

# **Reto final base de datos**

## **Aclaraciones del proceso**

Johan Steven Cifuentes  
Julio Vasquez Rojas  
Cristian Moreno Monsalve

Versión: 0100  
Fecha: 04/04/2024

## Contenido

Reto Asignado: Barbería .....	4
Modelo Entidad-Relación .....	4
Relaciones: .....	5
Atributos: .....	5
Diagrama ER: .....	7
Modelo-Relacional.....	7
Normalización del modelo: .....	8
Modelo diagramado en Workbench .....	11
Resolviendo conflicto: .....	11
Comandos SQL de creación de las Tablas: .....	12
Esquema barbería: .....	12
Poblado de base de datos: .....	21
Consultas de base de datos.....	35
Vistas: .....	36
Triggers:.....	41
Procedimientos: .....	43
Conexión desde Java .....	46
Conexión con la base de datos:.....	46
Inserción de personas: .....	47
Inserción de empleados: .....	48
Inserción de teléfonos .....	49
¿Opinión sobre los resultados obtenidos? .....	50
Conclusión.....	50



## Reto Asignado: Barbería

Barbería (Ejercicio A).

Una barbería desea llevar el **control** de sus **empleados** y de sus **clientes**, así como de los **servicios** que se prestan. Se desea almacenar la siguiente información:

**Empleados:** ID, cedula, Nombre, **Especialidad** (Masaje, Corte, Cejas, etc.)

**Clientes:** Datos personales (ID, cedula, Nombre, Profesión, Teléfono, correo, edad y Dirección).

**Historial de Servicios prestados** por la barbería: Un **registro** para **saber** información del **servicio prestado** por un **empleado** a un **cliente**, **productos consumidos**, **duración del procedimiento y fecha**.

**Citas:** **Fecha y Hora** en la que se **cita** al **cliente** y el **barbero** que **realizará** el **servicio**.

**Productos vendidos** por la barbería: **REF, Nombre, Cantidad y Precio**.

**Proveedor:** los productos **vendidos** deben tener una **fuentes**.

**Registro de Ventas:** Si un **barbero vende** un **producto** a un **cliente**, termina **obteniendo** una “**liga**” ganancia ocasional.

Se clasifica así:

- Las entidades con el color **rojo**
- Los atributos con el color **verde**
- Las relaciones con el color **azul**

## Modelo Entidad-Relación

El modelo permite establecer las siguientes entidades:

- Persona
- Cliente
- Empleado
- Persona
- Cita
- Producto
- Proveedor
- Compra
- Servicio
- Factura

- Historial de servicio

## Relaciones:

- Cliente-Reservar-Cita: Un cliente puede reservar muchas citas y una cita puede ser reservada por un cliente.
- Cliente-Acceder-Servicio atención: Un cliente puede acceder a muchos servicios y un servicio es prestado a un cliente.
- Cliente-Comprar-Producto compra: Un cliente puede comprar muchos productos y un producto puede ser comprado por un cliente.
- Cita-Requerir-Servicio atención: Una cita puede requerir muchos servicios y un servicio puede ser requerido por muchas citas.
- Empleado-Atender-Cita: Un empleado puede atender muchas citas Y una cita puede ser atendida por un empleado.
- Empleado-Vender-Producto Compra: Un vendedor puede vender muchos productos y un producto puede ser vendido por un vendedor.
- Empleado-Realizar-Servicio atención: Un empleado puede realizar muchos servicios y un servicio puede ser realizado por un empleado.
- Servicio-Incluir-Servicio atención: Un servicio puede ser incluido en muchos servicios de atención y en un servicio de atención solo puede incluir un servicio.
- Producto-Pertenecer-Producto compra: Un producto puede pertenecer a muchos productos compras y una compra solo puede incluir un producto compra.
- Proveedor-Vender-Producto: Un proveedor puede vender muchos productos y un producto solo es vendido por un proveedor.
- Factura-Pertenecer-Historial Servicio: Una factura puede pertenecer a un historial de servicios y un historial de servicio puede tener varias facturas.

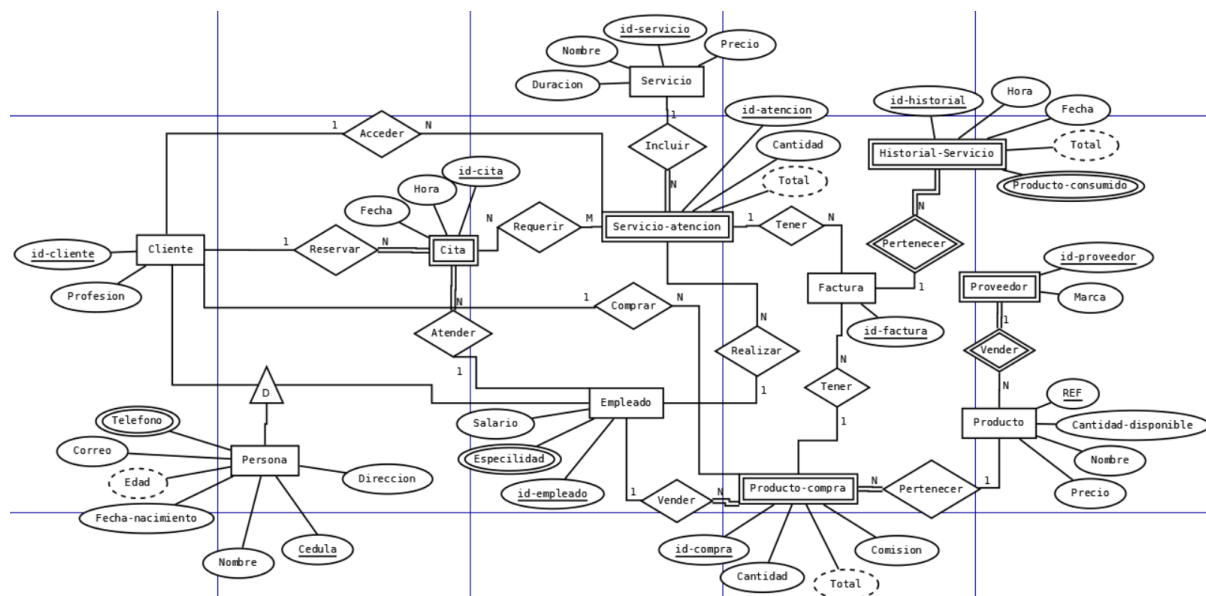
## Atributos:

- Persona
  - Cedula
  - Nombre
  - Dirección
  - Teléfono
  - Correo
  - Fecha de nacimiento

- Edad (derivado)
- Cliente
  - Profesión
- Empleado
  - Especialidad (multivaluado)
- Cita
  - Id
  - Fecha
  - Hora
- Producto
  - REF
  - Cantidad disponible
  - Nombre
  - Precio
- Proveedor
  - id
  - Marca
- Producto Compra
  - Id
  - Cantidad
  - Comisión
  - Total (derivado)
- Servicio
  - Id
  - Nombre
  - Precio
  - Duración
- Factura
  - Id
- Historial de servicio
  - Id
  - Fecha
  - Hora
  - Total
  - Productos consumidos (multivaluado)

Al identificar que tanto cliente como empleado comparten muchos atributos, se decidió por Crear una especialización para la entidad de empleado y la entidad cliente partiendo desde la entidad persona como una nueva entidad. De esta manera evitamos la repetición de datos y separamos los atributos de cliente y empleado.

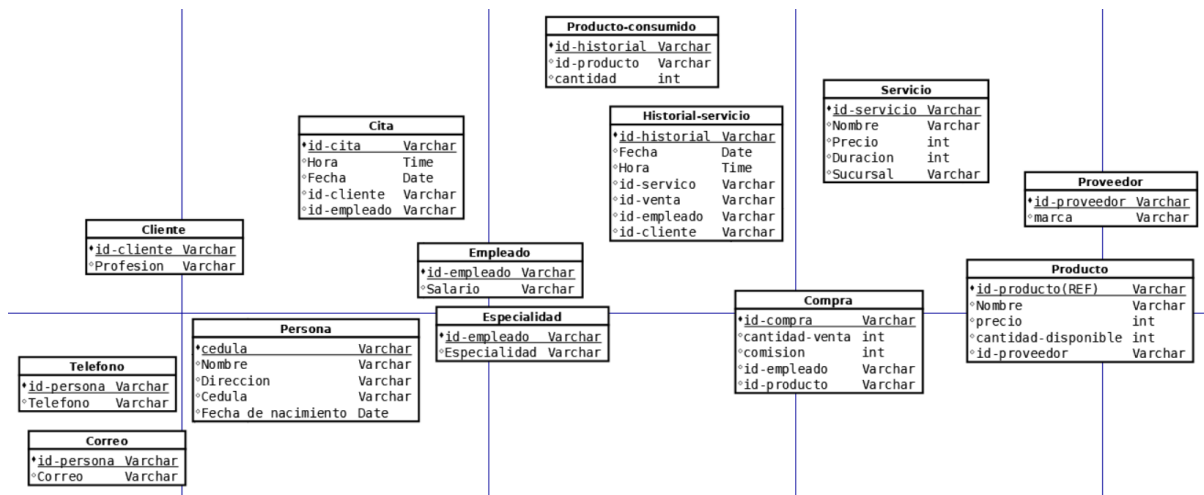
## Diagrama ER:



