



# Proyecto Taquilla virtual

...

15 de Mayo del 2023

Juan Cabaleiro Feijóo  
Laura González Lemos  
Diego Amil González  
Diego Pérez Martínez

# Índice

1. Introducción
2. Diagrama Entidad-Asociación
3. Diagrama Relacional
4. Procedimientos
  - CompraLocalidad()
  - QuitarOferta()
  - LimitarVentas()
  - EliminarEntradaEvento()
  - LimitarUsuarios()
5. Otras modificaciones

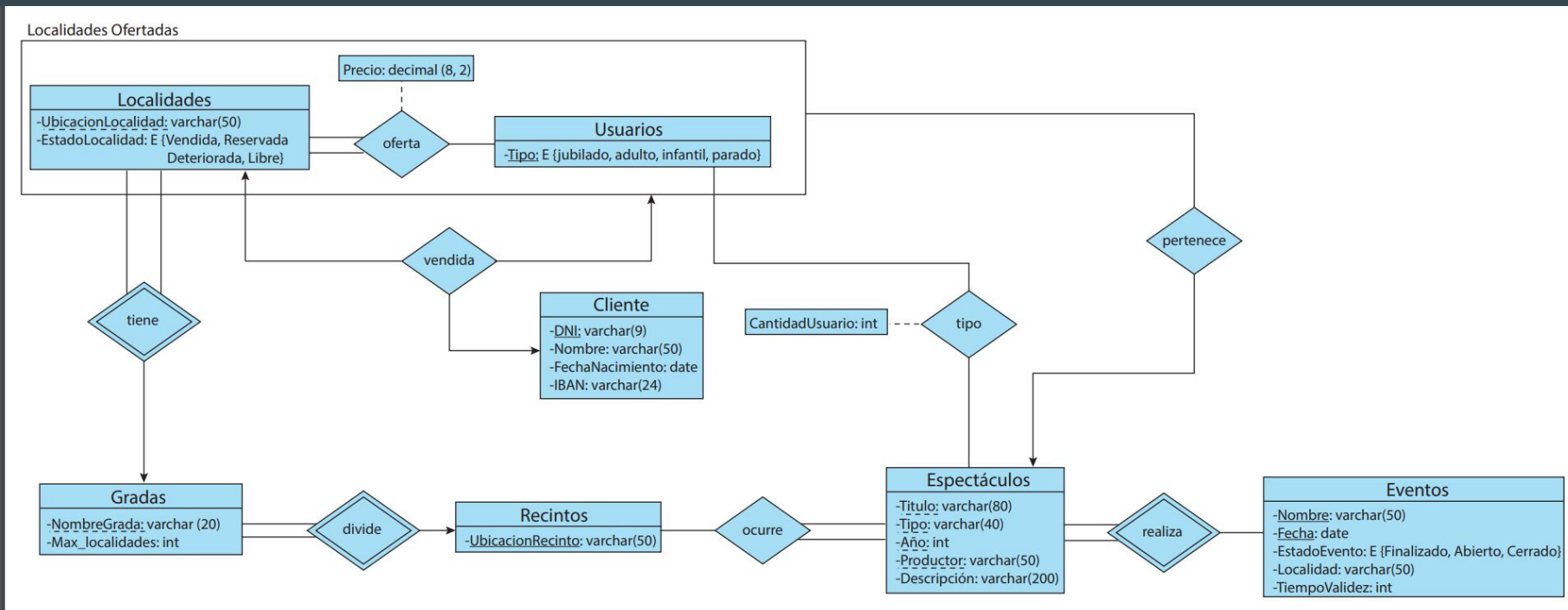
# Introducción

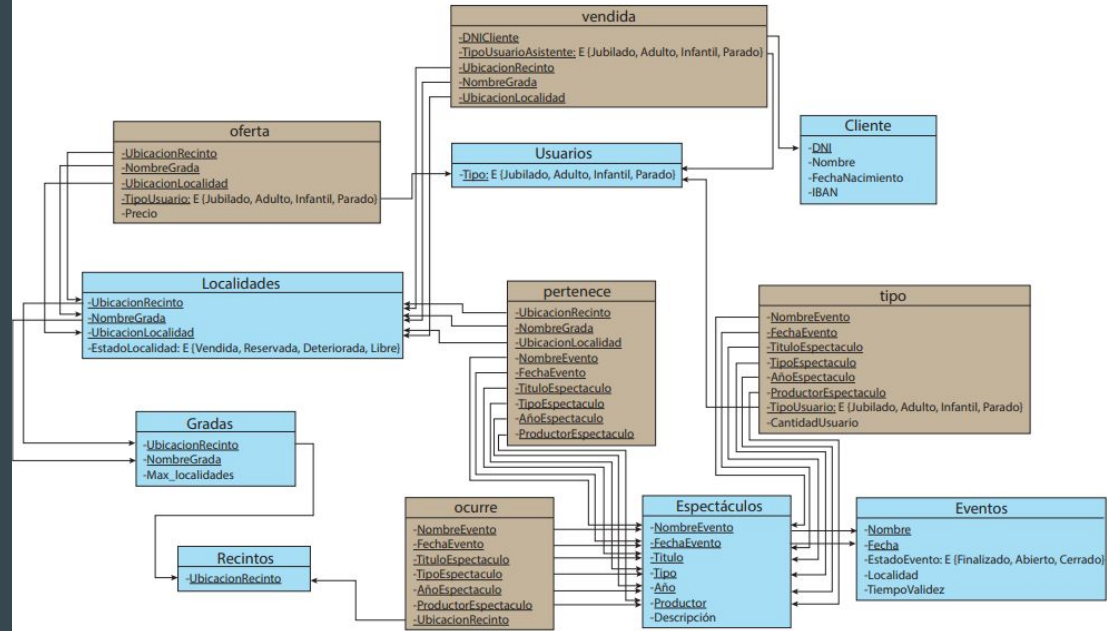
El proyecto a presentar consiste en la creación de una base de datos, que nos va a servir para desarrollar un sistema de distribución, basado en ecommerce, es decir, se ofrecerán servicios a partir de una taquilla virtual.

Dicho sistema constará tanto de una base de datos relacional como de un conjunto de procesos servidores.

En el día de hoy presentaremos cómo hemos diseñado finalmente el modelo Entidad-Asociación y el modelo Relacional. También mostraremos cómo hemos implementado estos modelos en nuestra base de datos y los distintos procedimientos que hemos creado para que se cumplan los requisitos del proyecto.

# Diagrama Entidad-Asociación





# Procedimientos

Para cumplir los requisitos que se nos pedían en este proyecto hemos creado varios procedimientos en SQL, los cuales hemos probado con miles de datos. Algunos de los requisitos que cumplen son:

- No vender dos veces la misma entrada.

- No comprar entradas para tipos de usuarios no permitidos.

- Venta al precio estipulado.

- Limitar número de entradas/usuarios para un evento.

- Que no se superen los aforos establecidos.

- Solo vender entradas de eventos abiertos.

- No vender entradas de localidades deterioradas/reservadas.

```

CREATE PROCEDURE Quitar_oferta()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE recinto varchar(50);
    DECLARE grada varchar(30);
    DECLARE localidad varchar(50);

    DECLARE Localidades_vendida CURSOR FOR
        SELECT UbicacionRecinto, NombreGrada, UbicacionLocalidad
        FROM Localidades
        WHERE EstadoLocalidad = 'Vendida' OR
