

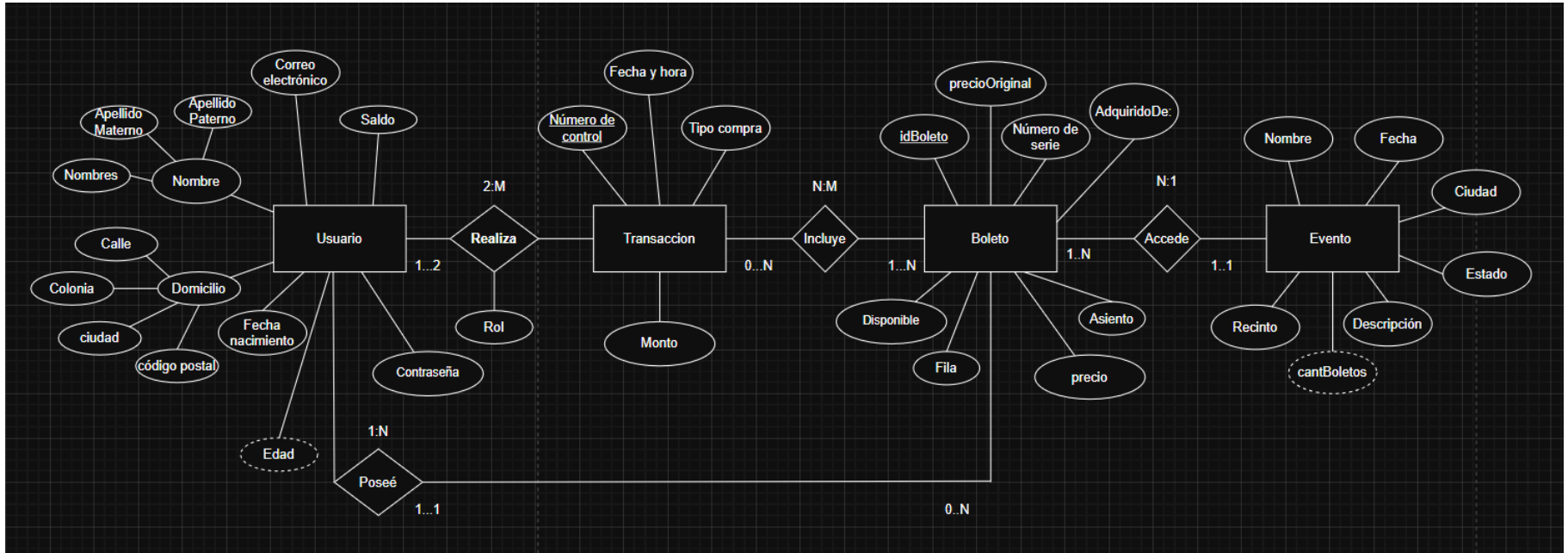
# Proyecto: TicketWizard

Enlace GitHub:

<https://github.com/IsaZzz/ProyectoBDDA>

Modelo Entidad Relación

<https://drive.google.com/file/d/1EMQLsxiZVjpcPMxEP12Ope9DPBbBL4ye/view?usp=sharing>



Modelo Relacional

[https://docs.google.com/document/d/1ajGw0G140RonVFFW\\_V1J2fGVhmq0vSYGPhmuRGmm85A/edit?tab=t.0](https://docs.google.com/document/d/1ajGw0G140RonVFFW_V1J2fGVhmq0vSYGPhmuRGmm85A/edit?tab=t.0)

## Modelo relacional

### Usuario(

id\_usuario: entero

nombre: texto(50) requerido

apellido\_paterno: texto(50) requerido

## Proyecto: TicketWizard

---

apellido\_materno: texto(50) requerido  
correo\_electronico: texto(150) requerido  
fecha\_nacimiento: fecha requerido  
saldo: decimal requerido  
**id\_direccion**: entero

)

id\_direccion pertenece a Direcciones(id\_direccion)

### Direcciones(

**id\_direccion**: entero  
ciudad: texto(100) requerido  
colonia: texto (100) requerido  
calle: texto(100) requerido  
codigo\_postal: texto(5) requerido

)

## Proyecto: TicketWizard

---

### Compras(

id\_compra: entero

fecha\_hora: fecha requerido

tipo\_compra: texto ("reventa", "boletera")

id\_usuario: entero requerido

)

id\_usuario pertenece a usuarios(id\_usuario)

### Boletos(

id\_boleto: entero

num\_serie: texto(8) requerido único

fila: texto(5) requerido

asiento: entero requerido

id\_evento: entero requerido

)

id\_evento pertenece a Eventos(id\_evento)