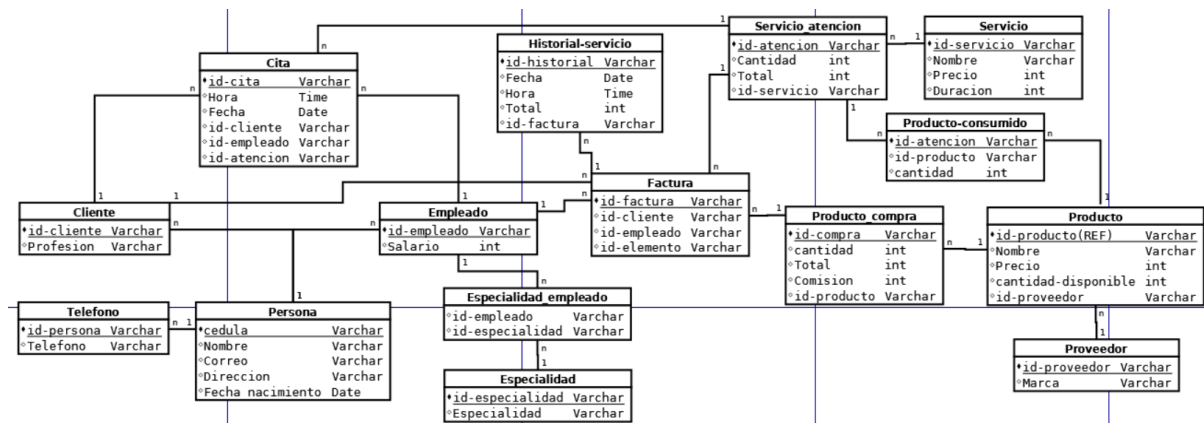
Se obtiene el siguiente diagrama en el cual se refleja las entidades y las relaciones que existen entre ellas, llevando cada entidad sus atributos los cuales son propios de cada una de las entidades, Además si tiene en cuenta su cardinalidad.

## Modelo-Relacional

Partiendo desde él el modelo entidad relación Realizado con anterioridad se puede tener una mejor perspectiva de cómo se van a guardar los datos. Teniendo en cuenta primeramente que las entidades se convierten en tabla y que los atributos multivaluados se transforman en tablas asociadas a estas entidades obtenemos lo siguiente:



Al restablecer las relaciones teniendo en cuenta su cardinalidad se obtienen la siguiente representación del diagrama:



Aquí se puede destacar que ha sido necesario crear una tabla intermedia entre empleado y especialidad ya que un empleado puede tener más de una especialidad y una especialidad puede estar en más de un empleado.

De la misma manera se puede destacar lo mismo en servicio atención y producto debido a un producto puede estar en muchos servicios y un servicio puede consumir muchos productos.

## Normalización del modelo:

Para la normalización del modelo se utilizaron las estrategias de llenar cada tabla con al menos 2 registros en un Excel, Para destacar como se relacionan Las entidades con sus llaves primarias y llaves foráneas, Obteniendo los siguientes registros:



Persona					
Cedula	nombre	fecha_nacimiento	dirección	edad	correo
c1	johan cifuentes	10/10/2001	calle 5	23	<a href="mailto:johan@gmail.com">johan@gmail.com</a>
c2	Julio	15/04/1981	calle 3	40	<a href="mailto:julio@gmail.com">julio@gmail.com</a>
c3	Cristian	11/11/2001	calle 10	23	<a href="mailto:crsmor@gmail.com">crsmor@gmail.com</a>

Cliente	
IdCliente	Profesión
c1	Manager
c2	Peluquero

Empleado	
IdEmpleado	Salario
c2	10000000
c3	10000000

Telefono	
Cedula	numero
c1	3143211201
c2	3112012254
c3	3112145980

Especialidad	
IdEspecialidad	nombre
es1	cejas
es2	barba

Especialidad_Empleado	
IdEmpleado	IdEspecialidad
c2	es1
c2	es2
c3	es1

Proveedor	
IdProveedor	Marca
prov1	marca1
prov2	marca2

Producto				
IdProducto	Nombre	IdProveedor	CantidadDisponible	Precio
pro1	Gel	prov1	100	1000
pro2	Cuchilla	prov1	200	200

Producto_compra				
idCompra	idProducto	cantidad	total	comision
com1	pro1	5	5000	500
com1	pro2	3	6000	600

Servicio			
IdServicio	nombre	precio	duracion
serv1	cejas	20000	20min
serv2	barba	28000	30min

Servicio_atencion			
IdAtencion	IdServicio	Cantidad	Total
at1	serv1	2	40000
at2	serv2	5	140000

Producto_consumido		
idAtencion	IdProducto	cantidad
at1	pro1	1
at1	pro2	2

Cita					
IdCita	IdCliente	IdEmpleado	id-atencion	Fecha	hora
ci1	c1	c2	at2	4/02/2024	8am
ci2	c2	c3	at1	5/02/2024	10am

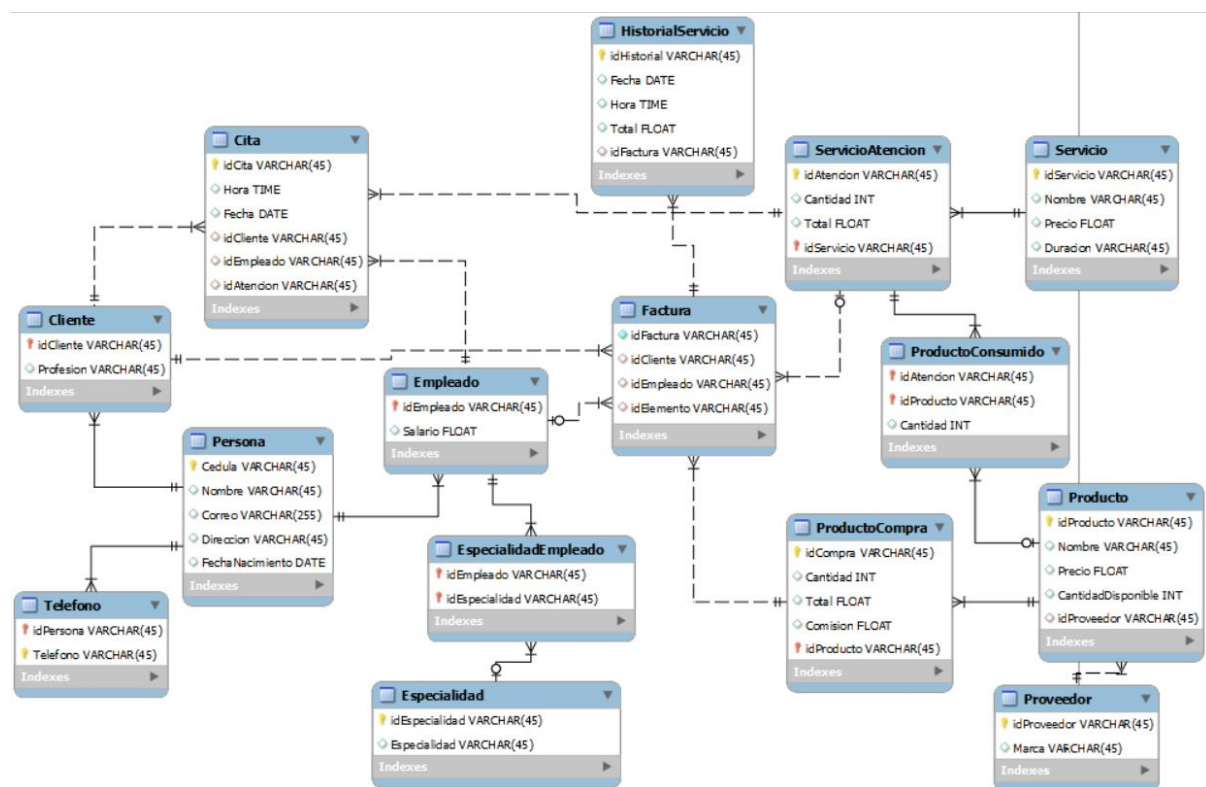
Factura			
IdFactura	IdCliente	IdEmpleado	idElemento
fac1	c1	c2	com1
fac1	c1	c3	at1

Historial_Servicio				
IdHistorial	IdFactura	Fecha	Hora	Total
hst1	fac1	4/02/2024	8am	com1 + at2

## Modelo diagramado en Workbench

Después de tener el diagrama del modelo relacional se procede a realizar el mismo en MYSQL para realizar posteriormente los comandos de creación de la tabla, Dado que el objetivo final es llevar estos diseños a una base de datos en donde se pueda guardar toda la información importante para la barbería.

