

# **TAQUILLA VIRTUAL**

# Índice

1. REQUISITOS
2. MODELO E-A
3. MODELO RELACIONAL
4. PROCEDIMIENTOS ALMACENADOS

# Requisitos

1. No se podrá vender la misma entrada a dos clientes distintos.
2. Vender al precio estipulado.
3. Saber las ventas.
4. No vender más entradas del aforo disponible.
5. No permitir que un cliente compre más entradas de las permitidas por espectáculo.
6. No vender entradas para público no permitido.
7. No anular reservas de otros.
8. No comprar localidades que no estén disponibles (deterioradas) en ese espectáculo.
9. No permitir anulaciones después de comenzado el espectáculo.

# MODELO E- A

# Conjuntos entidad identificados

## Conjunto-Entidad Localidad

- Observamos un objeto del mundo real: asiento de un estadio
- Identificado por la grada a la que pertenece, y su ubicación
- No se pueden diferenciar localidades sin saber su recinto -> entidad débil
- En un recinto hay varias localidades, una localidad solo pertenece a un recinto
- Una localidad necesita pertenecer a un recinto

## Conjunto-Entidad Recinto

- Observamos un objeto del mundo real: un estadio
- Identificado por su dirección
- Asociado con Espectáculo -> en un recinto se pueden celebrar varios espectáculos
- Un espectáculo necesita tener un recinto



## Conjunto-Entidad Espectáculo

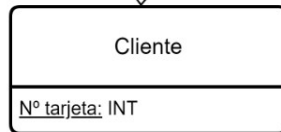
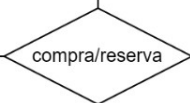
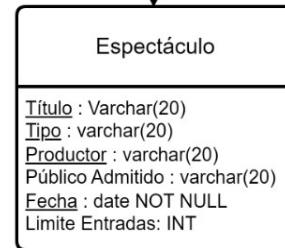
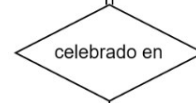
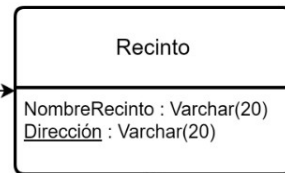
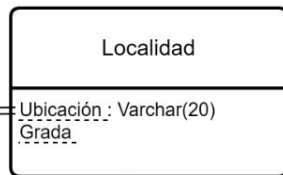
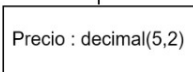
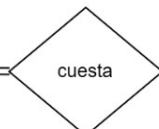
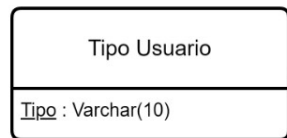
- Observamos un objeto del mundo real: un partido
- Identificado por el “título”, el tipo de espectáculo, el productor y la fecha
- Añadimos atributo público permitido para cumplir con el requisito 6
- Añadimos atributo límite entradas para cumplir con el requisito 5

## Conjunto-Entidad Cliente

- Observamos un objeto del mundo real: una persona
- Identificado por su nº de tarjeta
- Un cliente compra/reserva una o varias localidades
- Un cliente necesita comprar una localidad para existir