            EstadoLocalidad = 'Deteriorada' OR
            EstadoLocalidad = 'Reservada';

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    OPEN Localidades_vendida;

    read_loop: LOOP
        FETCH Localidades_vendida INTO recinto, grada, localidad;

        IF done THEN
            LEAVE read_loop;
        END IF;

        IF EXISTS (
            SELECT UbicacionRecinto, NombreGrada, UbicacionLocalidad
            FROM oferta
            WHERE UbicacionRecinto = recinto AND
                NombreGrada = grada AND
                UbicacionLocalidad = localidad)
        THEN
            DELETE FROM oferta
            WHERE UbicacionRecinto = recinto AND
                NombreGrada = grada AND
                UbicacionLocalidad = localidad;
        END IF;

    END LOOP read_loop;

    CLOSE Localidades_vendida;

    COMMIT;
END;

```

# QuitarOferta()

Este procedimiento eliminará las localidades Vendidas, Deterioradas o Reservadas de las ofertas.

Primero seleccionamos las localidades no disponibles (vendida, deteriorada, reservada), después comprobamos si existen esas localidades en las ofertas, si están, las eliminamos de esa tabla.

Llamaremos a este procedimiento cuando vendamos una entrada, ya que pasará a vendida.

---

# QuitarOferta() (prueba)

Tenemos una tabla llamada oferta en la cuál tenemos mil tuplas. Para probar que este procedimiento funciona seleccionamos una Localidad de esta tabla y vemos que existe y además se oferta para varios grupos de usuarios. Vamos a hacer la prueba con 3 localidades que tienen un estado distinto de “Libre”.

```
SELECT * FROM oferta WHERE
  UbicacionRecinto = '-0.4165288, 31.3998995' AND
  NombreGrada = 'Palcos' AND
  UbicacionLocalidad = 'Fila 34, Asiento 11';

SELECT * FROM oferta WHERE
  UbicacionRecinto = '-0.45964, 100.590599' AND
  NombreGrada = 'Anfiteatro' AND
  UbicacionLocalidad = 'Fila 53, Asiento 40';

SELECT * FROM oferta WHERE
  UbicacionRecinto = '-0.789275, 113.921327' AND
  NombreGrada = 'Zona Preferente' AND
  UbicacionLocalidad = 'Fila 57, Asiento 51';
```