### Compras\_Boletos(

id\_compra\_boleto: entero

id\_compra: entero requerido

id\_boleto: entero requerido

)

id\_compra pertenece a Compras(id\_compra)

id\_boleto: pertenece a Boletos(id\_boleto)

### Eventos(

id\_evento: entero

nombre: texto (100)

## Proyecto: TicketWizard

---

fecha: fecha requerido  
estado: texto(100)  
ciudad: texto(100)  
descripción: texto(800)  
recinto: texto(100)

)

### Script BD:

```
DROP DATABASE IF EXISTS ticketWizard;  
CREATE DATABASE ticketWizard;  
use ticketWizard;
```

```
Drop Table If Exists Direcciones;  
Create Table Direcciones (  
    idDireccion Int Primary Key AUTO_INCREMENT,  
    calle Varchar (100) Not Null,  
    colonia Varchar (100) Not Null,  
    ciudad Varchar (100) Not Null,  
    codigoPostal int(5) Not Null  
);
```

```
Drop Table If Exists Usuarios;  
Create Table Usuarios (  
    idUsuario Int Primary Key AUTO_INCREMENT,  
    nombres Varchar (50) Not Null,  
    apellidoMaterno Varchar(50) Not Null,  
    apellidoPaterno Varchar(50) Not Null,  
    correoElectronico Varchar(150) Not Null,
```

## Proyecto: TicketWizard

---

```
saldo double(10,2) Not Null,  
fechaNacimiento Date Not Null,  
idDireccion int,  
Foreign Key (idDireccion) References Direcciones(idDireccion)  
);
```

```
Drop Table If Exists Compras;  
Create Table Compras (  
    idUsuario Int Not Null,  
    idCompra Int Primary Key AUTO_INCREMENT,  
    fechaHoraCompra DateTime Not Null,  
    tipoCompra Enum ("Boletera", "Reventa") Not Null,  
    Foreign Key (idUsuario) References Usuarios(idUsuario)  
);
```

```
Drop Table If Exists Eventos;  
Create Table Eventos (  
    idEvento Int Primary Key AUTO_INCREMENT,  
    nombre Varchar (100) Not Null,  
    fecha Date Not Null,  
    ciudad Varchar (50) Not Null,  
    Estado Varchar (50) Not Null,  
    descripcion Varchar (800),  
    recinto Varchar (100) Not Null  
);
```

```
Drop Table If Exists Boletos;  
Create Table Boletos (  

```

## Proyecto: TicketWizard

---

```
        idBoleto Int Primary Key AUTO_INCREMENT,  
        idEvento Int Not Null,  
        numeroSerie Varchar (8) Not Null,  
        fila Varchar(5) Not Null,  
        asiento Int Not Null,  
        Foreign Key (idEvento) References Eventos(idEvento)  
    );
```

```
Drop Table If Exists ComprasBoletos;  
Create Table ComprasBoletos (  
        idCompraBoletos Int Primary Key AUTO_INCREMENT,  
        idCompra Int Not Null,  
        idBoleto Int Not Null,  
        Foreign Key (idCompra) References Compras(idCompra),  
        Foreign Key (idBoleto) References Boletos(idBoleto)  
    );
```

```
Script BD + Inserts + SP:  
DROP DATABASE IF EXISTS ticketWizard;  
CREATE DATABASE ticketWizard;  
use ticketWizard;
```

```
Drop Table If Exists Direcciones;  
Create Table Direcciones (  
        idDireccion Int Primary Key AUTO_INCREMENT,  
        calle Varchar (100) Not Null,  
        colonia Varchar (100) Not Null,  
        ciudad Varchar (100) Not Null,  
        estado Varchar(50) Not Null,  
        codigoPostal varchar(5) Not Null
```

## Proyecto: TicketWizard

---

);

Drop Table If Exists Usuarios;

Create Table Usuarios (

idUsuario Int Primary Key AUTO\_INCREMENT,

nombres Varchar (50) Not Null,

apellidoMaterno Varchar(50) Not Null,

apellidoPaterno Varchar(50) Not Null,

correoElectronico Varchar(150) Not Null,

contraseña Varchar(255) Not Null,

fechaNacimiento Date Not Null,

saldo double(10,2) Not Null,

idDireccion int,

Foreign Key (idDireccion) References Direcciones(idDireccion)

);

Drop Table If Exists Transacciones;

Create Table Transacciones (

idTransaccion Int Primary Key AUTO\_INCREMENT,

fechaHora DateTime Not Null,

TipoCompra Enum ("Boletera", "Reventa") Not Null,

monto double(10,2) Not Null

);

Drop Table If Exists Usuarios\_Transacciones;

