

# Implementación Taquilla Virtual SQL



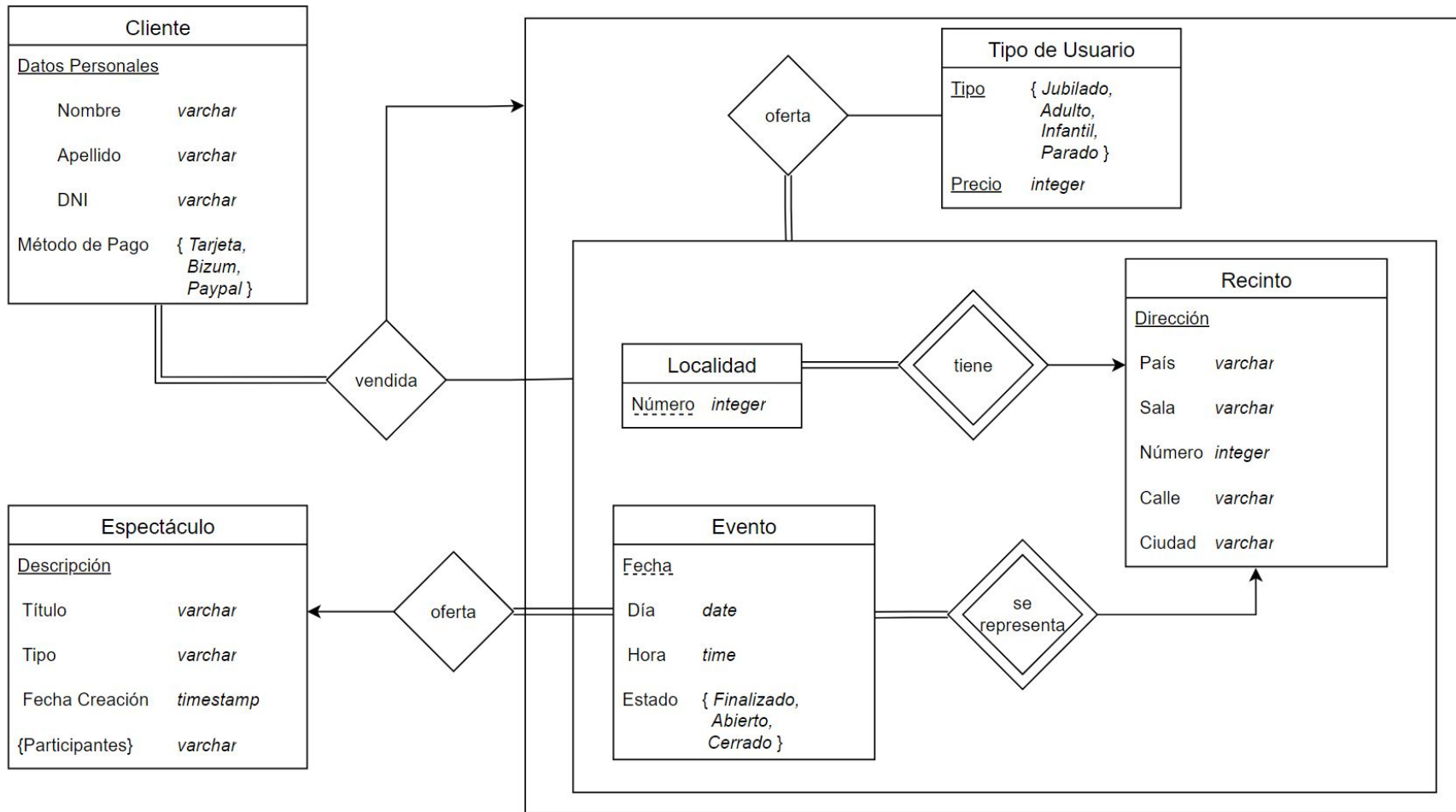
# Índice:

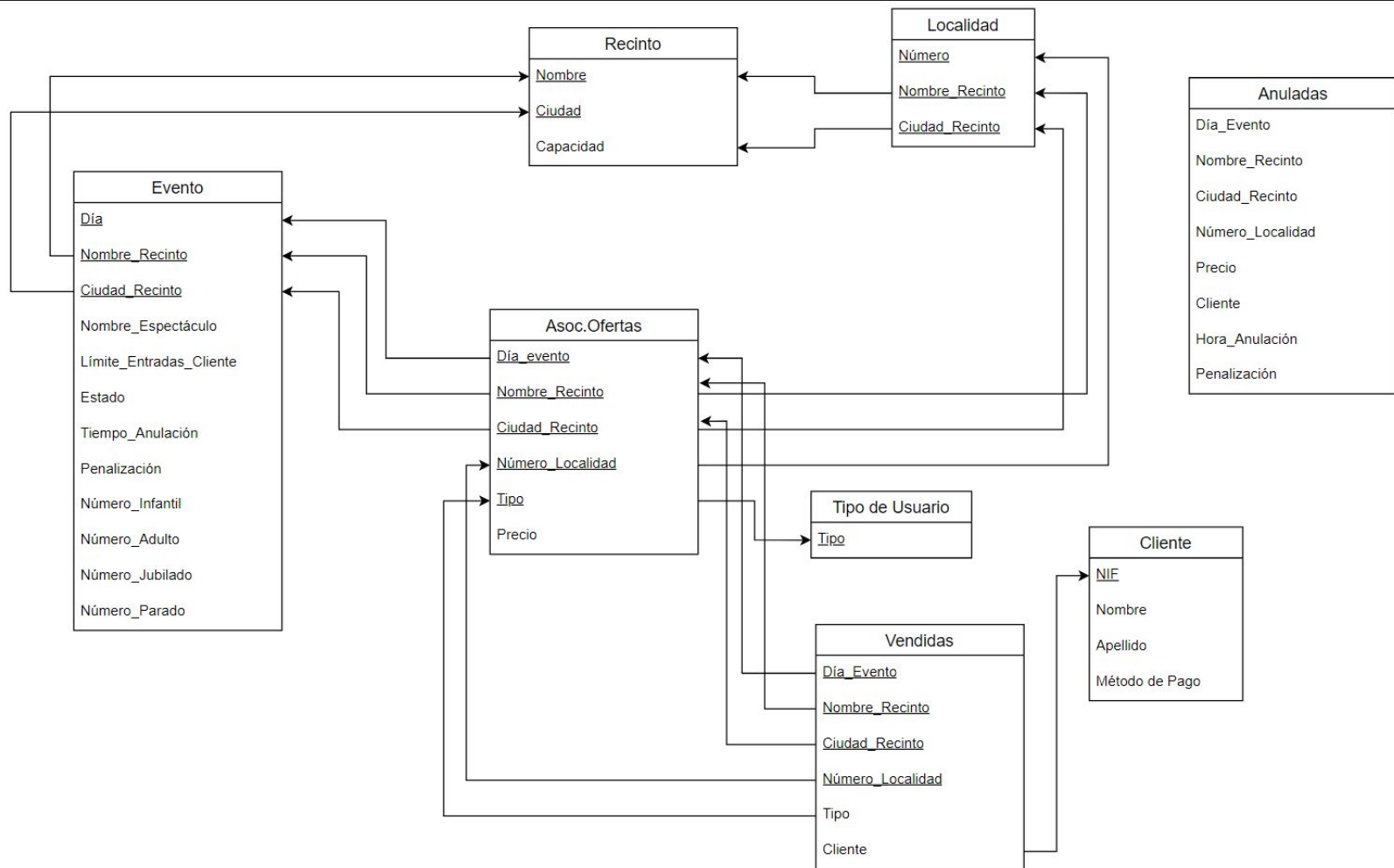
- |  |       |
|--|-------|
| 1. Esquema E-A $\rightarrow$ Esquema Relacional    | 3-5   |
| 2. Requisitos cumplidos con esquema y con triggers | 6-15  |
| 3. Presentación menú en java interactivo           | 16-21 |
| 4. Pruebas   | 22-25 |

