Primer taller tercer corte - Bases de datos

Presentado por

* Maverick Diaz Serna
* Owen Badel Hooker
* Mileidys Agamez Ospino
* Augusto Rafael Mallarino Montoya

Estudiantes de

Ingeniería de Sistemas

Unicolombo

Descripción Proyecto de Aula:  
  
Nuestro proyecto de aula está basado en un sistema de venta de servicios turísticos basado en un E-Commerce, buscamos hacer un sitio web accesible el cual tenga funcionalidades como, reserva de servicios, pagos en línea, facturación, estadísticas de ventas, sistema de cuentas, gestión de servicios, etc.

Diagrama entidad – relación:

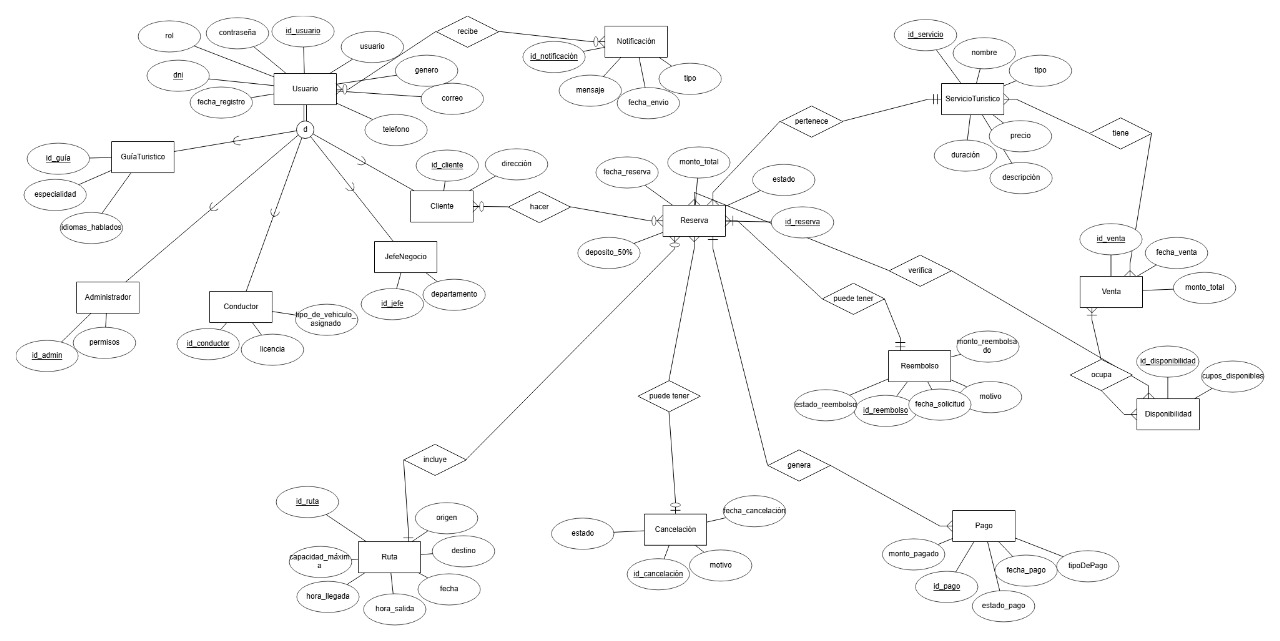
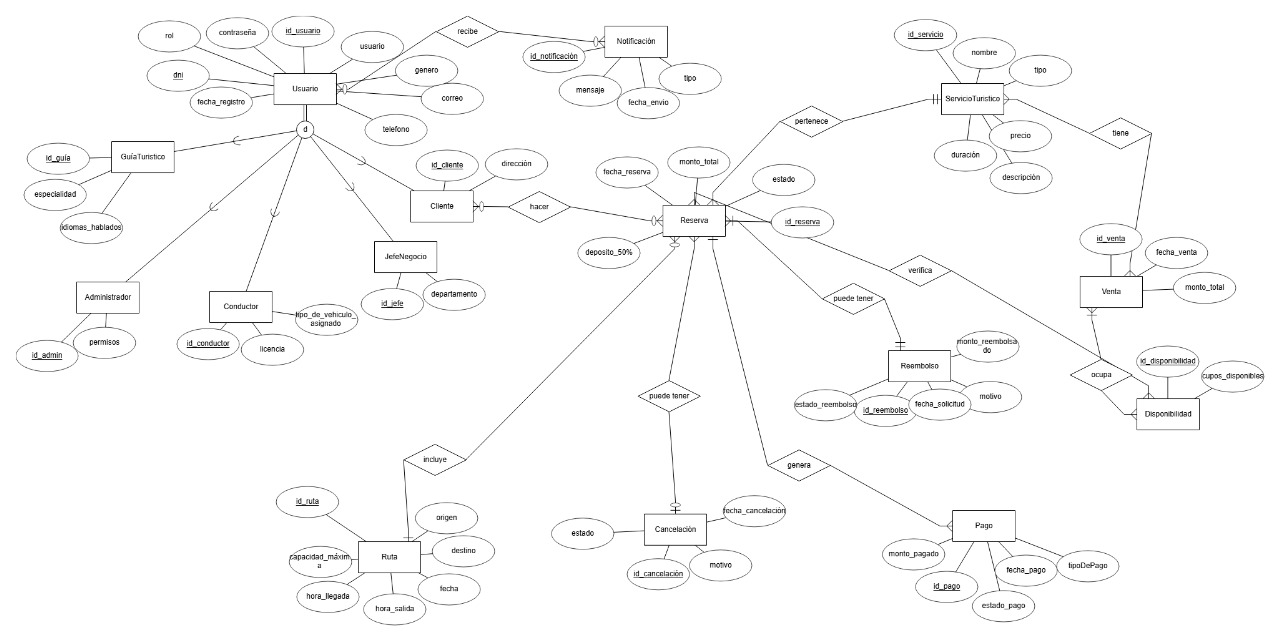