Create Table Usuarios\_Transacciones (

idUsuarioTransaccion Int Primary Key AUTO\_INCREMENT,

rol Enum("Comprador", "Vendedor") Not Null,

idUsuario Int Not Null,

idTransaccion Int Not Null,

Foreign Key (idUsuario) References Usuarios(idUsuario),

Foreign Key (idTransaccion) References Transacciones(idTransaccion)

);

Drop Table If Exists Eventos;

## Proyecto: TicketWizard

---

```
Create Table Eventos (  
    idEvento Int Primary Key AUTO_INCREMENT,  
    nombre Varchar (100) Not Null,  
    fecha Date Not Null,  
    ciudad Varchar (50) Not Null,  
    Estado Varchar (50) Not Null,  
    descripcion Varchar (800),  
    recinto Varchar (100) Not Null  
);
```

Drop Table If Exists Boletos;

```
Create Table Boletos (  
    idBoleto VARCHAR(15) Primary Key,  
    idEvento Int Not Null,  
    numSerie Varchar (8) Null Unique,  
    fila Varchar(5) Not Null,  
    asiento Int Not Null,  
    Disponible Bool Default 0,  
    precio decimal(10,2),  
    precioOriginal decimal(10,2),  
    idUsuario int,  
    fechaLimiteVenta date null,  
    foreign key (idUsuario) references Usuarios(idUsuario),  
    Foreign Key (idEvento) References Eventos(idEvento)  
);
```

Drop Table If Exists TransaccionesBoletos;

```
Create Table TransaccionesBoletos (  
    idTransaccionBoletos Int Primary Key AUTO_INCREMENT,  
    idTransaccion Int Not Null,  
    idBoleto varchar(15) Not Null,  
    Foreign Key (idTransaccion) References Transacciones(idTransaccion),  
    Foreign Key (idBoleto) References Boletos(idBoleto)  
);
```



## Proyecto: TicketWizard

---

Drop Table If Exists Reservas;

Create table Reservas(

idUsuario Int not null,

disponibleHasta TimeStamp Default (Current\_TimeStamp + Interval 10 Minute),

idBoleto varchar(15) primary key,

Foreign Key (idBoleto) References Boletos(idBoleto)

);

INSERT INTO Eventos (nombre, fecha, ciudad, estado, descripcion, recinto)

VALUES

('Concierto de Rock', '2025-03-15', 'Ciudad de México', 'Ciudad de México', 'Un emocionante concierto de rock con bandas nacionales e internacionales', 'Foro Sol'),

('Festival de Música Electrónica', '2025-04-10', 'Guadalajara', 'Jalisco', 'Una noche vibrante con los mejores DJs del mundo', 'Arena VFG'),

('Concierto de Música Pop', '2025-05-22', 'Morelia', 'Michoacán', 'Actuaciones en vivo de artistas pop nacionales e internacionales', 'Cinepolis Centro'),

('Festival de Jazz', '2025-06-05', 'Monterrey', 'Nuevo León', 'Una celebración del jazz con músicos talentosos de todo el mundo', 'Parque Fundidora'),

('Concierto de Música Regional', '2025-07-18', 'Oaxaca', 'Oaxaca', 'Una velada con los mejores exponentes de la música regional mexicana', 'Centro de Convenciones de Oaxaca');

-- Concierto de Rock

-- Concierto de Rock

INSERT INTO Boletos (idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, fechaLimiteVenta)

VALUES

('A100000', 1, NULL, 'A', 1, true, 1000.00, 1000.00, '2025-03-14'),

('A100001', 1, NULL, 'A', 2, true, 1000.00, 1000.00, '2025-03-14'),

('A100002', 1, NULL, 'A', 3, true, 1000.00, 1000.00, '2025-03-14'),

('A100003', 1, NULL, 'A', 4, true, 1000.00, 1000.00, '2025-03-14'),

('A100004', 1, NULL, 'A', 5, true, 1000.00, 1000.00, '2025-03-14'),

('A100005', 1, NULL, 'A', 6, true, 1000.00, 1000.00, '2025-03-14'),

('A100006', 1, NULL, 'A', 7, true, 1000.00, 1000.00, '2025-03-14'),

('A100007', 1, NULL, 'A', 8, true, 1000.00, 1000.00, '2025-03-14'),

('A100008', 1, NULL, 'A', 9, true, 1000.00, 1000.00, '2025-03-14'),

('A100009', 1, NULL, 'A', 10, true, 1000.00, 1000.00, '2025-03-14');

## Proyecto: TicketWizard

---

-- Festival de Música Electrónica

INSERT INTO Boletos (idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, fechaLimiteVenta)