ESQUEMA E-A



ESQUEMA RELACIONAL





# REQUISITOS

# Requisitos (I)

**[RF1]** No vender la misma entrada dos veces.

**[RF2]** No vender más entradas del aforo.

**[RF3]** Limitar número de entradas por comprador.

**[RF4]** No haya eventos en el mismo recinto al mismo tiempo.

**[RF5]** No vender entradas de eventos no operativos.

**[RF6]** Vender al precio estipulado.

**[RF7]** No anular entradas que no haya comprado.

**[RF8]** No vender entradas de un tipo de usuario no apto para el espectáculo.



## Requisitos (II)

**[RF1]** No vender la misma entrada dos veces.

Cumplido con el esquema:

No se puede insertar en la relación de Ventidas dos veces la misma entrada de un evento ya que esos atributos son PRIMARY KEY y por lo tanto no puede haber duplicados.



Vendidas
<u>Día_Evento</u>
<u>Nombre_Recinto</u>
<u>Ciudad_Recinto</u>
<u>Número_Localidad</u>
Tipo
Cliente



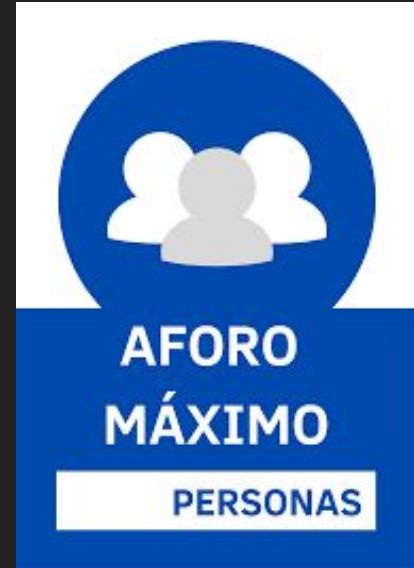
# Requisitos (III)

**[RF2]** No vender más entradas del aforo.

Cumplido mediante triggers:

No permitimos que se generen localidades con número mayor que la capacidad del recinto (Capacidad es un atributo de recinto).

Recinto
<u>Nombre</u>
<u>Ciudad</u>
Capacidad

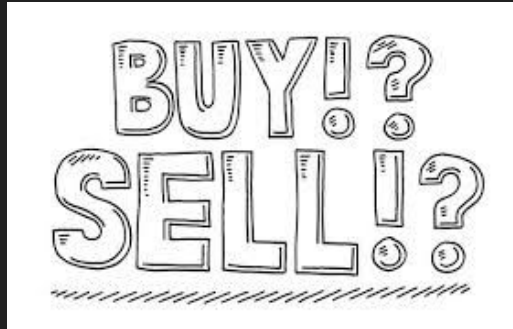


# Requisitos (IV)

**[RF3]** Limitar número de entradas por comprador.

Cumplido mediante triggers:

Usamos un atributo en Evento que limitará mediante un trigger el número de entradas que puede comprar un usuario.



Evento
<u>Día</u>
<u>Nombre_Recinto</u>
<u>Ciudad_Recinto</u>
Nombre_Espectáculo
Límite_Entradas_Cliente
Estado
Tiempo_Anulación
Penalización
Número_Infantil
Número_Adulto
Número_Jubilado
Número_Parado

# Requisitos (V)

**[RF4]** No haya eventos en el mismo recinto al mismo tiempo.

Cumplido con el esquema:

No se puede insertar en  
Eventos dos veces el mismo  
pues ambos son PRIMARY KEY y  
no puede haber duplicado.



Evento
<u>Día</u>
<u>Nombre_Recinto</u>
<u>Ciudad_Recinto</u>
Nombre_Espectáculo
Límite_Entradas_Cliente
Estado
Tiempo_Anulación
Penalización
Número_Infantil
Número_Adulto
Número_Jubilado
Número_Parado

de  
recinto  
tanto

# Requisitos (VI)

**[RF5]** No vender entradas de eventos no operativos.

Este requisito lo cumplimos mediante un trigger antes de introducir en la relación Ventidas que comprueba el estado del evento.