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-0.4165288, 31.3998995	Palcos	Fila 34, Asiento 11	Jubilado	50
-0.4165288, 31.3998995	Palcos	Fila 34, Asiento 11	Adulto	99
-0.4165288, 31.3998995	Palcos	Fila 34, Asiento 11	Infantil	145
-0.4165288, 31.3998995	Palcos	Fila 34, Asiento 11	Parado	126

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-0.45964, 100.590599	Anfiteatro	Fila 53, Asiento 40	Jubilado	103
-0.45964, 100.590599	Anfiteatro	Fila 53, Asiento 40	Adulto	44
-0.45964, 100.590599	Anfiteatro	Fila 53, Asiento 40	Infantil	118
-0.45964, 100.590599	Anfiteatro	Fila 53, Asiento 40	Parado	70

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-0.789275, 113.921327	Zona Preferente	Fila 57, Asiento 51	Jubilado	76
-0.789275, 113.921327	Zona Preferente	Fila 57, Asiento 51	Adulto	60
-0.789275, 113.921327	Zona Preferente	Fila 57, Asiento 51	Infantil	132
-0.789275, 113.921327	Zona Preferente	Fila 57, Asiento 51	Parado	99



Llamamos a nuestro procedimiento y vemos como después esa misma localidad ya no está en la tabla que contiene las localidades ofertadas.

```
CALL Quitar_oferta();

SELECT * FROM oferta WHERE
  UbicacionRecinto = '-0.4165288, 31.3998995' AND
  NombreGrada = 'Palcos' AND
  UbicacionLocalidad = 'Fila 34, Asiento 11';

SELECT * FROM oferta WHERE
  UbicacionRecinto = '-0.45964, 100.590599' AND
  NombreGrada = 'Anfiteatro' AND
  UbicacionLocalidad = 'Fila 53, Asiento 40';

SELECT * FROM oferta WHERE
  UbicacionRecinto = '-0.789275, 113.921327' AND
  NombreGrada = 'Zona Preferente' AND
  UbicacionLocalidad = 'Fila 57, Asiento 51';

SELECT * FROM oferta;
```

Resultado tras el call Quitar\_Oferta() en las localidades anteriores:

No data

Resultado tras el call Quitar\_Oferta() en el resto de localidades:

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-10.0017, 124.4251	Gran Tertulia	Fila 53, Asiento 15	Jubilado	112
-10.0017, 124.4251	Gran Tertulia	Fila 53, Asiento 15	Adulto	82
-10.0017, 124.4251	Gran Tertulia	Fila 53, Asiento 15	Infantil	125
-10.0017, 124.4251	Gran Tertulia	Fila 53, Asiento 15	Parado	61
-11.6876026, 27.5026174	Sector C	Fila 71, Asiento 95	Jubilado	85
-11.6876026, 27.5026174	Sector C	Fila 71, Asiento 95	Adulto	107
-11.6876026, 27.5026174	Sector C	Fila 71, Asiento 95	Infantil	123
-11.6876026, 27.5026174	Sector C	Fila 71, Asiento 95	Parado	78
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Jubilado	46
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Adulto	49
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Infantil	117
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Parado	103
-13.674114, -71.6777177	Gran Tertulia	Fila 40, Asiento 74	Jubilado	99
-13.674114, -71.6777177	Gran Tertulia	Fila 40, Asiento 74	Adulto	59
-13.674114, -71.6777177	Gran Tertulia	Fila 40, Asiento 74	Infantil	138
-13.674114, -71.6777177	Gran Tertulia	Fila 40, Asiento 74	Parado	108
-13.876472, -171.6425505	Galeria	Fila 92, Asiento 16	Jubilado	70
-13.876472, -171.6425505	Galeria	Fila 92, Asiento 16	Adulto	75
-13.876472, -171.6425505	Galeria	Fila 92, Asiento 16	Infantil	150

```

CREATE PROCEDURE CompraLocalidad(
    in DNICliente varchar(9),
    in NombreCliente varchar(50),
    in FechaNacimientoCliente date,
    in IBANCliente varchar(24),
    in DNIAasistente varchar(9),
    in TipoUsuarioAsistente ENUM('Jubilado', 'Adulto', 'Infantil', 'Parado'),
    in recinto varchar(50),
    in grada varchar(30),
    in localidad varchar(50)
)
BEGIN
    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    IF EXISTS (SELECT * FROM Localidades WHERE
        EstadoLocalidad = 'Libre' AND
        UbicacionRecinto = recinto AND
        NombreGrada = grada AND
        UbicacionLocalidad = localidad) THEN

        UPDATE Localidades SET EstadoLocalidad = 'Vendida' WHERE
            UbicacionRecinto = recinto AND
            NombreGrada = grada AND
            UbicacionLocalidad = localidad;

    IF NOT EXISTS (SELECT * FROM Cliente WHERE
        DNI = DNICliente AND
        Nombre = NombreCliente AND
        FechaNacimiento = FechaNacimientoCliente AND
        IBAN = IBANCliente)
    THEN
        INSERT INTO Cliente(DNI, Nombre, FechaNacimiento, IBAN) VALUES
            (DNICliente, NombreCliente, FechaNacimientoCliente, IBANCliente);
    END IF;

    CALL Quitar_oferta();

    IF NOT EXISTS (
        SELECT UbicacionRecinto, NombreGrada, UbicacionLocalidad FROM vendida WHERE
            UbicacionRecinto = recinto AND
            NombreGrada = grada AND
            UbicacionLocalidad = localidad)
    THEN
        INSERT INTO vendida (DNICliente, TipoUsuarioAsistente, UbicacionRecinto,
            NombreGrada, UbicacionLocalidad) VALUES
            (DNICliente, TipoUsuarioAsistente, recinto, grada, localidad);
    END IF;
END IF;
COMMIT;
END;