VALUES

('B150000', 2, NULL, 'B', 1, true, 1500.00, 1500.00, '2025-04-09'),  
('B150001', 2, NULL, 'B', 2, true, 1500.00, 1500.00, '2025-04-09'),  
('B150002', 2, NULL, 'B', 3, true, 1500.00, 1500.00, '2025-04-09'),  
('B150003', 2, NULL, 'B', 4, true, 1500.00, 1500.00, '2025-04-09'),  
('B150004', 2, NULL, 'B', 5, true, 1500.00, 1500.00, '2025-04-09'),  
('B150005', 2, NULL, 'B', 6, true, 1500.00, 1500.00, '2025-04-09'),  
('B150006', 2, NULL, 'B', 7, true, 1500.00, 1500.00, '2025-04-09'),  
('B150007', 2, NULL, 'B', 8, true, 1500.00, 1500.00, '2025-04-09'),  
('B150008', 2, NULL, 'B', 9, true, 1500.00, 1500.00, '2025-04-09'),  
('B150009', 2, NULL, 'B', 10, true, 1500.00, 1500.00, '2025-04-09');

-- Concierto de Música Pop

INSERT INTO Boletos (idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, fechaLimiteVenta)

VALUES

('C120000', 3, NULL, 'C', 1, true, 1200.00, 1200.00, '2025-05-21'),  
('C120001', 3, NULL, 'C', 2, true, 1200.00, 1200.00, '2025-05-21'),  
('C120002', 3, NULL, 'C', 3, true, 1200.00, 1200.00, '2025-05-21'),  
('C120003', 3, NULL, 'C', 4, true, 1200.00, 1200.00, '2025-05-21'),  
('C120004', 3, NULL, 'C', 5, true, 1200.00, 1200.00, '2025-05-21'),  
('C120005', 3, NULL, 'C', 6, true, 1200.00, 1200.00, '2025-05-21'),  
('C120006', 3, NULL, 'C', 7, true, 1200.00, 1200.00, '2025-05-21'),  
('C120007', 3, NULL, 'C', 8, true, 1200.00, 1200.00, '2025-05-21'),  
('C120008', 3, NULL, 'C', 9, true, 1200.00, 1200.00, '2025-05-21'),  
('C120009', 3, NULL, 'C', 10, true, 1200.00, 1200.00, '2025-05-21');

-- Festival de Jazz

INSERT INTO Boletos (idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, fechaLimiteVenta)

VALUES

('D180000', 4, NULL, 'D', 1, true, 1800.00, 1800.00, '2025-06-04'),  
('D180001', 4, NULL, 'D', 2, true, 1800.00, 1800.00, '2025-06-04'),  
('D180002', 4, NULL, 'D', 3, true, 1800.00, 1800.00, '2025-06-04'),  
('D180003', 4, NULL, 'D', 4, true, 1800.00, 1800.00, '2025-06-04');

## Proyecto: TicketWizard

---

```
('D180004', 4, NULL, 'D', 5, true, 1800.00, 1800.00, '2025-06-04'),
('D180005', 4, NULL, 'D', 6, true, 1800.00, 1800.00, '2025-06-04'),
('D180006', 4, NULL, 'D', 7, true, 1800.00, 1800.00, '2025-06-04'),
('D180007', 4, NULL, 'D', 8, true, 1800.00, 1800.00, '2025-06-04'),
('D180008', 4, NULL, 'D', 9, true, 1800.00, 1800.00, '2025-06-04'),
('D180009', 4, NULL, 'D', 10, true, 1800.00, 1800.00, '2025-06-04');
```