Evento
<u>Día</u>
<u>Nombre_Recinto</u>
<u>Ciudad_Recinto</u>
Nombre_Espectáculo
Límite_Entradas_Cliente
Estado
Tiempo_Anulación
Penalización
Número_Infantil
Número_Adulto
Número_Jubilado
Número_Parado

## Requisitos (VII)

**[RF6]** Vender al precio estipulado.

Este requisito se cumple mediante el modelo.

Ya que tenemos el atributo precio en la asociación entre la agregación y el tipo de usuario, el precio está limitado por esta asociación y el cliente no lo podrá modificar, se vende al precio estipulado por el administrador.



## Requisitos (VIII)

**[RF7]** No anular entradas que no haya comprado.

Este requisito se cumple mediante el esquema.

Ya que las entradas vendidas están en la asociación triple de ofertadas, vendidas y cliente, solo podrías anular una entrada vendida que estuviera asociada a tu NIF de cliente y nunca podrías anular una entrada que no se encontrase en la asociación vendidas.



## Requisitos (IX)

**[RF8]** No vender entradas de un tipo de usuario no apto para el espectáculo.

Este requisito se cumple mediante un trigger:

Introducimos en la relación Evento 4 atributos que nos permiten indicar un límite de entradas para cada tipo de usuario y en caso de ser 0 evitamos que se vendan entradas a tipos no aptos.



Número\_Infantil

Número\_Adulto

Número\_Jubilado

Número\_Parado

# MENÚ JAVA



# Menú Java (I)

Hemos hecho una aplicación Java que ejecuta los procedimientos para realizar diversas operaciones sobre la base de datos, dependiendo de la operación seleccionada hacemos una llamada a la base de datos que ejecuta un procedimiento con los datos aportados por el administrador/cliente.



# Menú Java (II)

- ***Introducir Recinto:*** Inserta un Recinto
- ***Introducir Recinto y generar sus localidades:*** Inserta un Recinto y automatiza la creación de sus localidades.
- ***Introducir Evento:*** Inserta un Evento
- ***Introducir Localidad:*** Inserta una localidad
- ***Introducir Entradas Ofertadas:*** Insertar ofertas
- ***Modificar Evento:*** Modificar estado del evento

## Menú Java (II)

- ***Buscar por espectáculo:*** Podremos buscar entradas ofertadas filtrando por espectáculo.
- ***Buscar por recinto:*** Se podrá buscar entradas ofertadas filtradas por recinto.
- ***Buscar por día:*** El usuario podrá buscar entradas ofertadas filtrando por la fecha.

## Menú Java (III)

- ***Buscar por día:*** El usuario podrá buscar entradas ofertadas filtrando por la fecha.
- ***Buscar por tipo de usuario:*** Filtraremos las entradas ofertadas por el tipo de usuario.
- ***Buscar localidades por evento:*** Mostraremos todas las localidades disponibles para un evento dado.

Estas funcionalidades estarán disponibles para tanto administradores como clientes.

## Menú Java (IV)

- ***Comprar una entrada:*** Esta funcionalidad exclusiva del cliente le permitirá comprar una entrada introduciendo los datos de la ofertada que desea.
- ***Consultar entradas por cliente:*** Buscaremos las entradas vendidas al cliente por su NIF
- ***Anular una entrada:*** Eliminaremos la entrada seleccionada de las ventas y la introduciremos en el historial de anuladas.
- ***Finalizar el programa.***

# PRUEBAS

# Pruebas (I)

- **Crear Recinto**
- **Intentar añadir más localidades de las que permite la capacidad del recinto**
- **Crear un Evento en el nuevo recinto**
- **Crear un Evento en el mismo recinto y el mismo día**
- **Añadir ofertas al evento creado**
- **Añadir oferta para una localidad inexistente**

# Pruebas (I)

- **Modificar un evento y ponerlo Cerrado para ver que no salen las entradas ofertadas**
- **Comprar varias entradas y comprobar tanto los triggers de limitación por tipo de usuario como la limitación de entradas compradas por un mismo cliente**
- **Intentar anular una entrada que no es nuestra**



MUCHAS GRACIAS POR SU ATENCIÓN

MARTÍN ALONSO PÉREZ

NICOLÁS ARAÚJO CALVAR

MARCOS COMESAÑA BASTERO