```

# CompraLocalidad()

Con este procedimiento un cliente realiza la compra de su entrada, insertando sus datos personales (DNI, nombre, IBAN...), el tipo de usuario de la persona asistente, que es para saber el precio que va a tener su entrada y los datos de la localidad (el recinto, la grada y la localidad).

El estado de esta localidad va a pasar a vendida, por tanto se insertará en vendidas y ya no podrá ser comprada por otro cliente.

Si el cliente no existe en la base de datos se creará uno en la tabla “Clientes”.

# CompraLocalidad() (prueba)

Para hacer la prueba de este procedimiento, primero mostramos que la localidad que se va a comprar existe en Localidades y tiene estado “Libre”. Esta se oferta para los distintos tipos de usuarios.

```
SELECT * FROM Localidades WHERE
UbicacionRecinto = '43.438429878398765, -5.830045432782776' AND
NombreGrada = 'Abono General' AND
UbicacionLocalidad = 'Zona Genérica';

SELECT * FROM oferta WHERE
UbicacionRecinto = '43.438429878398765, -5.830045432782776' AND
NombreGrada = 'Abono General' AND
UbicacionLocalidad = 'Zona Genérica';

SELECT * FROM Cliente WHERE
DNI = '39457218T';

SELECT * FROM venta WHERE
UbicacionRecinto = '43.438429878398765, -5.830045432782776' AND
NombreGrada = 'Abono General' AND
UbicacionLocalidad = 'Zona Genérica';
```

UbicacionRecinto	NombreGrada	UbicacionLocalidad	EstadoLocalidad
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Libre

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Jubilado	119.9
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Adulto	119.9
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Parado	119.9

También se puede ver que tanto el cliente que va a comprar la entrada no está registrado como cliente aún en nuestra base de datos, ni la localidad está registrada en la tabla “venta”.

```
SELECT * FROM Cliente WHERE
DNI = '39457218T';
```

No data

```
SELECT * FROM venta WHERE
UbicacionRecinto = '43.438429878398765, -5.830045432782776' AND
NombreGrada = 'Abono General' AND
UbicacionLocalidad = 'Zona Genérica';
```

No data

Al procedimiento le pasamos de parámetros la información del cliente, del tipo de usuario asistente y de la localidad que va a comprar.

```
CALL CompraLocalidad('39457218T', 'Francisco Molina Diaz', '1999-07-26', 'ES3456982307843910207655',  
'Infantil', '43.438429878398765', -5.830045432782776', 'Abono General', 'Zona Genérica');  
  
SELECT * FROM Localidades WHERE  
  UbicacionRecinto = '43.438429878398765, -5.830045432782776' AND  
  NombreGrada = 'Abono General' AND  
  UbicacionLocalidad = 'Zona Genérica';  
  
SELECT * FROM oferta WHERE  
  UbicacionRecinto = '43.438429878398765, -5.830045432782776' AND  
  NombreGrada = 'Abono General' AND  
  UbicacionLocalidad = 'Zona Genérica';  
  
SELECT * FROM Cliente WHERE  
  DNI = '39457218T';  
  
SELECT * FROM venta WHERE  
  DNICliente = '39457218T';
```

Como se puede ver la localidad que antes tenía de estado “libre” ahora consta como “vendida”.

UbicacionRecinto	NombreGrada	UbicacionLocalidad	EstadoLocalidad
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Vendida

Al estar la entrada vendida se elimina de la tabla oferta, ya que la localidad ya no está ofertada para ningún tipo de cliente.

No data

En la tabla Cliente ya aparecen los datos del cliente.