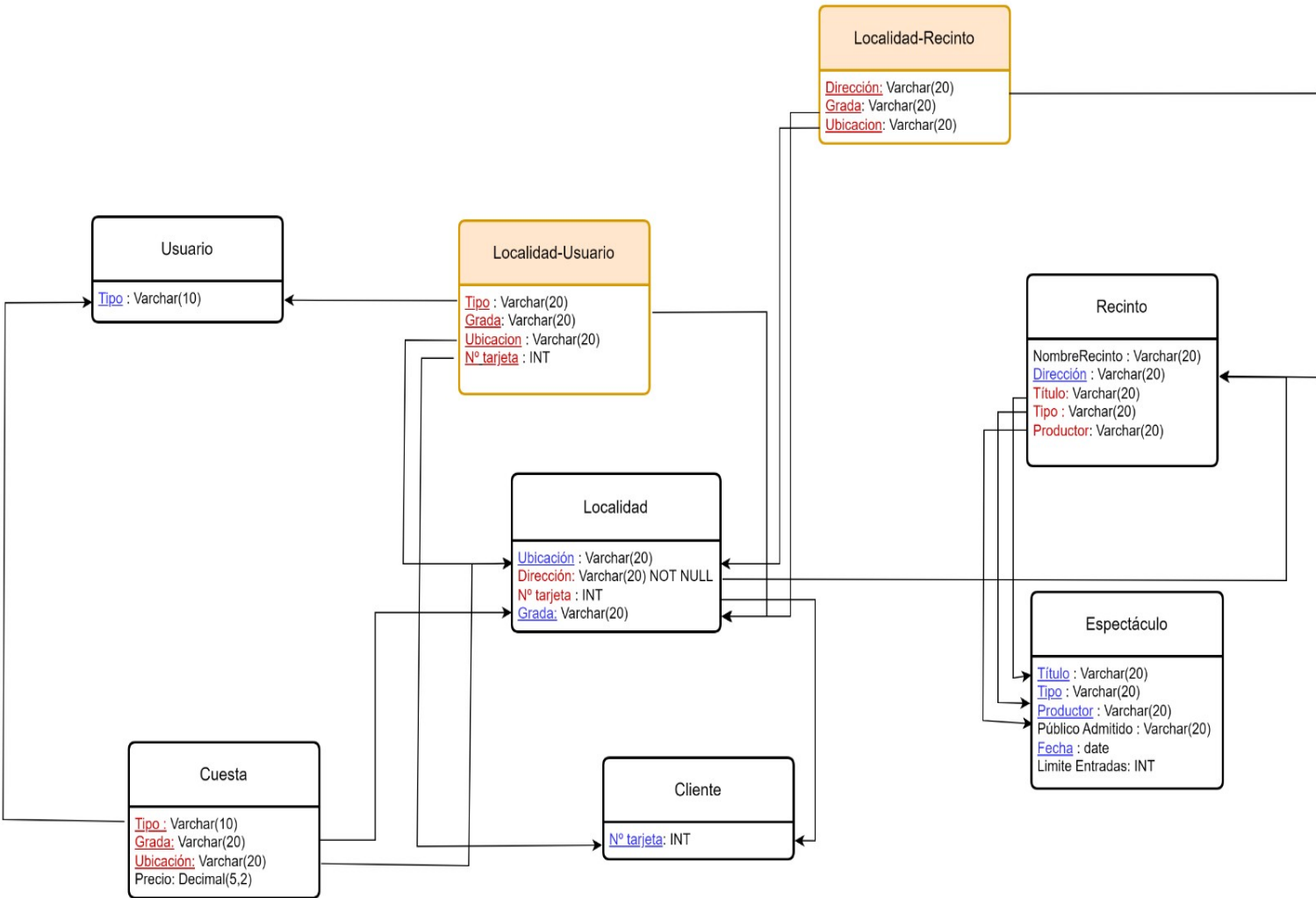
## Conjunto-Entidad Usuario

- Observamos un objeto del mundo real: personas según su rango de edad
- Un tipo de usuario tiene un valor para una localidad (dependiendo de su grada) -> atributo de asociación precio (conforma la oferta junto con la localidad)
- Para que exista un usuario, tiene que estar asociado con alguna localidad, y viceversa





