS numero_barberos
FROM barbershops b
JOIN barbershop_barbers bb ON b.id = bb.id_barbershop
GROUP BY b.name
HAVING numero_barberos >=1;
/* Explicación: Agrupa por barbería y filtra las si es mayor a igual a 1 barberos */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	numero_barberos
BarberShop Elite	1
Cortes Premium	1
Style Barber	1

```sql
-- Ejemplo 2: Clasificaciones de servicios con precio promedio mayor a 15000
SELECT sc.name, AVG(bs.price) AS precio_promedio
FROM services_classification sc
JOIN bank_services bs ON sc.id = bs.id_classification
GROUP BY sc.name
HAVING precio_promedio > 15000;
/* Explicación: Agrupa por clasificación y muestra solo las que tienen precio promedio superior a 30000 */
```

---------- IN ----------

```sql
-- Ejemplo 1: Encontrar personas en roles específicos
SELECT first_name, last_name
FROM people
WHERE id_role IN (SELECT id FROM roles WHERE name IN ('Barbero', 'Administrador'));
/* Explicación: Encuentra personas que tienen rol de barbero o administrador */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name
Maria	González
Carlos	Rodriguez

```sql
-- Ejemplo 2: Servicios en categorías populares
SELECT name
FROM bank_services
WHERE id_classification IN (
    SELECT id FROM services_classification
    WHERE name IN ('Corte de cabello', 'Barba', 'Afeitado')
);
/* Explicación: Encuentra servicios que pertenecen a las categorías especificadas */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name
Corte Clásico
Afeitado Premium

---------- EXISTS ----------

```sql
-- Ejemplo 1: Barberías que tienen al menos un barbero disponible
SELECT b.name
FROM barbershops b
WHERE EXISTS (
    SELECT 1
    FROM barbershop_barbers bb
    JOIN people p ON bb.id_people = p.id
    WHERE bb.id_barbershop = b.id
    AND p.id_barber_state = 1
);
/* Explicación: Encuentra barberías que tienen al menos un barbero en estado disponible */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name
Cortes Premium
Style Barber

```sql
-- Ejemplo 2: Clientes que han hecho comentarios
SELECT p.first_name, p.last_name
FROM people p
WHERE EXISTS (
    SELECT 1
    FROM comments c
    WHERE c.id_people = p.id
);
/* Explicación: Encuentra personas que han realizado al menos un comentario */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name
Juan	Pérez
Maria	González
Carlos	Rodriguez

---------- ANY ----------

```sql
-- Ejemplo 1: Servicios más caros que cualquier producto
SELECT name, price
FROM bank_services
WHERE price > ANY (
    SELECT price FROM bank_products
);
/* Explicación: Encuentra servicios cuyo precio es mayor que al menos un producto */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	price
Afeitado Premium	35000.00

---------- Subconsulta en FROM----------

```sql
-- Ejemplo 1: Ranking de barberías por ingresos
SELECT b.name, t.ingresos_totales,
       RANK() OVER (ORDER BY t.ingresos_totales DESC) as ranking
FROM barbershops b
JOIN (
    SELECT bs.id_barbershop, SUM(ap.total) as ingresos_totales
    FROM barbershop_services bs
    JOIN appointment_payments ap ON bs.id = ap.id_appointment
    GROUP BY bs.id_barbershop
) t ON b.id = t.id_barbershop;
/* Explicación: Crea un ranking de barberías basado en sus ingresos totales */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	ingresos_totales	ranking
Cortes Premium	35000.00	1
BarberShop Elite	25000.00	2
Style Barber	20000.00	3