De esta forma llevaron control adecuado de todos los servicios y productos consumidos durante estos servicios y los productos veintidós y las comisiones que los empleados reciben por la venta de cada producto, Se obtiene el siguiente diagrama como resultado:



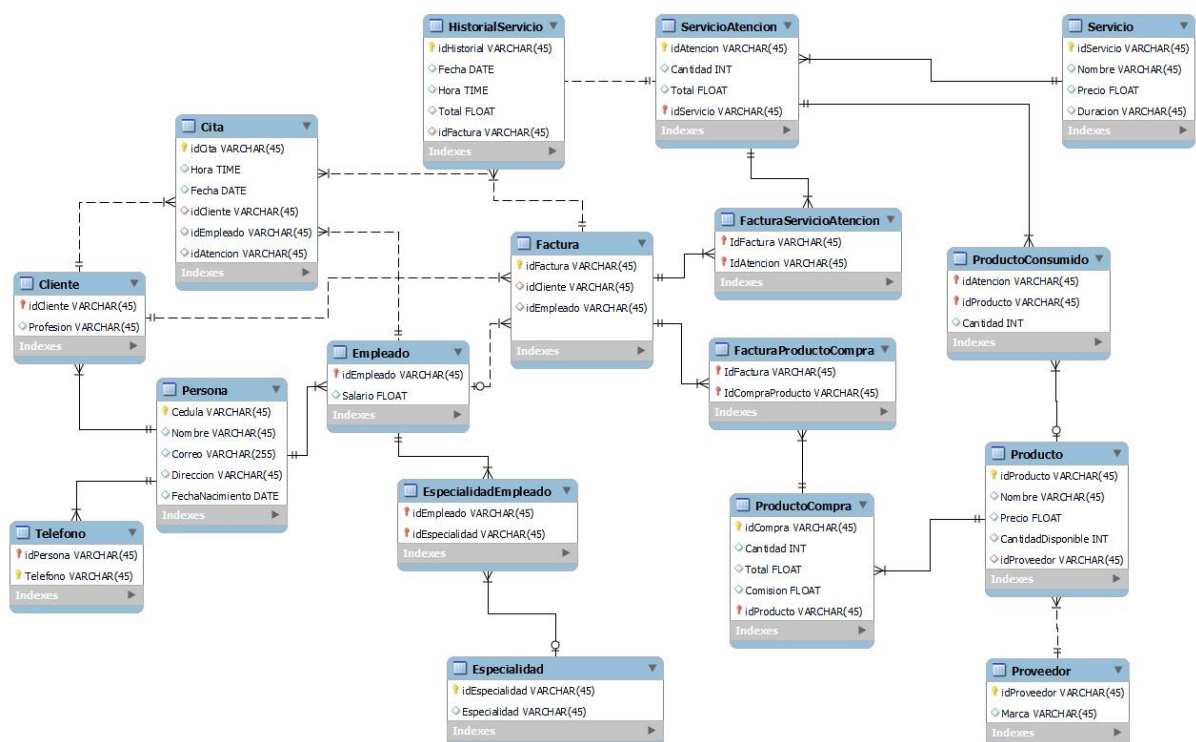
### Resolviendo conflicto:

Dado que los datos en la entidad factura de los productos comprados y de los servicios adquiridos por el cliente, Podría generar conflictos en la indexación en el ID elemento que relaciona a la entidad factura con las entidades mencionadas anteriormente, Se

decide crear dos tablas intermedias para mantener un registro más adecuado y un mejor control de toda esta información.

Debido a que se evidenció un error a relacionar dos tablas diferentes a un mismo atributo en la tabla de facturas, Todo esto al establecerlas como llaves foráneas lo que permitiría establecer relación entre estas entidades.

Por lo que se obtiene como resultado el siguiente diagrama final:



## Comandos SQL de creación de las Tablas:

A continuación, se escriben los comandos utilizados:

## Esquema barbería:

-- MySQL Workbench Forward Engineering

```

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

```

```

-----
-- Schema barberia
-----

```

```

-- Es la base de datos de la barberia
-----

```

```

-- Schema barberia
--

```

```

-- Es la base de datos de la barberia
-----

```

```

CREATE SCHEMA IF NOT EXISTS `barberia` DEFAULT CHARACTER SET utf8
COLLATE utf8_bin ;
USE `barberia` ;

```

```

-----
-- Table `barberia`.`Persona`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Persona` (
  `Cedula` VARCHAR(45) NOT NULL,
  `Nombre` VARCHAR(45) NULL,
  `Correo` VARCHAR(255) NULL,
  `Direccion` VARCHAR(45) NULL,
  `FechaNacimiento` DATE NULL,
  PRIMARY KEY (`Cedula`));

```

```

-----
-- Table `barberia`.`Telefono`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Telefono` (
  `idPersona` VARCHAR(45) NOT NULL,
  `Telefono` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idPersona`, `Telefono`),

```

```

CONSTRAINT `idPersona`
  FOREIGN KEY (`idPersona`)
    REFERENCES `barberia`.`Persona` (`Cedula`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Cliente`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Cliente` (
  `idCliente` VARCHAR(45) NOT NULL,
  `Profesion` VARCHAR(45) NULL,
  PRIMARY KEY (`idCliente`),
  CONSTRAINT `idCliente`
    FOREIGN KEY (`idCliente`)
      REFERENCES `barberia`.`Persona` (`Cedula`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Empleado`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Empleado` (
  `idEmpleado` VARCHAR(45) NOT NULL,
  `Salario` FLOAT NULL,
  PRIMARY KEY (`idEmpleado`),
  CONSTRAINT `fkEmpleado`
    FOREIGN KEY (`idEmpleado`)
      REFERENCES `barberia`.`Persona` (`Cedula`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Servicio`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Servicio` (
  `idServicio` VARCHAR(45) NOT NULL,
  `Nombre` VARCHAR(45) NULL,
  `Precio` FLOAT NULL,
  `Duracion` VARCHAR(45) NULL,
  PRIMARY KEY (`idServicio`))
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`ServicioAtencion`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`ServicioAtencion` (
  `idAtencion` VARCHAR(45) NOT NULL,
  `Cantidad` INT NULL,
  `Total` FLOAT NULL,
  `idServicio` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idAtencion`, `idServicio`),
  INDEX `idServicio_idx` (`idServicio` ASC) VISIBLE,
  CONSTRAINT `idServicio`
    FOREIGN KEY (`idServicio`)
      REFERENCES `barberia`.`Servicio` (`idServicio`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Cita`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Cita` (
  `idCita` VARCHAR(45) NOT NULL,
  `Hora` TIME NULL,
  `Fecha` DATE NULL,
  `idCliente` VARCHAR(45) NULL,
  `idEmpleado` VARCHAR(45) NULL,
  `idAtencion` VARCHAR(45) NULL,
  PRIMARY KEY (`idCita`),
  INDEX `idCliente_idx` (`idCliente` ASC) VISIBLE,
  INDEX `idEmpleado_idx` (`idEmpleado` ASC) VISIBLE,
  INDEX `idAtencion_idx` (`idAtencion` ASC) VISIBLE,

```

```

CONSTRAINT `fkCliente`
  FOREIGN KEY (`idCliente`)
  REFERENCES `barberia`.`Cliente` (`idCliente`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fkEmpleadoCi`
  FOREIGN KEY (`idEmpleado`)
  REFERENCES `barberia`.`Empleado` (`idEmpleado`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fkAtencion`
  FOREIGN KEY (`idAtencion`)
  REFERENCES `barberia`.`ServicioAtencion` (`idAtencion`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Especialidad`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Especialidad` (
  `idEspecialidad` VARCHAR(45) NOT NULL,
  `Especialidad` VARCHAR(45) NULL,
  PRIMARY KEY (`idEspecialidad`))
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`EspecialidadEmpleado`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`EspecialidadEmpleado` (
  `idEmpleado` VARCHAR(45) NOT NULL,
  `idEspecialidad` VARCHAR(45) NOT NULL,
  INDEX `idEspecialidad_idx` (`idEspecialidad` ASC) VISIBLE,
  PRIMARY KEY (`idEmpleado`, `idEspecialidad`),
  CONSTRAINT `fkEspecialidad`
    FOREIGN KEY (`idEspecialidad`)
    REFERENCES `barberia`.`Especialidad` (`idEspecialidad`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,

```



```

CONSTRAINT `fkEmpleadod`
  FOREIGN KEY (`idEmpleado`)
  REFERENCES `barberia`.`Empleado` (`idEmpleado`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Factura`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Factura` (
  `idFactura` VARCHAR(45) NOT NULL,
  `idCliente` VARCHAR(45) NULL,
  `idEmpleado` VARCHAR(45) NULL,
  PRIMARY KEY (`idFactura`),
  INDEX `idCliente_idx` (`idCliente` ASC) VISIBLE,
  INDEX `idEmpleado_idx` (`idEmpleado` ASC) VISIBLE,
  CONSTRAINT `fkClientef`
    FOREIGN KEY (`idCliente`)
    REFERENCES `barberia`.`Cliente` (`idCliente`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fkEmpleadof`
    FOREIGN KEY (`idEmpleado`)
    REFERENCES `barberia`.`Empleado` (`idEmpleado`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`HistorialServicio`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`HistorialServicio` (
  `idHistorial` VARCHAR(45) NOT NULL,
  `Fecha` DATE NULL,
  `Hora` TIME NULL,
  `Total` FLOAT NULL,
  `idFactura` VARCHAR(45) NULL,
  PRIMARY KEY (`idHistorial`),

```

```

INDEX `idFactura_idx` (`idFactura` ASC) VISIBLE,
CONSTRAINT `idFactura`
  FOREIGN KEY (`idFactura`)
  REFERENCES `barberia`.`Factura` (`idFactura`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Proveedor`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Proveedor` (
  `idProveedor` VARCHAR(45) NOT NULL,
  `Marca` VARCHAR(45) NULL,
  PRIMARY KEY (`idProveedor`))
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`Producto`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`Producto` (
  `idProducto` VARCHAR(45) NOT NULL,
  `Nombre` VARCHAR(45) NULL,
  `Precio` FLOAT NULL,
  `CantidadDisponible` INT NULL,
  `idProveedor` VARCHAR(45) NULL,
  PRIMARY KEY (`idProducto`),
  INDEX `idProveedor_idx` (`idProveedor` ASC) VISIBLE,
  CONSTRAINT `idProveedor`
    FOREIGN KEY (`idProveedor`)
    REFERENCES `barberia`.`Proveedor` (`idProveedor`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`ProductoCompra`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`ProductoCompra` (
  `idCompra` VARCHAR(45) NOT NULL,
  `Cantidad` INT NULL,
  `Total` FLOAT NULL,
  `Comision` FLOAT NULL,
  `idProducto` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`idCompra`, `idProducto`),
  INDEX `idProducto_idx` (`idProducto` ASC) VISIBLE,
  CONSTRAINT `idProducto`
    FOREIGN KEY (`idProducto`)
      REFERENCES `barberia`.`Producto` (`idProducto`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`ProductoConsumido`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`ProductoConsumido` (
  `idAtencion` VARCHAR(45) NOT NULL,
  `idProducto` VARCHAR(45) NOT NULL,
  `Cantidad` INT NULL,
  INDEX `idProducto_idx` (`idProducto` ASC) VISIBLE,
  PRIMARY KEY (`idAtencion`, `idProducto`),
  CONSTRAINT `fkAtencionP`
    FOREIGN KEY (`idAtencion`)
      REFERENCES `barberia`.`ServicioAtencion` (`idAtencion`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION,
  CONSTRAINT `fkProductoP`
    FOREIGN KEY (`idProducto`)
      REFERENCES `barberia`.`Producto` (`idProducto`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`FacturaServicioAtencion`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`FacturaServicioAtencion` (
  `IdFactura` VARCHAR(45) NOT NULL,
  `IdAtencion` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`IdFactura`, `IdAtencion`),
  INDEX `FK_AtencionA_idx` (`IdAtencion` ASC) VISIBLE,
  CONSTRAINT `FK_FacturaT`
    FOREIGN KEY (`IdFactura`)
      REFERENCES `barberia`.`Factura` (`idFactura`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `FK_AtencionA`
    FOREIGN KEY (`IdAtencion`)
      REFERENCES `barberia`.`ServicioAtencion` (`idAtencion`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

-----
-- Table `barberia`.`FacturaProductoCompra`
-----