DNI	Nombre	FechaNacimiento	IBAN
39457218T	Francisco Molina Diaz	Mon Jul 26 1999 00:00:00 GMT+0200 (hora de verano de Europa central)	ES3456982307843910207655

Comprobamos que en la tabla vendida aparece la localidad que ha comprado nuestro cliente.

DNICliente	TipoUsuario	UbicacionRecinto	NombreGrada	UbicacionLocalidad
39457218T	Infantil	43.438429878398765, -5.830045432782776	Abono General	Zona Genérica

```

CREATE PROCEDURE Limitar_ventas()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE recinto varchar(50);
    DECLARE grada varchar(30);
    DECLARE vendidas int;
    DECLARE total_loc int;

    DECLARE Cursor_Gradas CURSOR FOR
        SELECT UbicacionRecinto, NombreGrada
        FROM Gradass;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    OPEN Cursor_Gradas;

read_loop: LOOP
    FETCH Cursor_Gradas INTO recinto, grada;

    IF done THEN
        LEAVE read_loop;
    END IF;

    SELECT COUNT(*) INTO vendidas FROM Localidades WHERE
        EstadoLocalidad = 'Vendida' AND
        NombreGrada = grada;

    IF vendidas >= (SELECT Max_Localidades FROM Gradass WHERE
        NombreGrada = grada) THEN
        DELETE FROM oferta
        WHERE UbicacionRecinto = recinto AND
        NombreGrada = grada;
    END IF;

END LOOP read_loop;

CLOSE Cursor_Gradas;

COMMIT;
END;

```

# LimitarVentas()

Este procedimiento comprueba que el número de localidades vendidas de una grada no supere el número máximo de localidades de esa grada.

Primero seleccionamos el recinto y la grada, después contamos el número de localidades vendidas de esa grada. Si ese número es mayor al máximo de localidades, eliminamos el resto de localidades ofertadas pertenecientes a esa grada.

---



# LimitarVentas() (prueba)

Para la prueba de este procedimiento primero enseñamos las localidades vendidas de una grada, de un recinto concreto y las localidades que se ofertan de esa grada en dicho recinto.

```
SELECT * FROM Localidades WHERE
    EstadoLocalidad = 'Vendida' AND
    UbicacionRecinto = '-14.06355, 32.4382307' AND
    NombreGrada = 'Terraza';

SELECT * FROM oferta WHERE
    UbicacionRecinto = '-14.06355, 32.4382307' AND
    NombreGrada = 'Terraza';
```

UbicacionRecinto	NombreGrada	UbicacionLocalidad	EstadoLocalidad
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Vendida

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Jubilado	142
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Adulto	85
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Infantil	88
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Parado	99

Llamamos a `Limitar_ventas()`. Podemos ver que los datos no han variado ya que aún no limitamos la capacidad. Posteriormente actualizamos la grada con la que estamos trabajando para que solo pueda tener una localidad y lo comprobamos.

```
CALL Limitar_ventas();

SELECT * FROM oferta WHERE
  UbicacionRecinto = '-14.06355, 32.4382307' AND
  NombreGrada = 'Terraza';

UPDATE Gradas SET Max_Localidades = 1 WHERE
  UbicacionRecinto = '-14.06355, 32.4382307' AND
  NombreGrada = 'Terraza';

SELECT * FROM Gradas WHERE
  UbicacionRecinto = '-14.06355, 32.4382307' AND
  NombreGrada = 'Terraza';
```

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Jubilado	142
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Adulto	85
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Infantil	88
-14.06355, 32.4382307	Terraza	Fila 25, Asiento 87	Parado	99

UbicacionRecinto	NombreGrada	Max_Localidades
-14.06355, 32.4382307	Terraza	1



Después volvemos a llamar al procedimiento y se observa que en ese recinto no hay ninguna localidad ofertada para esa grada en concreto, ya que la única localidad que tiene está ocupada.

```
CALL Limitar_ventas();

SELECT * FROM oferta WHERE
    UbicacionRecinto = '-14.06355, 32.4382307' AND
    NombreGrada = 'Terraza';

SELECT * FROM oferta WHERE
    NombreGrada = 'Terraza';
```

No data

Aún así la grada con el mismo nombre sigue ofertando localidades pero en otros recintos:

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Jubilado	46
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Adulto	49
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Infantil	117
-12.4, -74.7	Terraza	Fila 81, Asiento 37	Parado	103
-15.4601506, -69.4355341	Terraza	Fila 00, Asiento 99	Jubilado	76
-15.4601506, -69.4355341	Terraza	Fila 00, Asiento 99	Adulto	85
-15.4601506, -69.4355341	Terraza	Fila 00, Asiento 99	Infantil	125
-15.4601506, -69.4355341	Terraza	Fila 00, Asiento 99	Parado	58
-22.4008873, 46.127902	Terraza	Fila 48, Asiento 25	Jubilado	97
-22.4008873, 46.127902	Terraza	Fila 48, Asiento 25	Adulto	101
-22.4008873, 46.127902	Terraza	Fila 48, Asiento 25	Infantil	38
-22.4008873, 46.127902	Terraza	Fila 48, Asiento 25	Parado	120
-23.0635314, -54.2134583	Terraza	Fila 72, Asiento 64	Jubilado	132
-23.0635314, -54.2134583	Terraza	Fila 72, Asiento 64	Adulto	95

```

CREATE PROCEDURE EliminarEntradaEvento()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE recinto varchar(50);
    DECLARE grada varchar(30);
    DECLARE localidad varchar(50);
    DECLARE evento varchar(50);
    DECLARE fecha date;

    DECLARE Localidades_disponibles CURSOR FOR
        SELECT UbicacionRecinto, NombreGrada, UbicacionLocalidad, NombreEvento, FechaEvento
        FROM pertenece
        WHERE NombreEvento IN (
            SELECT Nombre FROM Eventos WHERE
                EstadoEvento = 'Finalizado' OR
                EstadoEvento = 'Cerrado'
        );

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;

    OPEN Localidades_disponibles;

read_loop: LOOP
    FETCH Localidades_disponibles INTO recinto, grada, localidad, evento, fecha;

    IF done THEN
        LEAVE read_loop;
    END IF;

    IF EXISTS (
        SELECT UbicacionRecinto, NombreGrada, UbicacionLocalidad, NombreEvento, FechaEvento
        FROM pertenece
        WHERE UbicacionRecinto = recinto AND
        NombreGrada = grada AND
        UbicacionLocalidad = localidad AND
        NombreEvento = evento AND
        FechaEvento = fecha)
    THEN
        DELETE FROM oferta
        WHERE UbicacionRecinto = recinto AND
        NombreGrada = grada AND
        UbicacionLocalidad = localidad;
    END IF;

END LOOP read_loop;

CLOSE Localidades_disponibles;

COMMIT;
END;

```

# EliminarEntradaEvento()

Este procedimiento comprueba si el evento está abierto, en caso contrario elimina las localidades ofertadas restantes de dicho evento, ya que no se pueden vender localidades de eventos cerrados o finalizados.

Primero seleccionamos las localidades que pertenecen a eventos finalizados o cerrados y posteriormente eliminamos esas localidades de las ofertadas.

# EliminarEntradaEvento() (prueba)

En este procedimiento para probar que funciona mostramos dos eventos, uno abierto y otro finalizado. Al tener estos estados no se pueden vender entradas en uno de ellos.

```
SELECT * FROM Eventos WHERE
Nombre = 'Boombastic Festival' AND
Fecha = '2023-07-21';

SELECT * FROM ocurre WHERE
NombreEvento = 'Boombastic Festival' AND
FechaEvento = '2023-07-21';

SELECT * FROM oferta WHERE
UbicacionRecinto = '43.438429878398765, -5.830045432782776';
```

Nombre	Fecha	EstadoEvento	Localidad	TiempoValidez
Boombastic Festival	Fri Jul 21 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Abierto	Asturias	15

NombreEvento	FechaEvento	TituloEspectaculo	TipoEspectaculo	AñoEspectaculo	ProductorEspectaculo	UbicacionRecinto
Boombastic Festival	Fri Jul 21 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Concierto Quevedo	Musical	2023	Boombastic	43.438429878398765, -5.830045432782776

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Jubilado	119.9
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Adulto	119.9
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Parado	119.9

```

SELECT * FROM Eventos WHERE
  Nombre = 'Carrera de Autos' AND
  Fecha = '2023-04-14';

SELECT * FROM ocurre WHERE
  NombreEvento = 'Carrera de Autos' AND
  FechaEvento = '2023-04-14';

SELECT * FROM oferta WHERE
  UbicacionRecinto = '32.654079, -5.9213829';

```

Nombre	Fecha	EstadoEvento	Localidad	TiempoValidez
Carrera de Autos	Fri Apr 14 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Finalizado	Badajoz	17

NombreEvento	FechaEvento	TituloEspectaculo	TipoEspectaculo	AñoEspectaculo	ProductorEspectaculo	UbicacionRecinto
Carrera de Autos	Fri Apr 14 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Ópera La Traviata	Competición	2023	Martin Scorsese	32.654079, -5.9213829

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
32.654079, -5.9213829	Palcos	Fila 71, Asiento 96	Jubilado	83
32.654079, -5.9213829	Palcos	Fila 71, Asiento 96	Adulto	85
32.654079, -5.9213829	Palcos	Fila 71, Asiento 96	Infantil	111
32.654079, -5.9213829	Palcos	Fila 71, Asiento 96	Parado	75

Como resultado se puede observar que ya no hay entradas para el evento “Carrera de Autos” del día 14-04-2023, puesto que su estado era cerrado. Sin embargo sí que se muestran las localidades ofertadas del evento “Boombastic Festival” del día 21-07-2023, ya que su estado era abierto.