-- Musica Regional

```
INSERT INTO Boletos (idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, fechaLimiteVenta)
VALUES
```

```
('F50000', 4, NULL, 'D', 1, true, 500.00, 500.00, '2025-07-17'),
('F50001', 4, NULL, 'D', 2, true, 500.00, 500.00, '2025-07-17'),
('F50002', 4, NULL, 'D', 3, true, 500.00, 500.00, '2025-07-17'),
('F50003', 4, NULL, 'D', 4, true, 500.00, 500.00, '2025-07-17'),
('F50004', 4, NULL, 'D', 5, true, 500.00, 500.00, '2025-07-17'),
('F50005', 4, NULL, 'D', 6, true, 500.00, 500.00, '2025-07-17'),
('F50006', 4, NULL, 'D', 7, true, 500.00, 500.00, '2025-07-17'),
('F50007', 4, NULL, 'D', 8, true, 500.00, 500.00, '2025-07-17'),
('F50008', 4, NULL, 'D', 9, true, 500.00, 500.00, '2025-07-17'),
('F50009', 4, NULL, 'D', 10, true, 500.00, 500.00, '2025-07-17');
```

```
INSERT INTO Direcciones (calle, colonia, ciudad, estado, codigoPostal)
```

```
VALUES
```

```
('Av. Insurgentes Sur', 'Roma Norte', 'Ciudad de México', 'CDMX', '06700'),
('Blvd. Díaz Ordaz', 'Valle Oriente', 'San Pedro Garza García', 'Nuevo León', '66220'),
('Calle 60', 'Centro', 'Mérida', 'Yucatán', '97000'),
('Av. Revolución', 'San Ángel', 'Ciudad de México', 'CDMX', '06600'),
('Paseo de la Reforma', 'Juárez', 'Ciudad de México', 'CDMX', '01000');
```

```
INSERT INTO Usuarios (nombres, apellidoMaterno, apellidoPaterno, correoElectronico, contraseña, fechaNacimiento, saldo, idDireccion)
```

```
VALUES
```

```
('Juan Carlos', 'Gómez', 'López', 'juancarlos.gomez@example.com',
'$2a$12$XvxfnMS70eRvOAPHPRvvK.niti9Y6tddTBLQaOwO/b6bdzTAOYC7q', '1990-05-15', 0.00, 1),
('Maria', 'Hernández', 'García', 'maria.hernandez@gmail.com', '$2a$12$MLAh9YlwNvYzDI8yMkg2euB9r.WxltHho/7idJGH4Ss.hrl0olvPO',
'1992-08-22', 0.00, 2),
```

## Proyecto: TicketWizard

---

```
('Luis Alberto', 'Martínez', 'Sánchez', 'luis.martinez@outlook.com',
'$2a$12$HKq3MW/jQvzJz.9M8SUTmeYcYM2xY3/ByOmLQENuERhy5rSbQPLZK', '1985-11-30', 0.00, 3),
('Ana Sofía', 'Pérez', 'Rodríguez', 'ana.perez@gmail.com', '$2a$12$0N/QtcQbSWiRH0IUsPZy3uZbRZI8ZmN6M2nbKpM/gla9ETTh67Kqm',
'1995-03-10', 0.00, 4),
('Carlos Eduardo', 'Díaz', 'Ramírez', 'carlos.diaz@gmail.com',
'$2a$12$wLgR9Yrsu0HWtzBFP.eYpuOHvSYe9eHBP9JMyPv08Z7YakqR6o.Z2', '1988-07-25', 0.00, 5);
```

-- Boleto asignados a usuarios (disponibilidad = 0)

```
INSERT INTO Boleto (idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, idUsuario, fechaLimiteVenta)
VALUES
```

-- Boleto para el Concierto de Rock (idEvento = 1)

```
('A100010', 1, SUBSTRING(MD5(RAND()), 1, 8), 'B', 1, 0, 1000.00, 1000.00, 1, NULL),
('A100011', 1, SUBSTRING(MD5(RAND()), 1, 8), 'B', 2, 0, 1000.00, 1000.00, 2, NULL),
('A100012', 1, SUBSTRING(MD5(RAND()), 1, 8), 'B', 3, 0, 1000.00, 1000.00, 3, NULL),
('A100013', 1, SUBSTRING(MD5(RAND()), 1, 8), 'B', 4, 0, 1000.00, 1000.00, 4, NULL),
('A100014', 1, SUBSTRING(MD5(RAND()), 1, 8), 'B', 5, 0, 1000.00, 1000.00, 5, NULL),
```

-- Boleto para el Festival de Música Electrónica (idEvento = 2)

```
('B150010', 2, SUBSTRING(MD5(RAND()), 1, 8), 'C', 1, 0, 1500.00, 1500.00, 1, NULL),
('B150011', 2, SUBSTRING(MD5(RAND()), 1, 8), 'C', 2, 0, 1500.00, 1500.00, 2, NULL),
('B150012', 2, SUBSTRING(MD5(RAND()), 1, 8), 'C', 3, 0, 1500.00, 1500.00, 3, NULL),
('B150013', 2, SUBSTRING(MD5(RAND()), 1, 8), 'C', 4, 0, 1500.00, 1500.00, 4, NULL),
('B150014', 2, SUBSTRING(MD5(RAND()), 1, 8), 'C', 5, 0, 1500.00, 1500.00, 5, NULL),
```