```

```

CREATE TABLE IF NOT EXISTS `barberia`.`FacturaProductoCompra` (
  `IdFactura` VARCHAR(45) NOT NULL,
  `IdCompraProducto` VARCHAR(45) NOT NULL,
  PRIMARY KEY (`IdFactura`, `IdCompraProducto`),
  INDEX `FK_CompraC_idx` (`IdCompraProducto` ASC) VISIBLE,
  CONSTRAINT `FK_FacturaC`
    FOREIGN KEY (`IdFactura`)
      REFERENCES `barberia`.`Factura` (`idFactura`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION,
  CONSTRAINT `FK_CompraC`
    FOREIGN KEY (`IdCompraProducto`)
      REFERENCES `barberia`.`ProductoCompra` (`idCompra`)
      ON DELETE NO ACTION
      ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;

```

```
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## Poblado de base de datos:

```
USE barberia;
```

```
-- Insertar personas en la tabla Persona, seran 5 empleados y 15 clientes en total las personas
```

```
INSERT INTO Persona (Cedula, Nombre, Correo, Direccion, FechaNacimiento) VALUES  
( '1065832271', 'Julio Vasquez', 'julio@example.com', 'Calle 0, Ciudad 0', '1997-04-15'),  
( '1111111111', 'Juan Pérez', 'juan@example.com', 'Calle 1, Ciudad 1', '1990-01-01'),  
( '2222222222', 'María Rodríguez', 'maria@example.com', 'Calle 2, Ciudad 2', '1991-02-02'),  
( '3333333333', 'Carlos López', 'carlos@example.com', 'Calle 3, Ciudad 3', '1992-03-03'),  
( '4444444444', 'Ana Martínez', 'ana@example.com', 'Calle 4, Ciudad 4', '1993-04-04'),  
( '5555555555', 'Pedro García', 'pedro@example.com', 'Calle 5, Ciudad 5', '1994-05-05'),  
( '6666666666', 'Laura Fernández', 'laura@example.com', 'Calle 6, Ciudad 6', '1995-06-06'),  
( '7777777777', 'Miguel González', 'miguel@example.com', 'Calle 7, Ciudad 7', '1996-07-07'),  
( '8888888888', 'Sofía Díaz', 'sofia@example.com', 'Calle 8, Ciudad 8', '1997-08-08'),  
( '9999999999', 'Luis Sánchez', 'luis@example.com', 'Calle 9, Ciudad 9', '1998-09-09'),  
( '1010101010', 'Elena Ramírez', 'elena@example.com', 'Calle 10, Ciudad 10', '1999-10-10'),  
( '1212121212', 'David Herrera', 'david@example.com', 'Calle 11, Ciudad 11', '1980-11-11'),  
( '1313131313', 'Lucía Castro', 'lucia@example.com', 'Calle 12, Ciudad 12', '1981-12-12'),  
( '1414141414', 'Roberto Moreno', 'roberto@example.com', 'Calle 13, Ciudad 13', '1982-01-01'),  
( '1515151515', 'Carmen Ruiz', 'carmen@example.com', 'Calle 14, Ciudad 14', '1983-02-02'),
```

('1616161616', 'Sara López', 'sara@example.com', 'Calle 15, Ciudad 15', '1984-03-03'),  
 ('1717171717', 'Luz Petro', 'luzp@example.com', 'Calle 17, Ciudad 17', '1991-02-12'),  
 ('1818181818', 'Ronald Martínez', 'ronalmart@example.com', 'Calle 18, Ciudad 18', '1989-05-01'),  
 ('1919191919', 'Carolina Higuera', 'carolinaHig@example.com', 'Calle 19, Ciudad 19', '1973-09-02'),  
 ('2020202020', 'Clara Solano', 'clarasol@example.com', 'Calle 20, Ciudad 20', '1994-08-03'),  
 ('2121212121', 'Carlos Pérez', 'calorp@example.com', 'Calle 1, Ciudad 1', '1999-04-01'),  
 ('2323232323', 'Jose Rodríguez', 'joser@example.com', 'Calle 2, Ciudad 2', '1981-02-02'),  
 ('2424242424', 'Juan López', 'juanl@example.com', 'Calle 3, Ciudad 3', '1997-04-03'),  
 ('2525252525', 'Pablo Martínez', 'plabom@example.com', 'Calle 4, Ciudad 4', '1999-04-14'),  
 ('2626262626', 'Pedro Cortez', 'pedroc@example.com', 'Calle 5, Ciudad 5', '1993-05-08'),  
 ('2727272727', 'Lorena Fernández', 'lorenaf@example.com', 'Calle 6, Ciudad 6', '1985-07-05'),  
 ('2828282828', 'Dalena González', 'dalenag@example.com', 'Calle 7, Ciudad 7', '1995-07-06'),  
 ('2929292929', 'Luis Díaz', 'luisd@example.com', 'Calle 8, Ciudad 8', '1993-03-03'),  
 ('3030303030', 'Ramon Sánchez', 'ramons@example.com', 'Calle 9, Ciudad 9', '1988-08-07'),  
 ('3131313131', 'Moises Ramírez', 'moisesr@example.com', 'Calle 10, Ciudad 10', '1989-11-12');

-- Insertar números de teléfono para cada cliente en la tabla Telefono

INSERT INTO Telefono (idPersona, Telefono) VALUES

('1111111111', '555-1234'),  
 ('2222222222', '555-5678'),  
 ('3333333333', '555-2345'),  
 ('4444444444', '555-3456'),  
 ('5555555555', '555-6789'),

```
('6666666666', '555-4567'),  
('7777777777', '555-5678'),  
('8888888888', '555-6789'),  
('9999999999', '555-7890'),  
('1010101010', '555-8901'),  
('1212121212', '555-9012'),  
('1313131313', '555-0123'),  
('1414141414', '555-1111'),  
('1515151515', '555-2222'),  
('1065832271', '555-3154'),  
('1616161616', '555-3322'),  
('1212121212', '555-9013'),  
('1313131313', '555-1123'),  
('1414141414', '555-1881'),  
('1515151515', '555-2332'),  
('1065832271', '555-3186');
```

-- Insertar clientes en la tabla Cliente

```
INSERT INTO Cliente (idCliente, Profesion) VALUES
```

```
('1111111111', 'Abogado'),  
('2222222222', 'QA'),  
('3333333333', 'Arquitecto'),  
('4444444444', 'Electrico'),  
('5555555555', 'Ingeniero'),  
('6666666666', 'Tester'),  
('7777777777', 'Medico'),  
('8888888888', 'Empresario'),  
('9999999999', 'Anestesiologo'),
```

```
('1065832271', 'Electronico'),  
('1010101010', 'Juez'),  
('1212121212', 'Fiscal'),  
('1414141414', 'Contador'),  
('1515151515', 'Conductor'),  
('1616161616', 'Enfermera'),  
('1717171717', 'Soldador'),  
('1818181818', 'Emprendedor'),  
('1919191919', 'Independiente'),  
('2020202020', 'Medico'),  
('2121212121', 'Electronico'),  
('2323232323', 'Desarrollador');
```

-- Insertar empleados en la tabla Empleado