No data

```
CALL EliminarEntradaEvento();

SELECT * FROM oferta WHERE
    UbicacionRecinto = '43.438429878398765, -5.830045432782776';

SELECT * FROM oferta WHERE
    UbicacionRecinto = '32.654079, -5.9213829';
```

UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario	Precio
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Jubilado	119.9
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Adulto	119.9
43.438429878398765, -5.830045432782776	Abono General	Zona Genérica	Parado	119.9

# LimitarUsuarios()

```
CREATE PROCEDURE Limitar_usuarios()
BEGIN
    DECLARE done INT DEFAULT FALSE;
    DECLARE evento varchar(50);
    DECLARE fecha date;
    DECLARE titulo varchar(80);
    DECLARE tipo varchar(40);
    DECLARE año int;
    DECLARE productor varchar(50);
    DECLARE usuario ENUM('Jubilado', 'Adulto', 'Infantil', 'Parado');
    DECLARE cantidad int;
    DECLARE ventas int;

    DECLARE Cursor_Espectaculos CURSOR FOR
        SELECT NombreEvento, FechaEvento, TituloEspectaculo, TipoEspectaculo,
            AñoEspectaculo, ProductorEspectaculo, TipoUsuario, CantidadUsuario
        FROM tipo;

    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    DECLARE EXIT HANDLER FOR SQLEXCEPTION
    BEGIN
        ROLLBACK;
    END;

    START TRANSACTION;
```

Este procedimiento comprueba que no se supere el máximo de usuarios de cada tipo que pueden asistir a un espectáculo.

En primer lugar seleccionamos el evento, espectáculo, tipo de usuario y la cantidad de usuarios que pueden entrar.



```

OPEN Cursor_Espectaculos;

read_loop: LOOP
    FETCH Cursor_Espectaculos INTO evento, fecha, titulo, tipo, año, productor,
        usuario, cantidad;

    IF done THEN
        LEAVE read_loop;
    END IF;

    SELECT COUNT(*) INTO ventas FROM venta v, pertenece p, Espectaculos e WHERE
        v.TipoUsuarioAsistente = usuario AND
        p.UbicacionLocalidad = v.UbicacionLocalidad AND
        p.NombreGrada = v.NombreGrada AND
        p.UbicacionRecinto = v.UbicacionRecinto AND
        e.NombreEvento = p.NombreEvento AND
        e.FechaEvento = p.FechaEvento AND
        e.Titulo = p.TituloEspectaculo AND
        e.Tipo = p.TipoEspectaculo AND
        e.Año = p.AñoEspectaculo AND
        e.Productor = p.ProductorEspectaculo AND
        e.NombreEvento = evento AND
        e.FechaEvento = fecha AND
        e.Titulo = titulo AND
        e.Tipo = tipo AND
        e.Año = año AND
        e.Productor = productor;

    IF ventas >= cantidad THEN
        DELETE FROM oferta WHERE
            TipoUsuario = usuario AND
            UbicacionRecinto IN (SELECT DISTINCT p.UbicacionRecinto FROM pertenece p WHERE
                p.NombreEvento = evento AND
                p.FechaEvento = fecha AND
                p.TituloEspectaculo = titulo AND
                p.TipoEspectaculo = tipo AND
                p.AñoEspectaculo = año AND
                p.ProductorEspectaculo = productor);
    END IF;

END LOOP read_loop;

CLOSE Cursor_Espectaculos;

COMMIT;

END;

```

Después contamos el número de localidades vendidas para ese tipo de usuario en un espectáculo concreto, y miramos que no sea superior a la cantidad permitida. Por último, eliminamos las localidades ofertadas para este tipo de usuario en dicho espectáculo..

## LimitarUsuarios() (prueba)

Para hacer la prueba de este procedimiento primero mostramos la cantidad de entradas que hay de un cierto tipo de Usuario en un determinado espectáculo.

```
SELECT COUNT(*) FROM venta v, pertenece p, Espectaculos e WHERE
    v.TipoUsuario = 'Adulto' AND
    p.UbicacionLocalidad = v.UbicacionLocalidad AND
    p.NombreGrada = v.NombreGrada AND
    p.UbicacionRecinto = v.UbicacionRecinto AND
    e.NombreEvento = p.NombreEvento AND
    e.FechaEvento = p.FechaEvento AND
    e.Titulo = p.TituloEspectaculo AND
    e.Tipo = p.TipoEspectaculo AND
    e.Año = p.AñoEspectaculo AND
    e.Productor = p.ProductorEspectaculo AND
    e.NombreEvento = 'Velada del Año 3' AND
    e.FechaEvento = '2023-07-01' AND
    e.Titulo = 'Viruzz vs Shelao' AND
    e.Tipo = 'Deportivo' AND
    e.Año = 2023 AND
    e.Productor = 'Ibai';

SELECT * FROM tipo WHERE
    NombreEvento = 'Velada del Año 3' AND
    TituloEspectaculo = 'Viruzz vs Shelao';
```

COUNT(*)
1



También mostramos los límites de asistentes de cada tipo de usuario que están dentro de la relación tipo.