Diagrama relacional o modelo relacional:



Script de la creación de base de datos:

CREATE DATABASE Classy;

Script de la creación del usuario:

CREATE USER 'admin' IDENTIFIED BY '010203';

Script para asignarle los permisos al usuario:

GRANT ALL PRIVILEGES ON Classy .\* TO 'admin'@'%';

Scripts para la creación de las tablas:

* Tabla usuario:

CREATE TABLE Usuario (

id\_usuario INT AUTO\_INCREMENT PRIMARY KEY,

usuario VARCHAR(50) NOT NULL UNIQUE,

contraseña VARCHAR(255) NOT NULL,

rol ENUM('Cliente', 'Guía', 'Administrador', 'Conductor', 'Jefe de Negocio') NOT NULL,

dni VARCHAR(15) UNIQUE NOT NULL,

genero ENUM('Masculino', 'Femenino', 'Otro'),

correo VARCHAR(100) NOT NULL UNIQUE,

telefono VARCHAR(20),

fecha\_registro TIMESTAMP DEFAULT CURRENT\_TIMESTAMP

);

* Subtipos de usuarios:

CREATE TABLE Guia\_Turistico (

id\_guia INT PRIMARY KEY,

especialidad VARCHAR(100),

idiomas\_hablados TEXT,

FOREIGN KEY (id\_guia) REFERENCES Usuario(id\_usuario) ON DELETE CASCADE

);

CREATE TABLE Administrador (

id\_admin INT PRIMARY KEY,

permisos TEXT,

FOREIGN KEY (id\_admin) REFERENCES Usuario(id\_usuario) ON DELETE CASCADE

);

CREATE TABLE Conductor (

id\_conductor INT PRIMARY KEY,

licencia VARCHAR(50) UNIQUE NOT NULL,

tipo\_de\_vehiculo\_asignado VARCHAR(50),

FOREIGN KEY (id\_conductor) REFERENCES Usuario(id\_usuario) ON DELETE CASCADE

);

CREATE TABLE Jefe\_Negocio (

id\_jefe INT PRIMARY KEY,

departamento VARCHAR(100),

FOREIGN KEY (id\_jefe) REFERENCES Usuario(id\_usuario) ON DELETE CASCADE

);

CREATE TABLE Cliente (

id\_cliente INT PRIMARY KEY,

direccion TEXT,

FOREIGN KEY (id\_cliente) REFERENCES Usuario(id\_usuario) ON DELETE CASCADE

);

* Crear tabla notificación:

CREATE TABLE notificacion (

id\_notificacion INT AUTO\_INCREMENT PRIMARY KEY,

id\_usuario INT NOT NULL,

mensaje TEXT,

tipo VARCHAR(40),

fecha\_envio DATETIME DEFAULT CURRENT\_TIMESTAMP,

FOREIGN KEY (id\_usuario) REFERENCES usuario(id\_usuario)

ON DELETE CASCADE

);