```
INSERT INTO Empleado (idEmpleado, Salario) VALUES
```

```
('1111111111', 2000),  
('2222222222', 2500),  
('3333333333', 3000),  
('4444444444', 3000),  
('5555555555', 4100),  
('1212121212', 2000),  
('1313131313', 2500),  
('1414141414', 3000),  
('1515151515', 3000),  
('1065832271', 4100),  
('2121212121', 2000),  
('2323232323', 2500),
```



('2424242424', 3000),  
('2525252525', 3000),  
('2626262626', 4100),  
('2727272727', 2000),  
('2828282828', 2500),  
('2929292929', 3000),  
('3030303030', 3000),  
('3131313131', 4100);

-- Insertar servicios en la tabla Servicio

INSERT INTO Servicio (idServicio, Nombre, Precio, Duracion) VALUES

('s1', 'Corte de cabello masculino', 150.00, 30),  
('s2', 'Corte de cabello femenino', 200.00, 45),  
('s3', 'Afeitado de barba', 100.00, 20),  
('s4', 'Tratamiento facial', 250.00, 60),  
('s5', 'Depilación de cejas', 50.00, 15),  
('s6', 'Manicura', 80.00, 30),  
('s7', 'Pedicura', 100.00, 45),  
('s8', 'Masaje capilar', 120.00, 40),  
('s9', 'Tinte de cabello', 180.00, 60),  
('s10', 'Alisado de cabello', 300.00, 90),  
('s11', 'Mascarilla facial', 150.00, 30),  
('s12', 'Corte de cabello alegre', 200.00, 45),  
('s13', 'Afeitado de barba + cejas', 100.00, 20),  
('s14', 'Tratamiento facial + barba', 250.00, 60),  
('s15', 'Rand Honors', 50.00, 15),  
('s16', 'Manicura + pedicura', 170.00, 30),  
('s17', 'Decoloración y Tinte de Cabello', 100.00, 45),

('s18', 'Masaje general', 120.00, 40),  
('s19', 'Tinte de barba', 180.00, 60),  
('s20', 'Corte + barba + cejas', 300.00, 90),  
('s21', 'Corte ´ cejas', 150.00, 45);

-- Insertar especialidades en la tabla Especialidad

INSERT INTO Especialidad (idEspecialidad, Especialidad) VALUES

('e1', 'Corte de cabello'),  
('e2', 'Afeitado de barba'),  
('e3', 'Tratamiento facial'),  
('e4', 'Depilación de cejas'),  
('e5', 'Manicura'),  
('e6', 'Pedicura'),  
('e7', 'Masaje capilar'),  
('e8', 'Tinte de cabello'),  
('e9', 'Alisado de cabello'),  
('e10', 'Peinado para eventos'),  
('e11', 'Corte degradado'),  
('e12', 'Afeitado de barba extrema'),  
('e13', 'Tratamiento facial especial'),  
('e14', 'Depilación de cejas con cera'),  
('e15', 'Manicura + pedicura'),  
('e16', 'Pedicura laser'),  
('e17', 'Masaje general'),  
('e18', 'Tinte de barba'),  
('e19', 'Mascarilla'),  
('e20', 'Mascarilla puntos negros');

-- Insertar asignaciones de especialidades a empleados

INSERT INTO EspecialidadEmpleado (idEmpleado, idEspecialidad) VALUES

('1111111111', 'e1'), ('1111111111', 'e2'),  
('2222222222', 'e3'), ('2222222222', 'e4'),  
('3333333333', 'e5'), ('3333333333', 'e6'),  
('4444444444', 'e7'), ('4444444444', 'e8'),  
('5555555555', 'e9'), ('5555555555', 'e10'),  
('1212121212', 'e11'), ('1212121212', 'e2'),  
('1313131313', 'e3'), ('1313131313', 'e14'),  
('1414141414', 'e15'), ('1414141414', 'e6'),  
('1515151515', 'e17'), ('1515151515', 'e8'),  
('2121212121', 'e19'), ('2121212121', 'e20'),  
('2323232323', 'e3'), ('2323232323', 'e14'),  
('2424242424', 'e5'), ('2424242424', 'e16'),  
('2525252525', 'e17'), ('2525252525', 'e8'),  
('2626262626', 'e1'), ('2626262626', 'e12'),  
('2727272727', 'e13'), ('2727272727', 'e14'),  
('2828282828', 'e15'), ('2828282828', 'e16'),  
('2929292929', 'e20'), ('2929292929', 'e8'),  
('3030303030', 'e2'), ('3030303030', 'e12'),  
('3131313131', 'e12'), ('3131313131', 'e5');

-- Insertar proveedores en la tabla Proveedor

INSERT INTO Proveedor (idProveedor, Marca) VALUES

('pv1', 'Marca A'),  
('pv2', 'Marca B'),  
('pv3', 'Marca C'),

('pv4', 'Marca D'),  
('pv5', 'Marca F'),  
('pv6', 'Marca G'),  
('pv7', 'Marca H'),  
('pv8', 'Marca I'),  
('pv9', 'Marca J'),  
('pv10', 'Marca K'),  
('pv11', 'Marca L'),  
('pv12', 'Marca M'),  
('pv13', 'Marca N'),  
('pv14', 'Marca O'),  
('pv15', 'Marca P'),  
('pv16', 'Marca Q'),  
('pv17', 'Marca R'),  
('pv18', 'Marca S'),  
('pv19', 'Marca T'),  
('pv20', 'Marca U'),  
('pv21', 'Marca V');

-- Insertar productos en la tabla Producto

INSERT INTO Producto (idProducto, Nombre, Precio, CantidadDisponible, idProveedor)  
VALUES

('pp1', 'Gel para cabello', 10, 100, 'pv1'),  
('pp2', 'Cuchillas de afeitar', 5, 200, 'pv2'),  
('pp3', 'Maquina de afeitar', 50, 50, 'pv3'),  
('pp4', 'Tijeras para cortar cabello', 20, 150, 'pv1'),  
('pp5', 'Crema para después de afeitar', 15, 100, 'pv2'),  
('pp6', 'Cepillo para barba', 8, 120, 'pv3'),

```

('pp7', 'Shampoo para barba', 12, 80, 'pv1'),
('pp8', 'Aceite para barba', 18, 90, 'pv2'),
('pp9', 'Espuma de afeitar', 7, 180, 'pv3'),
('pp10', 'Gel para Barba', 20, 100, 'pv10'),
('pp11', 'Cuchillas para depilar', 5, 300, 'pv12'),
('pp12', 'Maquina de motilar', 500, 50, 'pv13'),
('pp13', 'Tijeras de lujo', 200, 30, 'pv11'),
('pp14', 'Crema para depilar', 15, 180, 'pv14'),
('pp15', 'Cepillo para cejas', 8, 120, 'pv15'),
('pp16', 'Shampoo para cejas', 12, 80, 'pv16'),
('pp17', 'Aceite para cejas', 18, 90, 'pv17'),
('pp18', 'Espuma para cejas', 7, 120, 'pv18'),
('pp19', 'Peine para cejas', 5, 200, 'pv19'),
('pp20', 'Espuma para cejas', 7, 170, 'pv3');

```

-- Insertar compras en la tabla ProductoCompra

```

INSERT INTO ProductoCompra (idCompra, Cantidad, Total, Comision, idProducto)
VALUES

```

```

('p1', 20, (SELECT Precio * 20 FROM Producto WHERE idProducto = 'pp1'), 10, 'pp1'),
('p2', 15, (SELECT Precio * 15 FROM Producto WHERE idProducto = 'pp2'), 10, 'pp2'),
('p3', 30, (SELECT Precio * 30 FROM Producto WHERE idProducto = 'pp3'), 10, 'pp3'),
('p4', 25, (SELECT Precio * 25 FROM Producto WHERE idProducto = 'pp4'), 10, 'pp4'),
('p5', 20, (SELECT Precio * 20 FROM Producto WHERE idProducto = 'pp5'), 10, 'pp5'),
('p6', 15, (SELECT Precio * 15 FROM Producto WHERE idProducto = 'pp6'), 10, 'pp6'),
('p7', 30, (SELECT Precio * 30 FROM Producto WHERE idProducto = 'pp7'), 10, 'pp7'),
('p8', 25, (SELECT Precio * 25 FROM Producto WHERE idProducto = 'pp8'), 10, 'pp8'),
('p9', 20, (SELECT Precio * 20 FROM Producto WHERE idProducto = 'pp9'), 10, 'pp9'),
('p10', 15, (SELECT Precio * 15 FROM Producto WHERE idProducto = 'pp10'), 10, 'pp10'),

```

```

('p11', 30, (SELECT Precio * 30 FROM Producto WHERE idProducto = 'pp11'), 10, 'pp11'),
('p12', 25, (SELECT Precio * 25 FROM Producto WHERE idProducto = 'pp12'), 10, 'pp12'),
('p13', 20, (SELECT Precio * 20 FROM Producto WHERE idProducto = 'pp13'), 10, 'pp13'),
('p14', 15, (SELECT Precio * 15 FROM Producto WHERE idProducto = 'pp14'), 10, 'pp14'),
('p15', 30, (SELECT Precio * 30 FROM Producto WHERE idProducto = 'pp15'), 10, 'pp15'),
('p16', 25, (SELECT Precio * 25 FROM Producto WHERE idProducto = 'pp16'), 10, 'pp16'),
('p17', 20, (SELECT Precio * 20 FROM Producto WHERE idProducto = 'pp17'), 10, 'pp17'),
('p18', 15, (SELECT Precio * 15 FROM Producto WHERE idProducto = 'pp18'), 10, 'pp18'),
('p19', 30, (SELECT Precio * 30 FROM Producto WHERE idProducto = 'pp19'), 10, 'pp19'),
('p20', 25, (SELECT Precio * 25 FROM Producto WHERE idProducto = 'pp20'), 10, 'pp20'),
('p21', 10, (SELECT Precio * 10 FROM Producto WHERE idProducto = 'pp5'), 10, 'pp5');

```

-- Insertar registros en la tabla ServicioAtencion

```

INSERT INTO ServicioAtencion (idAtencion, Cantidad, Total, idServicio) VALUES

```

```

('a1', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's1'), 's1'),
('a2', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's2'), 's2'),
('a3', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's3'), 's3'),
('a4', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's4'), 's4'),
('a5', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's5'), 's5'),
('a6', 3, (SELECT Precio * 3 FROM Servicio WHERE idServicio = 's6'), 's6'),
('a7', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's7'), 's7'),
('a8', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's8'), 's8'),
('a9', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's9'), 's9'),
('a10', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's10'), 's10'),
('a11', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's11'), 's11'),
('a12', 3, (SELECT Precio * 3 FROM Servicio WHERE idServicio = 's12'), 's12'),
('a13', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's13'), 's13'),
('a14', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's14'), 's14'),

```