-- Boleto para el Concierto de Música Pop (idEvento = 3)

```
('C120010', 3, SUBSTRING(MD5(RAND()), 1, 8), 'D', 1, 0, 1200.00, 1200.00, 1, NULL),
('C120011', 3, SUBSTRING(MD5(RAND()), 1, 8), 'D', 2, 0, 1200.00, 1200.00, 2, NULL),
('C120012', 3, SUBSTRING(MD5(RAND()), 1, 8), 'D', 3, 0, 1200.00, 1200.00, 3, NULL),
('C120013', 3, SUBSTRING(MD5(RAND()), 1, 8), 'D', 4, 0, 1200.00, 1200.00, 4, NULL),
('C120014', 3, SUBSTRING(MD5(RAND()), 1, 8), 'D', 5, 0, 1200.00, 1200.00, 5, NULL),
```

-- Boleto para el Festival de Jazz (idEvento = 4)

```
('D180010', 4, SUBSTRING(MD5(RAND()), 1, 8), 'E', 1, 0, 1800.00, 1800.00, 1, NULL),
('D180011', 4, SUBSTRING(MD5(RAND()), 1, 8), 'E', 2, 0, 1800.00, 1800.00, 2, NULL),
('D180012', 4, SUBSTRING(MD5(RAND()), 1, 8), 'E', 3, 0, 1800.00, 1800.00, 3, NULL),
```

## Proyecto: TicketWizard

---

```
('D180013', 4, SUBSTRING(MD5(RAND()), 1, 8), 'E', 4, 0, 1800.00, 1800.00, 4, NULL),  
('D180014', 4, SUBSTRING(MD5(RAND()), 1, 8), 'E', 5, 0, 1800.00, 1800.00, 5, NULL),
```

-- Boletos para el Concierto de Música Regional (idEvento = 5)

```
('F50010', 5, SUBSTRING(MD5(RAND()), 1, 8), 'F', 1, 0, 500.00, 500.00, 1, NULL),  
('F50011', 5, SUBSTRING(MD5(RAND()), 1, 8), 'F', 2, 0, 500.00, 500.00, 2, NULL),  
('F50012', 5, SUBSTRING(MD5(RAND()), 1, 8), 'F', 3, 0, 500.00, 500.00, 3, NULL),  
('F50013', 5, SUBSTRING(MD5(RAND()), 1, 8), 'F', 4, 0, 500.00, 500.00, 4, NULL),  
('F50014', 5, SUBSTRING(MD5(RAND()), 1, 8), 'F', 5, 0, 500.00, 500.00, 5, NULL);
```

```
drop procedure if exists realizarRegistro;
```

```
DELIMITER $$
```

```
CREATE PROCEDURE realizarRegistro(  
    IN p_nombres VARCHAR(50),  
    IN p_apellidoPaterno VARCHAR(50),  
    IN p_apellidoMaterno VARCHAR(50),  
    IN p_correoElectronico VARCHAR(150),  
    IN p_contrasena VARCHAR(255),  
    IN p_fechaNacimiento DATE,  
    IN p_saldo DECIMAL(10,2),  
    IN p_calle VARCHAR(100),  
    IN p_colonia VARCHAR(100),  
    IN p_ciudad VARCHAR(50),  
    IN p_codigoPostal VARCHAR(5),  
    IN p_estado VARCHAR(50)  
)
```

```
)  
BEGIN
```

```
    DECLARE existe INT DEFAULT 0;
```

```
    DECLARE direccion_id INT;
```

```
    START TRANSACTION;
```

```
    SELECT COUNT(*) INTO existe  
    FROM Usuarios
```

## Proyecto: TicketWizard

---

```
WHERE correoElectronico = p_correoElectronico;

IF existe > 0 THEN
    ROLLBACK;
    SELECT 0 AS resultado;
ELSE
    INSERT INTO Direcciones (calle, colonia, ciudad, codigoPostal, estado)
    VALUES (p_calle, p_colonia, p_ciudad, p_codigoPostal, p_estado);

    SET direccion_id = LAST_INSERT_ID();

    INSERT INTO Usuarios (
        nombres,
        apellidoPaterno,
        apellidoMaterno,
        correoElectronico,
        contraseña,
        fechaNacimiento,
        saldo,
        idDireccion
    ) VALUES (
        p_nombres,
        p_apellidoPaterno,
        p_apellidoMaterno,
        p_correoElectronico,
        p_contrasena,
        p_fechaNacimiento,
        p_saldo,
        direccion_id
    );

    COMMIT;
    SELECT 1 AS resultado;
END IF;
END $$
```

## Proyecto: TicketWizard

---

DELIMITER ;

DROP PROCEDURE IF EXISTS revenderBoleto;

-- Reventa de boletos por parte del usuario

DELIMITER //

CREATE PROCEDURE revenderBoleto(

IN IDUsuario INT,

IN precioNuevo DECIMAL(10,2),

IN idBoletoVenta VARCHAR(20),

IN fechaLimite DATE

-- IN FECHA DATE

)