NombreEvento	FechaEvento	TituloEspectaculo	TipoEspectaculo	AñoEspectaculo	ProductorEspectaculo	TipoUsuario	Cantidad
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Jubilado	68456
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Adulto	68456
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Infantil	68456
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Parado	68456

Hemos llamado al proceso `Limitar_usuarios()`, pero como el límite aún no había sido actualizado, comprobamos que los datos no han cambiado y se sigue ofertando para todos los tipos de usuario. Seguido de esto actualizamos el máximo de adultos a 1.

```
CALL Limitar_usuarios();

UPDATE tipo SET CantidadUsuario = 1 WHERE
    TipoUsuario = 'Adulto' AND
    NombreEvento = 'Velada del Año 3' AND
    TituloEspectaculo = 'Viruzz vs Shelao';

SELECT * FROM tipo WHERE
    NombreEvento = 'Velada del Año 3' AND
    TituloEspectaculo = 'Viruzz vs Shelao';
```

NombreEvento	FechaEvento	TituloEspectaculo	TipoEspectaculo	AñoEspectaculo	ProductorEspectaculo	TipoUsuario	Cantidad
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Jubilado	68456
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Adulto	1
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Infantil	68456
Velada del Año 3	Sat Jul 01 2023 00:00:00 GMT+0200 (hora de verano de Europa central)	Viruzz vs Shelao	Deportivo	2023	Ibai	Parado	68456

Aquí se puede observar que los datos no han cambiado y que las localidades siguen ofertadas para los distintos tipos de usuarios, incluyendo los del tipo “Adulto”.

```
SELECT e.Titulo, o.UbicacionRecinto, o.NombreGrada, o.UbicacionLocalidad, o.TipoUsuario, o.Precio FROM
oferta o, pertenece p, Espectaculos e WHERE
o.UbicacionRecinto = p.UbicacionRecinto AND
o.NombreGrada = p.NombreGrada AND
o.UbicacionLocalidad = p.UbicacionLocalidad AND
p.NombreEvento = e.NombreEvento AND
p.FechaEvento = e.FechaEvento AND
p.TituloEspectaculo = e.Titulo AND
p.TipoEspectaculo = e.Tipo AND
p.AñoEspectaculo = e.Año AND
p.ProductorEspectaculo = e.Productor AND
e.NombreEvento = 'Velada del Año 3' AND
e.FechaEvento = '2023-07-01' AND
e.Titulo = 'Viruzz vs Shelao' AND
e.Tipo = 'Deportivo' AND
e.Año = 2023 AND
e.Productor = 'Ibai';
```

Titulo	UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Jubilado
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Adulto
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Infantil
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Parado

Finalmente volvemos a llamar a la función, pero como habíamos actualizado el límite de adultos, podemos ver que para el espectáculo en concreto con título “Viruzz vs Shelao”, las localidades ofertadas han cambiado y ya no es posible que acudan más adultos.

```
CALL Limitar_usuarios();

SELECT e.Titulo, o.UbicacionRecinto, o.NombreGrada, o.UbicacionLocalidad, o.TipoUsuario, o.Precio FROM
oferta o, pertenece p, Espectaculos e WHERE
o.UbicacionRecinto = p.UbicacionRecinto AND
o.NombreGrada = p.NombreGrada AND
o.UbicacionLocalidad = p.UbicacionLocalidad AND
p.NombreEvento = e.NombreEvento AND
p.FechaEvento = e.FechaEvento AND
p.TituloEspectaculo = e.Titulo AND
p.TipoEspectaculo = e.Tipo AND
p.AñoEspectaculo = e.Año AND
p.ProductorEspectaculo = e.Productor AND
e.NombreEvento = 'Velada del Año 3' AND
e.FechaEvento = '2023-07-01' AND
e.Titulo = 'Viruzz vs Shelao' AND
e.Tipo = 'Deportivo' AND
e.Año = 2023 AND
e.Productor = 'Ibai';
```

Titulo	UbicacionRecinto	NombreGrada	UbicacionLocalidad	TipoUsuario
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Jubilado
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Infantil
Viruzz vs Shelao	40.43618570541075, -3.5993493869409128	Pista Zona Casters	Fila 8, Asiento 92	Parado

# Otras Modificaciones

- Al implementar las entidades, ya limitamos los tipos de usuarios a Jubilado, Adulto, Infantil y Parado, no puede haber de otro tipo.
- A la hora de implementar los Eventos hacemos lo mismo, sólo pueden estar Abiertos, Cerrados o Finalizados.

FIN