```

('a15', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's15'), 's15'),
('a16', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's16'), 's16'),
('a17', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's17'), 's17'),
('a18', 3, (SELECT Precio * 3 FROM Servicio WHERE idServicio = 's18'), 's18'),
('a19', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's19'), 's19'),
('a20', 1, (SELECT Precio * 1 FROM Servicio WHERE idServicio = 's20'), 's20'),
('a21', 3, (SELECT Precio * 3 FROM Servicio WHERE idServicio = 's4'), 's4'),
('a22', 2, (SELECT Precio * 2 FROM Servicio WHERE idServicio = 's5'), 's5');

```

-- Insertar registros en la tabla Factura

```

INSERT INTO Factura (idFactura, idCliente, idEmpleado) VALUES

```

```

('f1','6666666666','1212121212'),
('f2','7777777777','1515151515'),
('f3','8888888888','2121212121'),
('f4','9999999999','3030303030'),
('f5','1010101010','1313131313'),
('f6','1616161616','4444444444'),
('f7','1717171717','2424242424'),
('f8','1818181818','3131313131'),
('f9','1919191919','3131313131'),
('f10','1717171717','1515151515');

```

```

INSERT INTO `barberia`.`facturaproductocompra` (`IdFactura`,
`IdCompraProducto`) VALUES

```

```

('f1','p1'),
('f1','p2'),

```

('f2','p3'),  
('f2','p4'),  
('f3','p5'),  
('f3','p6'),  
('f4','p7'),  
('f4','p8'),  
('f5','p9'),  
('f5','p10'),  
('f6','p11'),  
('f6','p12'),  
('f7','p13'),  
('f7','p14'),  
('f8','p15'),  
('f8','p16'),  
('f9','p17'),  
('f9','p18'),  
('f10','p19'),  
('f10','p20');

INSERT INTO `barberia`.`facturaservicioatencion` (`IdFactura`,`IdAtencion`)

VALUES

('f1','a1'),  
('f1','a2'),  
('f2','a3'),  
('f2','a4'),  
('f3','a5'),  
('f3','a6'),



```
('f4','a7'),  
('f4','a8'),  
('f5','a9'),  
('f5','a10'),  
('f6','a11'),  
('f6','a12'),  
('f7','a13'),  
('f7','a14'),  
('f8','a15'),  
('f8','a16'),  
('f9','a17'),  
('f9','a18'),  
('f10','a19'),  
('f10','a20');
```

-- Insertar registros en la tabla ProductoConsumido

```
INSERT INTO ProductoConsumido (idAtencion, idProducto, Cantidad) VALUES
```

```
('a1','pp1',3),  
('a2','pp2',5),  
('a3','pp1',10),  
('a4','pp3',3),  
('a5','pp4',1),  
('a6','pp5',4),  
('a7','pp6',2),  
('a8','pp7',7),  
('a9','pp8',20),  
('a10','pp9',10),
```

```

('a11','pp1',40),
('a12','pp10',30),
('a13','pp11',7),
('a14','pp12',14),
('a15','pp17',3),
('a15','pp4',30),
('a16','pp13',10),
('a17','pp18',24),
('a18','pp2',3),
('a19','pp4',5),
('a20','pp6',2);

```

-- Insertar registros en la tabla HistorialServicio

```

INSERT INTO HistorialServicio (idHistorial, Fecha, Hora, Total, idFactura) VALUES
('h1','2023-02-03','15:30:01',30000, 'f1'),
('h2','2023-02-04','08:00:01',20000, 'f2'),
('h3','2023-02-05','16:30:01',40000, 'f3'),
('h4','2023-02-06','18:35:01',50000, 'f4'),
('h5','2023-02-07','19:30:01',30000, 'f5'),
('h6','2023-02-08','20:00:00',20000, 'f6'),
('h7','2023-02-09','15:30:01',50000, 'f7'),
('h8','2023-02-11','13:30:01',20000, 'f9'),
('h9','2023-02-12','15:30:01',30000, 'f10');

```

-- Insertar registros en la tabla Cita

```

INSERT INTO Cita (idCita, Fecha, Hora, idCliente, idEmpleado, idAtencion) VALUES
('ct1','2023-02-03','15:30:01', '7777777777', '1515151515', 'a2'),
('ct2','2023-02-04','15:30:01', '8888888888', '2121212121', 'a3'),
('ct3','2023-02-05','15:30:01', '9999999999', '1515151515', 'a4'),

```

('ct4','2023-02-06','15:30:01','1010101010','1515151515','a3'),  
('ct5','2023-02-07','15:30:01','6666666666','2121212121','a5'),  
('ct6','2023-02-08','15:30:01','1616161616','1515151515','a6'),  
('ct7','2023-04-03','15:30:01','1717171717','1515151515','a7'),  
('ct8','2023-04-04','15:30:01','1818181818','2121212121','a8'),  
('ct9','2023-04-05','15:30:01','1919191919','1515151515','a9'),  
('ct10','2023-04-06','15:30:01','2020202020','2121212121','a10'),  
('ct11','2023-04-07','15:30:01','1111111111','2121212121','a11'),  
('ct12','2023-04-08','15:30:01','2222222222','1515151515','a12'),  
('ct13','2023-04-09','15:30:01','3333333333','1515151515','a14'),  
('ct14','2023-04-10','15:30:01','4444444444','2121212121','a15'),  
('ct15','2023-04-11','15:30:01','5555555555','1515151515','a16'),  
('ct16','2023-04-12','15:30:01','1212121212','3131313131','a17'),  
('ct17','2023-04-14','15:30:01','1414141414','1515151515','a18'),  
('ct18','2023-04-15','15:30:01','1818181818','3030303030','a19'),  
('ct19','2023-04-16','15:30:01','1212121212','1515151515','a7'),  
('ct20','2023-04-17','15:30:01','1414141414','2323232323','a8'),  
('ct21','2023-04-18','15:30:01','4444444444','2323232323','a9');

## Consultas de base de datos

```
use `barberia`;
```

```
select * from `cita`;
```

```
select * from `cliente`;
```

```
select * from `empleado`;
```

```

select * from `especialidad`;
select * from `especialidadempleado`;
select * from `factura`;
select * from `historialservicio`;
select * from `persona`;
select * from `producto`;
select * from `productocompra`;
select * from `productoconsumido`;
select * from `proveedor`;
select * from `servicio`;
select * from `servicioatencion`;
select * from `telefono`;

```

## Vistas:

*Vista 1:* La siguiente vista muestra a los empleados con cantidad de servicios ordenados de mayor a menor cantidad. Le permite al administrador de la barbería ver cuáles empleados son los que más trabajan

### *Script:*

```

CREATE VIEW ServiciosPorEmpleados AS

SELECT

    E.idEmpleado

    ,P.Nombre

    ,COUNT(E.idEmpleado) AS ServiciosRealizados

FROM empleado E

INNER JOIN Persona P ON (E.idEmpleado = P.cedula)

INNER JOIN Factura F ON E.idEmpleado = F.idEmpleado

INNER JOIN facturaseservicioatencion FSA ON F.IdFactura = FSA.IdFactura

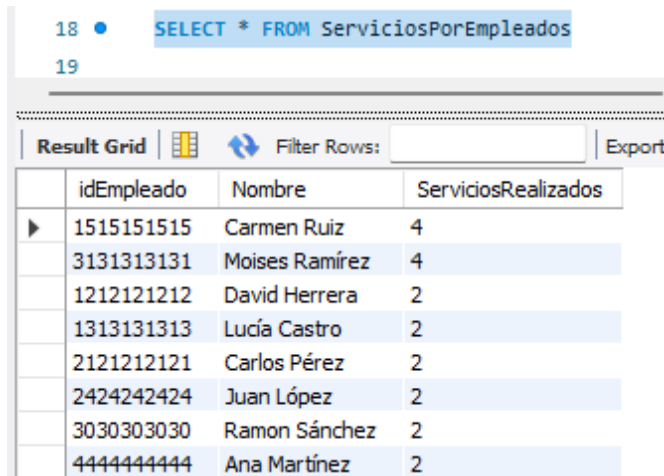
INNER JOIN servicioatencion SA ON FSA.IdAtencion = SA.IdAtencion

```

INNER JOIN servicio S ON (s.idServicio = SA.idServicio)

GROUP BY E.idEmpleado ORDER BY 3 DESC;

*Ejemplo resultado:*



18 • `SELECT * FROM ServiciosPorEmpleados`

19

Result Grid | Filter Rows: | Export

	idEmpleado	Nombre	ServiciosRealizados
▶	1515151515	Carmen Ruiz	4
	3131313131	Moises Ramírez	4
	1212121212	David Herrera	2
	1313131313	Lucía Castro	2
	2121212121	Carlos Pérez	2
	2424242424	Juan López	2
	3030303030	Ramon Sánchez	2
	4444444444	Ana Martínez	2

*Vista 2:* La siguiente vista muestra la cantidad de productos consumidos por tipo de servicio, esto nos muestra que servicios usan que componentes y que tanto, útil para administrar correctamente el inventario

*Script:*

CREATE VIEW ProductoMasConsumido AS

SELECT

S.nombre AS NombreServicio

,p.Nombre AS NombreProducto

,SUM(PC.cantidad) AS CantidadUsada

FROM Producto P

INNER JOIN ProductoConsumido PC ON P.idProducto = PC.idProducto

INNER JOIN ServicioAtencion SA ON SA.IdAtencion = PC.IdAtencion

INNER JOIN Servicio S ON S.IdServicio = SA.IdServicio

GROUP BY S.Nombre, P.Nombre;

*Ejemplo resultado:*

52 • `SELECT * FROM VentasPorEmpleado`

53 ↺ /\*

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

	Cedula	Nombre empleado	Productos vendidos
▶	1515151515	Carmen Ruiz	110
	3131313131	Moises Ramírez	90
	3030303030	Ramon Sánchez	55
	4444444444	Ana Martínez	55
	1212121212	David Herrera	35
	1313131313	Lucía Castro	35
	2121212121	Carlos Pérez	35
	2424242424	Juan López	35

*Vista 3:* Esta vista se creó con la intención de mostrar los productos comprados por un cliente, el precio, el producto y la cantidad total tiene la intención de simular una factura de simplemente los productos comprados

*Script:*