BEGIN

START TRANSACTION;

-- Validación de porcentaje

IF precioNuevo > (SELECT precioOriginal FROM Boletos B WHERE B.idBoleto = idBoletoVenta)\*1.03 THEN

ROLLBACK; -- Revertir si el porcentaje es mayor a 103% del precio original

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Porcentaje inválido'; -- Mensaje de error

ELSE

-- Actualiza el boleto y su precio

UPDATE BOLETOS

SET DISPONIBLE = TRUE, PRECIO = PrecioNuevo, fechaLimiteVenta = fechaLimite

WHERE idBoleto = idBoletoVenta;

-- Confirmar transacción

COMMIT;

END IF;

SELECT idBoleto, idEvento, numSerie, fila, asiento, Disponible, precio, precioOriginal, idUsuario, fechalimiteventa FROM boletos  
WHERE idBoleto = idBoletoVenta;

## Proyecto: TicketWizard

---

```
END //
DELIMITER ;

-- SP para consultar las consultas, divididas por usuario.
drop procedure if exists consultarTransacciones;

DELIMITER //
CREATE PROCEDURE consultarTransacciones(IN ID VARCHAR(20))
BEGIN
SELECT TR.FECHAHORA as "Fecha-hora",
TR.TIPOCOMPRA as "Tipo de compra",
TR.MONTO as "Monto",
UT.ROL as "Rol",
b.fila as "Fila",
b.asiento as "Asiento",
E.NOMBRE as "Evento",
TB.IDTRANSACCION as "ID de la transacción",
ut.idUsuario as "ID usuario"

FROM usuarios_transacciones as ut
JOIN transacciones as tr on ut.idtransaccion = tr.idtransaccion
JOIN TRANSACCIONESBOLETOS AS TB ON UT.idtransaccion = TB.idtransaccion
JOIN BOLETOS AS B ON TB.idBoleto = B.idBoleto
JOIN EVENTOS AS E ON B.IDEVENTO = E.IDEVENTO
where ID=UT.IDUSUARIO;
END //
DELIMITER ;

-- SP para consultar boletos por usuario
drop procedure if exists consultarBoletosUsuario;

DELIMITER //
CREATE PROCEDURE consultarBoletosUsuario(IN ID VARCHAR(20))
BEGIN
```



## Proyecto: TicketWizard

---

```
SELECT B.IDBOLETO as "idBoleto", E.idEvento as "idEvento", E.NOMBRE, B.FILA as "Fila", B.ASIENTO as "Asiento", E.FECHA as
"Fecha", E.CIUDAD as "Ciudad", B.precio as "precio", B.precioOriginal as "PrecioOriginal", B.numSerie as "numero de serie"
FROM BOLETOS AS B
INNER JOIN USUARIOS AS U ON B.idUsuario = U.idUsuario
INNER JOIN EVENTOS AS E ON B.idEvento = E.idEvento
WHERE ID=B.IDUSUARIO;
END //
DELIMITER ;

-- Consultar saldo
-- CONSULTA
DELIMITER //
CREATE PROCEDURE consultarSaldoUsuario(IN ID VARCHAR(20))
BEGIN
SELECT SALDO
FROM USUARIOS AS U
WHERE ID=U.IDUSUARIO;
END//
DELIMITER ;

drop procedure if exists ActualizarUsuario;
DELIMITER $$
CREATE PROCEDURE ActualizarUsuario(
    IN p_usuario_id INT,

    IN p_nombres VARCHAR(50),
    IN p_apellidoPaterno VARCHAR(50),
    IN p_apellidoMaterno VARCHAR(50),
    IN p_correo VARCHAR(150),
    IN p_contrasena VARCHAR(255),
    IN p_fechaNacimiento DATE,

    IN p_calle VARCHAR(100),
    IN p_colonia VARCHAR(100),
    IN p_ciudad VARCHAR(100),
```

## Proyecto: TicketWizard

---

```
IN p_estado VARCHAR(50),
IN p_codigoPostal VARCHAR(5)
)
BEGIN
    DECLARE v_direccion_id INT;
    DECLARE v_correo VARCHAR(100);
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
        SELECT 0 AS resultado;
    END;

    START TRANSACTION;

    SELECT correoElectronico, idDireccion
    INTO v_correo, v_direccion_id
    FROM Usuarios
    WHERE idUsuario = p_usuario_id;

    IF v_direccion_id IS NULL THEN
        ROLLBACK;
        SELECT 0 AS resultado;
    ELSE
        IF p_correo <> v_correo THEN
            IF EXISTS (SELECT 1 FROM Usuarios WHERE correoElectronico = p_correo) THEN
                ROLLBACK;
                SELECT 0 AS resultado;
            END IF;
        END IF;

        UPDATE Direcciones
        SET
            calle = p_calle,
            colonia = p_colonia,
```