* Crear tabla servicio\_turistico

CREATE TABLE servicio\_turistico (

id\_servicio INT AUTO\_INCREMENT PRIMARY KEY,

nombre VARCHAR(120) NOT NULL,

tipo ENUM('TOUR','TRANSPORTE','ENTRADA','OTRO') NOT NULL,

precio DECIMAL(10,2) NOT NULL,

duracion VARCHAR(60),

descripcion TEXT

);

* Crear tabla ruta

CREATE TABLE ruta (

id\_ruta INT AUTO\_INCREMENT PRIMARY KEY,

origen VARCHAR(120) NOT NULL,

destino VARCHAR(120) NOT NULL,

fecha DATE NOT NULL,

hora\_salida TIME,

hora\_llegada TIME,

capacidad\_maxima INT

);

* Crear tabla reserva:

CREATE TABLE reserva (

id\_reserva INT AUTO\_INCREMENT PRIMARY KEY,

id\_cliente INT NOT NULL,

id\_servicio INT NOT NULL,

id\_ruta INT DEFAULT NULL,

fecha\_reserva DATETIME DEFAULT CURRENT\_TIMESTAMP,

monto\_total DECIMAL(10,2) NOT NULL,

deposito\_50 DECIMAL(10,2),

estado ENUM('PENDIENTE','CONFIRMADA','CANCELADA') DEFAULT 'PENDIENTE',

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente),

FOREIGN KEY (id\_servicio) REFERENCES servicio\_turistico(id\_servicio),

FOREIGN KEY (id\_ruta) REFERENCES ruta(id\_ruta)

);

* Tabla Cancelacion:

CREATE TABLE cancelacion (

id\_cancelacion INT AUTO\_INCREMENT PRIMARY KEY,

id\_reserva INT NOT NULL UNIQUE,

fecha\_cancelacion DATETIME DEFAULT CURRENT\_TIMESTAMP,

motivo VARCHAR(200),

estado ENUM('SOLICITADA','APROBADA','RECHAZADA') DEFAULT 'SOLICITADA',

FOREIGN KEY (id\_reserva) REFERENCES reserva(id\_reserva)

ON DELETE CASCADE

);

* Tabla reembolso:

CREATE TABLE reembolso (

id\_reembolso INT AUTO\_INCREMENT PRIMARY KEY,

id\_reserva INT NOT NULL UNIQUE,

fecha\_solicitud DATETIME DEFAULT CURRENT\_TIMESTAMP,

motivo VARCHAR(200),

estado\_reembolso ENUM('PENDIENTE','APROBADO','RECHAZADO') DEFAULT 'PENDIENTE',

monto\_reembolsado DECIMAL(10,2) DEFAULT 0,

FOREIGN KEY (id\_reserva) REFERENCES reserva(id\_reserva)

ON DELETE CASCADE

);

* Tabla de pago:

CREATE TABLE pago (

id\_pago INT AUTO\_INCREMENT PRIMARY KEY,

id\_reserva INT NOT NULL,

fecha\_pago DATETIME DEFAULT CURRENT\_TIMESTAMP,

monto\_pagado DECIMAL(10,2) NOT NULL,

tipoDePago ENUM('EFECTIVO','TARJETA','TRANSFERENCIA','OTRO') NOT NULL,

estado\_pago ENUM('PENDIENTE','COMPLETADO','FALLIDO') DEFAULT 'PENDIENTE',

FOREIGN KEY (id\_reserva) REFERENCES reserva(id\_reserva)

ON DELETE CASCADE

);

* Tabla de disponibilidad:

CREATE TABLE disponibilidad (

id\_disponibilidad INT AUTO\_INCREMENT PRIMARY KEY,

id\_servicio INT NOT NULL,

fecha DATE NOT NULL,

cupos\_disponibles INT NOT NULL,

UNIQUE(id\_servicio,fecha),

FOREIGN KEY (id\_servicio) REFERENCES servicio\_turistico(id\_servicio)

ON DELETE CASCADE

);

* Crear tabla de ventas

CREATE TABLE venta (

id\_venta INT AUTO\_INCREMENT PRIMARY KEY,

id\_disponibilidad INT NOT NULL,

fecha\_venta DATETIME DEFAULT CURRENT\_TIMESTAMP,

monto\_total DECIMAL(10,2) NOT NULL,

FOREIGN KEY (id\_disponibilidad) REFERENCES disponibilidad(id\_disponibilidad)

ON DELETE CASCADE

);

CRUD para las tablas:

En este caso vamos a usar la tabla usuarios como ejemplo ya que es de las que más tiene caracteres:

Insertar:

INSERT INTO Usuario (nombre, correo, contrasena, rol, dni, fecha\_registro, genero, telefono)