```
SELECT
    F.idFactura
    ,PCli.nombre AS "Nombre Cliente"
    ,PE.nombre AS "Nombre Empleado"
    ,HS.Fecha
    ,P.Nombre AS "Nombre Producto"
    ,PC.Cantidad AS "Cantidad Producto Comprada"
    ,P.Precio AS "Precio Por unidad"
    ,PC.Cantidad * P.Precio AS "Total Compra"
FROM Factura F
INNER JOIN historialservicio HS ON HS.idFactura = F.idFactura
INNER JOIN facturaproductocompra FPC ON F.idFactura = FPC.idFactura
INNER JOIN Persona AS PE ON PE.cedula = F.idEmpleado
INNER JOIN Persona AS PCli ON PCli.cedula = F.idCliente
INNER JOIN productocompra PC ON PC.idCompra = FPC.IdCompraProducto
INNER JOIN producto P ON P.idProducto = PC.idProducto
ORDER BY 1;
```


### Ejemplo resultado:


80 • `SELECT * FROM factura_productos_comprados WHERE IdFactura = 'f1';`

81

Result Grid

Filter Rows:

Export: 

Wrap Cell Content: 

	idFactura	Nombre Cliente	Nombre Empleado	Fecha	Nombre Producto	Cantidad Producto Comprada	Precio Por unidad	Total Compra
▶	f1	Laura Fernández	David Herrera	2023-02-03	Gel para cabello	20	10	200
	f1	Laura Fernández	David Herrera	2023-02-03	Cuchillas de afeitar	15	5	75

**Vista 4:** Esta vista tiene la finalidad de simular una factura con los servicios adquiridos, la cantidad, precio y monto total esta vista sirve para reflejar el gasto de un cliente simplemente incluyendo los servicios

### Script:

```
CREATE VIEW factura_servicios_consumidos
AS
SELECT
    F.idFactura
    ,PCli.nombre AS "Nombre Cliente"
    ,PE.nombre AS "Nombre Empleado"
    ,HS.Fecha
    ,S.Nombre AS "Nombre Servicio"
    ,SA.Cantidad AS "Cantidad Servicio Adquirido"
    ,S.Precio AS "Precio Servicio"
    ,SA.Cantidad * S.precio AS "Total Servicio Adquirido"
FROM Factura F
INNER JOIN historialservicio HS ON HS.idFactura = F.idFactura
INNER JOIN Persona AS PE ON PE.cedula = F.idEmpleado
INNER JOIN Persona AS PCli ON PCli.cedula = F.idCliente
INNER JOIN facturaserVICIOatencion FSA ON F.idFactura = FSA.IdFactura
INNER JOIN servicioatencion SA ON SA.IdAtencion = FSA.IdAtencion
INNER JOIN servicio S ON S.idServicio = SA.idServicio
ORDER BY 1;
```

### Ejemplo resultado:

```
80 • SELECT * FROM factura_servicios_consumidos WHERE IdFactura = 'f1';
```

```
81
```

```
82
```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:								
	idFactura	Nombre Cliente	Nombre Empleado	Fecha	Nombre Servicio	Cantidad Servicio Adquirido	Precio Servicio	Total Servi Adquirido
▶	f1	Laura Fernández	David Herrera	2023-02-03	Corte de cabello masculino	2	150	300
	f1	Laura Fernández	David Herrera	2023-02-03	Corte de cabello femenino	1	200	200

### Vista 5:

Esta vista pretende simular el precio total de los productos y servicios adquiridos por cada cliente. Trabaja en conjunto con las 2 anteriores vistas

### Script:

```
CREATE VIEW `factura_servicios_productos_comprados_consumidos`
```

```
AS
```

```
SELECT
```

```
    FPC.IdFactura
```

```
    ,FPC.`Nombre Cliente`
```

```
    ,FPC.`Nombre Empleado`
```

```
    ,SUM(FPC.`Total Compra` + FSC.`Total Servicio Adquirido`) AS "Monto Total"
```

```
FROM factura_servicios_consumidos FSC
```

```
INNER JOIN factura_productos_comprados FPC ON FSC.IdFactura = FPC.IdFactura
```

```
GROUP BY FPC.IdFactura
```

### Ejemplo resultado:

```
80 • SELECT * FROM factura_servicios_productos_comprados_consumidos;
```

```
81
```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:				
	IdFactura	Nombre Cliente	Nombre Empleado	Monto Total
▶	f1	Laura Fernández	David Herrera	1550
	f10	Luz Petro	Carmen Ruiz	1970
	f2	Miguel González	Carmen Ruiz	5400
	f3	Sofía Díaz	Carlos Pérez	1420
	f4	Luis Sánchez	Ramon Sánchez	2260
	f5	Elena Ramírez	Lucía Castro	2800
	f6	Sara López	Ana Martínez	26800
	f7	Luz Petro	Juan López	9350

### Vista 6:



La siguiente vista muestra la cantidad de productos consumidos por tipo de servicio, esto nos muestra que servicios usan que componentes y que tanto, útil para administrar correctamente el inventario

### Script:

```
CREATE VIEW ProductoMasConsumido AS

SELECT

    S.nombre AS NombreServicio

    ,p.Nombre AS NombreProducto

    ,SUM(PC.cantidad) AS CantidadUsada

FROM Producto P

INNER JOIN ProductoConsumido PC ON P.idProducto = PC.idProducto

INNER JOIN ServicioAtencion SA ON SA.IdAtencion = PC.IdAtencion

INNER JOIN Servicio S ON S.IdServicio = SA.IdServicio

GROUP BY S.Nombre, P.Nombre;
```

### Ejemplo resultado:

52 • `SELECT * FROM ProductoMasConsumido`

53 ☺ /\*

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

	NombreServicio	NombreProducto	CantidadUsada
▶	Corte de cabello masculino	Gel para cabello	3
	Mascarilla facial	Gel para cabello	40
	Afeitado de barba	Gel para cabello	10
	Corte de cabello alegre	Gel para Barba	30
	Afeitado de barba + cejas	Cuchillas para depilar	7
	Tratamiento facial + barba	Maquina de motilar	14
	Manicura + pedicura	Tijeras de lujo	10
	Rand Honors	Aceite para cejas	3
	Decoloración y Tinte de Cabello	Espuma para cejas	24

### Triggers:

-- Tabla: control\_de\_cambios\_barberia

```
CREATE TABLE control_de_cambios_barberia (  
    usuario VARCHAR(100),  
    accion VARCHAR(100),  
    fecha DATETIME  
);
```

#### *Creación del 1er Trigger: Control de eliminación*

Este trigger registra en la tabla 'control\_de\_cambios\_barberia' cada vez que se inserta un nuevo producto en la tabla 'Producto'.

#### *Ejemplo de uso:*

```
CALL sp_eliminar_producto('pp9');
```

```
SELECT * FROM control_de_cambios_barberia;
```

#### *Script:*

```
DELIMITER //  
  
CREATE TRIGGER controlEliminacionProducto  
AFTER INSERT ON producto  
FOR EACH ROW  
BEGIN  
    INSERT INTO control_de_cambios_barberia (usuario, accion, fecha)  
    VALUES (USER(), 'Delete Product', NOW());  
END;  
  
//  
DELIMITER ;
```

### *Creación del segundo Trigger: Control de creación de cita*

Este trigger registra en la tabla 'control\_de\_cambios\_barberia' cada vez que se inserta un nuevo registro en la tabla 'cita'.

### *Ejemplo de uso:*

```
CALL CrearCita('ct4','15:30:01', '2023-02-03', '4444444444', '1515151515', 'a3');
```

```
SELECT * FROM control_de_cambios_barberia;
```

### *Script:*

```
DELIMITER //

CREATE TRIGGER tr_crear_cita

AFTER INSERT ON cita

FOR EACH ROW

BEGIN

    INSERT INTO control_de_cambios_barberia (usuario, accion)

    VALUES (USER(), 'Creación cita');

END //

DELIMITER ;
```

## **Procedimientos:**

### *SP 1: Agregar producto*

Este procedimiento almacenado pretende agregar un nuevo producto a la tabla 'Producto' con los detalles proporcionados.

Llamada al procedimiento almacenado:

### *Como usar:*

```
CALL sp_Agregar_producto('pp11', 'tijeras especiales', 12.99, 3, 'pv1');
```

### *Script:*

```
DELIMITER //
```

```

CREATE PROCEDURE sp_Agregar_producto
    (IN IdProductoN VARCHAR(45),
    IN NombreProductoN VARCHAR(45),
    IN PrecioProductoN FLOAT,
    IN CantidadDisponibleN INT,
    IN idProveedorN VARCHAR(45))
BEGIN
    INSERT INTO Producto (idProducto, Nombre, Precio, CantidadDisponible, idProveedor) VALUES
    (IdProductoN, NombreProductoN, PrecioProductoN, CantidadDisponibleN, idProveedorN);
END //

DELIMITER ;

```

### *SP 2: Eliminar producto*

Este procedimiento almacenado tiene como objetivo eliminar un producto de la tabla 'Producto' según el ID proporcionado.

Llamada al procedimiento almacenado:

#### *Como usar:*