## Proyecto: TicketWizard

---

```
        ciudad = p_ciudad,
        estado = p_estado,
        codigoPostal = p_codigoPostal
WHERE id = v_direccion_id;

UPDATE Usuarios
SET
    nombres = p_nombres,
    apellidoPaterno = p_apellidoPaterno,
    apellidoMaterno = p_apellidoMaterno,
    correo = p_correo,
    contraseña = p_contrasena,
    fechaNacimiento = p_fechaNacimiento
WHERE id = p_usuario_id;

COMMIT;
SELECT 1 AS resultado;
END IF;
END $$

DELIMITER ;

-- create procedure comprarBoletera()
DELIMITER //
CREATE PROCEDURE compraBoletera(
    IN idBoleto VARCHAR(20),
    IN precio DECIMAL(10,2),
    IN serie VARCHAR(10),
    IN IDU INT(20)
)
BEGIN
    -- Iniciar la transacción
    START TRANSACTION;

    -- Actualizar el estado del boleto
```

## Proyecto: TicketWizard

---

```
UPDATE BOLETOS
SET DISPONIBLE = FALSE, numSerie = serie
WHERE idBoleto = idBoleto;
```

```
-- Actualizar el saldo del usuario
UPDATE USUARIOS
SET SALDO = SALDO - precio
WHERE IDU = IDU;
```

```
-- Insertar una nueva transacción
INSERT INTO TRANSACCIONES (FECHAHORA, TIPOCOMPRA, MONTO)
VALUES (NOW(), 'Boletera', precio);
```

```
-- Obtener el ID de la última transacción
SET @IDTRANSACCION = LAST_INSERT_ID();
```

```
-- Insertar en la tabla de relaciones usuario-transacción
INSERT INTO USUARIOS_TRANSACCIONES (ROL, IDUSUARIO, IDTRANSACCION)
VALUES ('Comprador', IDU, @IDTRANSACCION);
```

```
-- Insertar en la tabla de transacciones de boletos
INSERT INTO TRANSACCIONESBOLETOS (IDTRANSACCION, IDBOLETO)
VALUES (@IDTRANSACCION, idBoleto);
```

```
-- Confirmar la transacción
COMMIT;
IF (SELECT saldo FROM usuarios where idUsuario = IDU) < 0 THEN
    ROLLBACK;
ELSE
    COMMIT;
END IF;
```

```
END //
DELIMITER ;
```

## Proyecto: TicketWizard

---

```
-- create procedure ventaBoletera()
drop procedure if exists comprarReventa;

DELIMITER //
create procedure comprarReventa(
    IN idBoleto VARCHAR(20),
    IN precio DECIMAL(10,2),
    IN serie VARCHAR(10),
    IN IDUC INT,
    IN IDUV INT
)
BEGIN
    DECLARE CONTINUE HANDLER FOR SQLEXCEPTION
    BEGIN
        -- Aquí puedes manejar el error, como hacer un ROLLBACK y un mensaje de error
        ROLLBACK;
        SELECT 'Error en la transacción. Los cambios han sido revertidos.' AS error_message;
    END;
    start transaction;

    UPDATE USUARIOS
    SET SALDO = SALDO - precio
    WHERE IDUC = IDUC;

    UPDATE USUARIOS
    SET SALDO = SALDO + precio
    WHERE IDUV = IDUV;

    -- Insertar una nueva transacción
    INSERT INTO TRANSACCIONES (FECHAHORA, TIPOCOMPRA, MONTO)
    VALUES (NOW(), 'Reventa', precio);

    -- Obtener el ID de la última transacción
    SET @IDTRANSACCION = LAST_INSERT_ID();
```

```
-- Insertar en la tabla de relaciones usuario-transacción
INSERT INTO USUARIOS_TRANSACCIONES (ROL, IDUSUARIO, IDTRANSACCION)
VALUES ('Comprador', IDU, @IDTRANSACCION);

-- Insertar en la tabla de transacciones de boletos
INSERT INTO TRANSACCIONESBOLETOS (IDTRANSACCION, IDBOLETO)
VALUES (@IDTRANSACCION, idBoleto);
-- Confirmar la transacción
IF (SELECT saldo FROM usuarios where idUsuario = IDU) < 0 THEN
    ROLLBACK;
ELSE
    COMMIT;
END IF;
end //
DELIMITER ;
```