VALUES ('Maverick', 'AdminMaverick@gmail.com', '1234', 'administrador', '104312842', 'NOW()', 'M', '3013670678');

Leer:  
SELECT \* FROM Usuario;

SELECT \* FROM Usuario WHERE id\_usuario = 3;

Update  
UPDATE Usuario SET nombre = 'Rafa', correo = 'adminrafa@gmail.com' WHERE id\_usuario = 3;

Eliminar  
DELETE FROM Usuario WHERE id\_usuario = 3;

A continuación cada integrante realizó unos ejemplos de las operaciones que podríamos llegar a realizar durante el uso de nuestro sistema

Insertar datos Maverick Diaz

Usuario

INSERT INTO Usuario (nombre, correo, contrasena, rol, dni, fecha\_registro, genero, telefono)

VALUES ('Maverick', 'AdminMaverick@gmail.com', '1234', 'administrador', '104312842', 'NOW()', 'M', '3013670678');

Ruta

INSERT INTO Ruta (nombre\_ruta, descripcion) VALUES ('Ruta Caribe', 'Recorrido por la costa norte.');

Servicio Turístico

INSERT INTO Servicio\_Turistico (nombre\_servicio, descripcion, precio) VALUES ('Tour Islas del Rosario', 'Recorrido en lancha por las islas', 150000);

Reserva

INSERT INTO Reserva (id\_cliente, id\_servicio, id\_ruta, fecha\_reserva, estado) VALUES (1, 1, 1, CURDATE(), 'confirmada');

Pago

INSERT INTO Pago (id\_reserva, monto, metodo\_pago, fecha\_pago) VALUES (1, 150000, 'efectivo', NOW());

Cancelación

INSERT INTO Cancelacion (id\_reserva, motivo, fecha\_cancelacion) VALUES (1, 'Condiciones climáticas adversas', NOW());

Reembolso

INSERT INTO Reembolso (id\_cancelacion, monto\_reembolsado, fecha\_reembolso) VALUES (1, 150000, NOW());

Disponibilidad

INSERT INTO Disponibilidad (id\_servicio, fecha\_disponible, cupo) VALUES (1, '2025-06-01', 10);

Venta

INSERT INTO Venta (id\_disponibilidad, cantidad, fecha\_venta) VALUES (1, 2, NOW());

Notificación

INSERT INTO Notificacion (id\_usuario, mensaje, fecha\_envio) VALUES (1, '¡Gracias por tu reserva!', NOW());

Consultar Datos Augusto Mallarino

Todos los usuarios

SELECT \* FROM Usuario;

Usuario específico

SELECT \* FROM Usuario WHERE usuario = 'Adminrafa' AND rol = 'administrador';

Todas las reservas confirmadas

SELECT \* FROM Reserva WHERE estado = 'cancelada';

Pagos usando como método de pago efectivo

SELECT \* FROM Pago WHERE metodo\_pago = 'efectivo';

Disponibilidad para una fecha específica

SELECT \* FROM Disponibilidad WHERE fecha\_disponible = '2025-06-01';

Reembolsos realizados en mayo

SELECT \* FROM Reembolso WHERE MONTH(fecha\_reembolso) = 5;

Servicios turísticos con precio mayor a un millón de pesos

SELECT \* FROM Servicio\_Turistico WHERE precio > 100000;

Actualizar datos Owen

Cambiar nombre de un usuario, en este caso administrador

UPDATE Usuario SET nombre = 'Rafael' WHERE nombre = 'Rafa' AND rol = 'ADMINISTRADOR';

Cambiar estado de las reservas pendientes

UPDATE Reserva SET estado = 'completada' WHERE estado = 'PENDIENTE';

Modificar precio de un servicio

UPDATE Servicio\_Turistico SET precio = 420000 WHERE NOMBRE = 'Bora Bora';

Actualizar método de pago

UPDATE Pago SET metodo\_pago = 'tarjeta' WHERE id\_pago = 21 AND método\_pago = 'efectivo';

Eliminar datos Mileydis

Eliminar un usuario

DELETE FROM Usuario WHERE id\_usuario = 1;

Eliminar una reserva

DELETE FROM Reserva WHERE fecha\_reserva = '2025-04-05' AND id\_reserva = 14;

Eliminar un pago

DELETE FROM Pago WHERE id\_pago = 25 AND monto = '1300000';

Eliminar un servicio turístico

DELETE FROM Servicio\_Turistico WHERE id\_servicio = 21 AND NOMBRE = 'Bora Bora';

Eliminar una notificación

DELETE FROM Notificacion WHERE id\_notificacion = 1;