---------- Subconsulta en SELECT ----------

```sql
-- Ejemplo 1: Información detallada de barberías
SELECT b.name,
       (SELECT COUNT(*) FROM barbershop_barbers bb WHERE bb.id_barbershop = b.id) as total_barberos,
       (SELECT AVG(rating) FROM reviews r WHERE r.id_barbershop = b.id) as calificacion_promedio
FROM barbershops b;
/* Explicación: Muestra nombre de barbería junto con total de barberos y calificación promedio */
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	total_barberos	calificacion_promedio
BarberShop Elite	1	5.0000
Cortes Premium	1	5.0000
Style Barber	1	4.0000

MANUAL DE MI BASE DE DATOS BARBEREXPRESS

Descripción General

BarberExpress es un sistema de gestión para barberías que permite administrar usuarios, servicios, productos, citas, pagos y contenido social.

1. Gestión de Usuarios y Perfiles

Sistema de Usuarios

- **Users**: Tabla fundamental que almacena las credenciales de acceso
  - ✓ Email: Identificador único para inicio de sesión
  - ✓ Password: Contraseña encriptada

Roles y Permisos

- **Roles**: Define tres niveles de acceso:
  - ✓ **Cliente**: Usuario regular que puede agendar citas y comprar productos
  - ✓ **Barbero**: Profesional que presta servicios en las barberías
  - ✓ **Administrador**: Propietario/gestor de barbería con permisos completos

Información Personal

- **People**: Almacena datos personales detallados
    - ✓ Nombres y apellidos del usuario
    - ✓ Número de teléfono para contacto
    - ✓ Imagen de perfil (URL)
    - ✓ Estado del barbero (solo aplica para rol barbero)
    - ✓ Fechas de creación y actualización del perfil

Estados de Barberos

- **Barber_status**: Gestiona la disponibilidad de los barberos
    - ✓ Disponible: Listo para atender clientes
    - ✓ Ocupado: Actualmente atendiendo
    - ✓ En descanso: No disponible temporalmente

2. Sistema de Ubicación y Geografía

Estructura Geográfica

- **Countries**: Registro de países disponibles en el sistema
- **Departments**: Divisiones administrativas/estados de cada país

Localización Específica

- **People_location**: Sistema de geolocalización que incluye:
    - ✓ Coordenadas precisas (latitud/longitud)
    - ✓ Referencias a país y departamento
    - ✓ Vinculación con usuarios específicos

3. Administración de Barberías

Información Básica

- **Barbershops**: Datos principales del negocio

- ✓ Nombre comercial
- ✓ Descripción detallada del negocio
- ✓ Imagen de banner para marketing
- ✓ Estado actual de operación
- ✓ Ubicación específica
- ✓ Horarios de atención

Control de Horarios

- **Days_week**: Días de operación
- **Hours**: Sistema de horarios que maneja:
  - ✓ Horas (formato 12 horas)
  - ✓ Minutos
  - ✓ Período (AM/PM)
- **Schedules**: Combinación de días y horarios específicos

Estados de Operación

- **States_barbershop**: Define el estado operativo
  - ✓ Abierto: En funcionamiento normal
  - ✓ Cerrado: Fuera de horario o temporalmente
  - ✓ En mantenimiento: No disponible por trabajos o renovaciones

4. Catálogo de Servicios y Productos

Servicios

- **Services_classification**: Categorías detalladas de servicios
  - ✓ Cada tipo de servicio está clasificado para fácil navegación
  - ✓ Permite organización eficiente del catálogo

- **Bank_services**: Catálogo completo que incluye:
    - ✓ Nombre del servicio
    - ✓ Descripción detallada
    - ✓ Precio establecido
    - ✓ Clasificación del servicio

Productos

- **Products_classification**: Organización de productos por categorías
    - ✓ Facilita la búsqueda y organización del inventario
- **Bank_products**: Inventario detallado con:
    - ✓ Nombre del producto
    - ✓ Descripción completa
    - ✓ Precio de venta
    - ✓ Imagen del producto
    - ✓ Categoría asociada

5. Sistema Financiero y Pagos

Control de Pagos

- **Payment_status**: Estados posibles de transacciones:
    - ✓ Pendiente: Pago iniciado pero no completado
    - ✓ Completado: Transacción exitosa
    - ✓ Fallido: Error en el proceso de pago
    - ✓ Reembolsado: Devolución procesada

Sistema de Comisiones

- **Commission_value_product**: Comisiones por venta de productos

- **Commission_value_appointment**: Comisiones por servicios prestados

Registro de Transacciones

- **Product_payments**: Detalle de ventas de productos
  - ✓ Cliente que realiza la compra
  - ✓ Producto adquirido
  - ✓ Cantidad
  - ✓ Total de la venta
  - ✓ Comisión aplicada
  - ✓ Monto final para la barbería
- **Appointment_payments**: Registro de pagos por servicios
  - ✓ Cliente que recibe el servicio
  - ✓ Servicio prestado
  - ✓ Total cobrado
  - ✓ Comisión del sistema
  - ✓ Monto final para la barbería

6. Sistema Social y Engagement

Publicaciones y Contenido

- **Posts_classification**: Tipos de contenido:
  - ✓ Promociones
  - ✓ Eventos
  - ✓ Noticias
  - ✓ Consejos
  - ✓ Tendencias
  - ✓ Historias

    - ✓ Tutoriales
    - ✓ Entrevistas
    - Productos
- **Posts**: Sistema de publicaciones que incluye:
    - ✓ Título del post
    - ✓ Contenido detallado
    - ✓ Elementos multimedia
    - ✓ Autor
    - ✓ Barbería asociada
    - ✓ Fecha de publicación

Interacción con Usuarios

- **Comments**: Sistema de comentarios en publicaciones
    - ✓ Autor del comentario
    - ✓ Contenido
    - ✓ Fecha
    - ✓ Post relacionado
- **Reviews**: Sistema de reseñas y calificaciones
    - ✓ Calificación numérica (1-5)
    - ✓ Comentario detallado
    - ✓ Cliente que realiza la reseña
    - ✓ Barbería evaluada
    - ✓ Fecha de la reseña

Notas Importantes de Implementación

1. **Seguridad**:
    - ✓ Todas las contraseñas deben estar encriptadas
    - ✓ Los permisos deben validarse en cada operación
2. **Validaciones**:
    - ✓ Verificar roles antes de acciones específicas
    - ✓ Controlar estados de barberos antes de asignar citas
    - ✓ Validar horarios disponibles
3. **Geolocalización**:
    - ✓ Usar coordenadas precisas para ubicación
    - ✓ Implementar cálculos de distancia para búsquedas
4. **Transacciones**:
    - ✓ Asegurar la integridad en operaciones de pago
    - ✓ Mantener registro detallado de comisiones
    - ✓ Validar estados de pago

De momento, es todo, este seria el segundo avance.

Anexo el repositorio de GitHub:

https://github.com/1007748140/BD_REPOSITORIO.git