# MODELO RELACIONAL



**CLAVE PRIMARIA**

**CLAVE FORÁNEA**

# Requisito 5: Comprobación del número de entradas compradas por cliente.

DELIMITER //

```
CREATE PROCEDURE verificarLimiteEntradas(  
    IN cliente_tarjeta INT, IN espectaculo_titulo VARCHAR(20), IN espectaculo_tipo VARCHAR(20),  
    IN espectaculo_fecha VARCHAR(20), IN espectaculo_productor VARCHAR(20), IN cantidad_entradas INT,  
    OUT resultado INT  
)  
BEGIN  
  
    SELECT COUNT(*) AS entradas_compradas, Localidad.numero_tarjeta  
    FROM Localidad  
    JOIN Cliente ON Localidad.cliente_tarjeta = Cliente.numero_tarjeta  
    JOIN Recinto ON Localidad.direccion = Recinto.direccion  
    JOIN Espectaculo ON Recinto.titulo = Espectaculo.titulo  
    WHERE Cliente.numero_tarjeta = cliente_tarjeta AND Espectaculo.titulo = espectaculo_titulo AND  
    Espectaculo.tipo = espectaculo_tipo AND Espectaculo.productor = espectaculo_productor  
    GROUP BY Localidad.numero_tarjeta;  
  
    IF entradas_compradas + cantidad_entradas > ( SELECT limiteEntradas FROM Espectaculo  
        WHERE titulo = espectaculo_titulo  
    ) THEN  
        SET resultado = 0;  
    ELSE  
        SET resultado = 1;  
    END IF;  
END//
```

DELIMITER ;

## Requisito 6: Comprobación tipo público

```
DELIMITER //
```

```
CREATE PROCEDURE verificarPublicoPermitido(  
    IN espectaculo_titulo VARCHAR(20), IN tipo_cliente VARCHAR(20),  
    IN espectaculo_productor VARCHAR(20), IN espectaculo_fecha DATE,  
    OUT resultado INT  
)  
BEGIN  
  
    SELECT COUNT(*) AS clientePuede, publico_admitido  
    FROM Espectaculo  
    WHERE Espectaculo.titulo = espectaculo_titulo AND Espectaculo.publico_admitido = tipo_cliente  
  
    AND Espectaculo.productor = espectaculo_productor AND Espectaculo.fecha = espectaculo_fecha  
    GROUP BY Espectaculo.publico_admitido;  
  
    IF clientePuede == 1 THEN  
        SET resultado = 1;  
    ELSE  
        SET resultado = 0;  
    END IF;  
END//  
  
DELIMITER ;
```

# Requisitos 7 y 9 : Comprobación propietario reserva e inicio de espectáculo

DELIMITER //

```
CREATE PROCEDURE verificarPropietarioReserva(  
    IN cliente_tarjeta INT, IN espectáculo_titulo VARCHAR(20), IN espectáculo_tipo VARCHAR(20),  
    IN espectáculo_fecha VARCHAR(20), IN espectáculo_productor VARCHAR(20),  
    IN direccion_localidad VARCHAR(20), IN ubicacion_localidad VARCHAR(20), IN grada_localidad VARCHAR(20),  
    OUT resultado  
)  
BEGIN  
    DECLARE fecha_actual datetime;  
    SET fecha_actual = now();  
    SELECT Cliente.numero_tarjeta, COUNT(*) AS puedeAnular  
    FROM Cliente  
    JOIN Localidad ON Cliente.numero_tarjeta = Localidad.numero_tarjeta  
    JOIN Recinto ON Localidad.direccion = Recinto.direccion  
    JOIN Espectaculo ON Recinto.titulo = Espectaculo.titulo  
    WHERE Cliente.numero_tarjeta = cliente_tarjeta AND Espectaculo.titulo = espectáculo_titulo AND  
    Espectaculo.tipo = espectáculo_tipo AND Espectaculo.fecha = espectáculo_fecha AND Espectaculo.productor =  
    espectáculo_productor AND Localidad.direccion = direccion_localidad AND Localidad.ubicacion =  
    ubicacion_localidad AND Localidad.grada = grada_localidad  
    GROUP BY Cliente.numero_tarjeta;  
  
    IF ((puedeAnular == 1) AND (fecha_actual < espectáculo_fecha)) THEN SET resultado = 1;  
    ELSE SET resultado = 0;  
    END IF;  
END//
```

DELIMITER ;



# ¿Preguntas?