```
CALL sp_eliminar_producto('pp11');
```

#### *Script:*

```

DELIMITER //

CREATE PROCEDURE sp_eliminar_producto
    (IN IdProductoD VARCHAR(45))
BEGIN
    DELETE FROM producto
    WHERE idProducto = idProductoD;
END //

```

DELIMITER ;

### *SP 3: Crear cita*

Este procedimiento almacenado tiene como objetivo crear una nueva cita.

Llamada al procedimiento almacenado:

#### *Como usar:*

CALL CrearCita('ct3','15:30:01', '2023-02-03', '44444444444', '1515151515', 'a4');

#### *Script:*

DELIMITER //

```
CREATE PROCEDURE CrearCita(
    IN idCita VARCHAR(45),
    IN Hora TIME,
    IN Fecha DATE,
    IN idCliente VARCHAR(45),
    IN idEmpleado VARCHAR(45),
    IN idAtencion VARCHAR(45)
)
BEGIN
    INSERT INTO cita (idCita, Hora, Fecha, idCliente, idEmpleado, idAtencion)
    VALUES (idCita, Hora, Fecha, idCliente, idEmpleado, idAtencion);
END //
```

DELIMITER ;

#### *SP 4: Borrar cita*

Este procedimiento almacenado pretende eliminar una cita existente.

Llamada al procedimiento almacenado:

#### *Como usar:*

CALL BorrarCita('ct3');

#### *Script:*

```
DELIMITER //
```

```
CREATE PROCEDURE BorrarCita(
```

```
    IN idCitaArg VARCHAR(45)
```

```
)
```

```
BEGIN
```

```
    DELETE FROM cita
```

```
    WHERE idCita = idCitaArg;
```

```
END //
```

```
DELIMITER ;
```

## Conexión desde Java

### Conexión con la base de datos:

Se establece la URL para la conexión a la base de datos, se establecen el usuario y contraseña relacionada con la BD de interés y finalmente se llama Clase Faker para la generación de datos aleatorios

```

Cristian +1 *
public class Main {

    1 usage
    private static final String URL = "jdbc:mysql://localhost:3306/barberia";
    1 usage
    private static final String USER = "root";
    1 usage
    private static final String PASSWORD = "johan2233Z.";
    7 usages
    private static Connection conn;
    9 usages
    private static Faker faker = new Faker();

```

## Inserción de personas:

Se crea la función insertIntoPersona la cual tiene la funcionalidad de insertar personas en la base de datos a la tabla 'Persona', utilizando Faker para generar nombres, cedulas, correos, direcciones y fechas de nacimiento totalmente aleatorias, finalmente se ejecuta el script.

```

public static void insertIntoPersona() throws SQLException {
    String sql = "INSERT INTO Persona (Cedula, Nombre, Correo, Direccion, FechaNacimiento) " +
        "VALUES (?, ?, ?, ?, ?)";
    PreparedStatement stmt = conn.prepareStatement(sql);
    for (int i = 0; i < 50; i++) { // Inserting 2 records in this example
        // Generate fake data
        String cedula = faker.number().digits(count: 10);
        cedulasPersonas.add(cedula);
        String nombre = faker.name().fullName();
        String correo = faker.internet().emailAddress();
        String direccion = faker.address().streetAddress();
        SimpleDateFormat sdf = new SimpleDateFormat(pattern: "yyyy-MM-dd");
        Date fechaNacimiento = java.sql.Date.valueOf(sdf.format(faker.date().birthday()));

        // Set values to the prepared statement
        stmt.setString(parameterIndex: 1, cedula);
        stmt.setString(parameterIndex: 2, nombre);
        stmt.setString(parameterIndex: 3, correo);
        stmt.setString(parameterIndex: 4, direccion);
        stmt.setDate(parameterIndex: 5, fechaNacimiento);

        // Execute the statement
        stmt.executeUpdate();
    }
}

```

Obteniendo los siguientes resultados:

Result Grid					
Filter Rows:					
Edit: Export/Import: Wrap Cell Content:					
Cedula	Nombre	Correo	Direccion	FechaNacimiento	
0044708610	Lupe Dickinson MD	tuyet.auer@yahoo.com	54588 Joesph Cirde	1988-03-16	
0067995511	Virgina Huel	rolland.okeefe@gmail.com	867 Guadalupe Spur	1965-03-09	
0082343843	Jenine Howe	kamilah.daugherty@hotmail.com	837 Corkery Summit	1999-07-10	
0146134920	Mrs. Dustin Hyatt	daisy.hintz@gmail.com	638 Cole Field	2006-01-21	
0159507335	Ms. Kenneth Fisher	yadira.brueen@yahoo.com	1480 Jeanice Plaza	1995-04-16	
0223468142	Belva Bins	art.frami@yahoo.com	99551 Bo Turnpike	2002-12-26	
0234171766	Claribel Wilderman V	sandra.leuschke@yahoo.com	02171 Mosciski Fords	1998-03-27	
0256515858	Percy Stanton	cecil.stamm@hotmail.com	450 Parker Islands	1988-06-30	
0259335358	Mr. Rickey Schmidt	jacques.mueller@hotmail.com	1970 Georgianne Isle	1983-04-21	
0272605439	Forest Runolfsdottir	salvatore.kihn@hotmail.com	9106 Sunni Hollow	2001-09-10	
0318351281	Charlie Lakin	arden.collier@gmail.com	514 Pollich Underpass	1968-10-15	
0346954588	Ngan West I	karine.beatty@gmail.com	527 Jefferey Throug...	1988-07-16	
0353754056	Sherman Trantow	frankie.kautzer@hotmail.com	50793 Keisha Drive	1990-08-01	
0415508666	Mrs. Jody Paucek	german.stokes@gmail.com	3997 Garnet Corner	2004-08-22	
0473725333	Bernard O'Reilly	lorette.braun@yahoo.com	18969 Haywood Plain	2002-03-09	
0490331157	Mrs. Christiana Ritc...	see.langosh@hotmail.com	77906 Franecki Mou...	1993-05-21	
0522802638	Emily Barton	tonja.rempel@hotmail.com	32774 Russel Spring	1975-05-23	
0559025163	Miss Edgardo Alten...	renaldo.feest@yahoo.com	7979 Travis Summit	1991-07-17	

## Inserción de empleados:

Se crea la siguiente función, la cual usa una lista de las cedulas registradas en el sistema para crear a los empleados usando Faker para generar el salario aleatoriamente.

```

public static void insertIntoEmpleado() throws SQLException {
    String sql = "INSERT INTO empleado (idEmpleado, salario) " +
        "VALUES (?, ?)";
    PreparedStatement stmt = conn.prepareStatement(sql);

    for (String cedula : cedulasPersonas.subList(0,5)) {
        String salarioEmpleado = faker.number().digits(count: 6);
        stmt.setString(parameterIndex: 1, cedula);
        stmt.setString(parameterIndex: 2, salarioEmpleado);
        stmt.executeUpdate();
    }
}

```

Obteniendo los siguientes resultados:



	idEmpleado	Salario
	2385511443	147261
	2825435672	255411
	3028886433	110237
	3162116658	247414
	3653378655	83571
	3670297285	414548
	3867403871	27402
	4314241448	684505
	4330956704	821527
	5080058637	764326
	5085384207	981803
	5977265609	28638
	6175850600	70891
	6374341133	424881
	7042670441	857161
	7065260856	976946
	7544708413	326487
	8183727538	546635

## Inserción de teléfonos

Se crean teléfonos aleatoriamente utilizando Faker y se usan las cédulas registradas en el sistema para hacer referencia a las personas ya registradas

```
1 usage  Cristian *
public static void insertIntoTelefono() throws SQLException {
    String sql = "INSERT INTO telefono (idPersona, Telefono) " +
        "VALUES (?, ?)";
    PreparedStatement stmt = conn.prepareStatement(sql);
    for (String cedula : cedulasPersonas) {
        String telephoneNumber = faker.phoneNumber().cellPhone();
        stmt.setString( parameterIndex: 1, cedula);
        stmt.setString( parameterIndex: 2, telephoneNumber);

        stmt.executeUpdate();
    }
}
```

Obteniendo los siguientes resultados:

	idPersona	Telefono
▶	0044708610	328-357-1275
	0067995511	(649) 540-8479
	0082343843	1-983-912-7597
	0146134920	1-704-057-0561
	0159507335	574-576-0077
	0223468142	931.805.7353
	0234171766	1-171-264-0235
	0256515858	(558) 699-8642
	0259335358	980-845-5812
	0272605439	1-638-685-9708
	0318351281	195-670-1688
	0346954588	467-761-3289
	0353754056	122-993-1878
	0415508666	(542) 745-5556
	0473725333	(509) 711-6123
	0490331157	243.015.3874
	0522802638	1-431-660-8359
	0559025163	860-834-0464

## ¿Opinión sobre los resultados obtenidos?

Los resultados obtenidos de acuerdo con el contexto son buenos ya que se logra una normalización de la base de datos y una relación entre las diferentes entidades sin la necesidad de repetir información vinculando todo con llave foráneas, este permite establecer una configuración de manera específica que permite evitar la redundancia de datos para cumplir con los objetivos de los Requerimientos establecidos en la información que se desea guardar y el control que se le debe dar a los datos, las bases de datos relacional permite realizar el manejo de la información de los negocios de manera organizada.

## Conclusión

- Se documento y evidencio todo el proceso del taller.
- Se alcanzaron los objetivos propuestos.
- Se diseño un modelo que cumple con los requisitos y necesidades del negocio plane.
- Se logro modelar la base de datos cumplimiento todos los requerimientos establecidos.
- Se realizo plan de calidad y matriz de riesgo los objetivos propuestos.