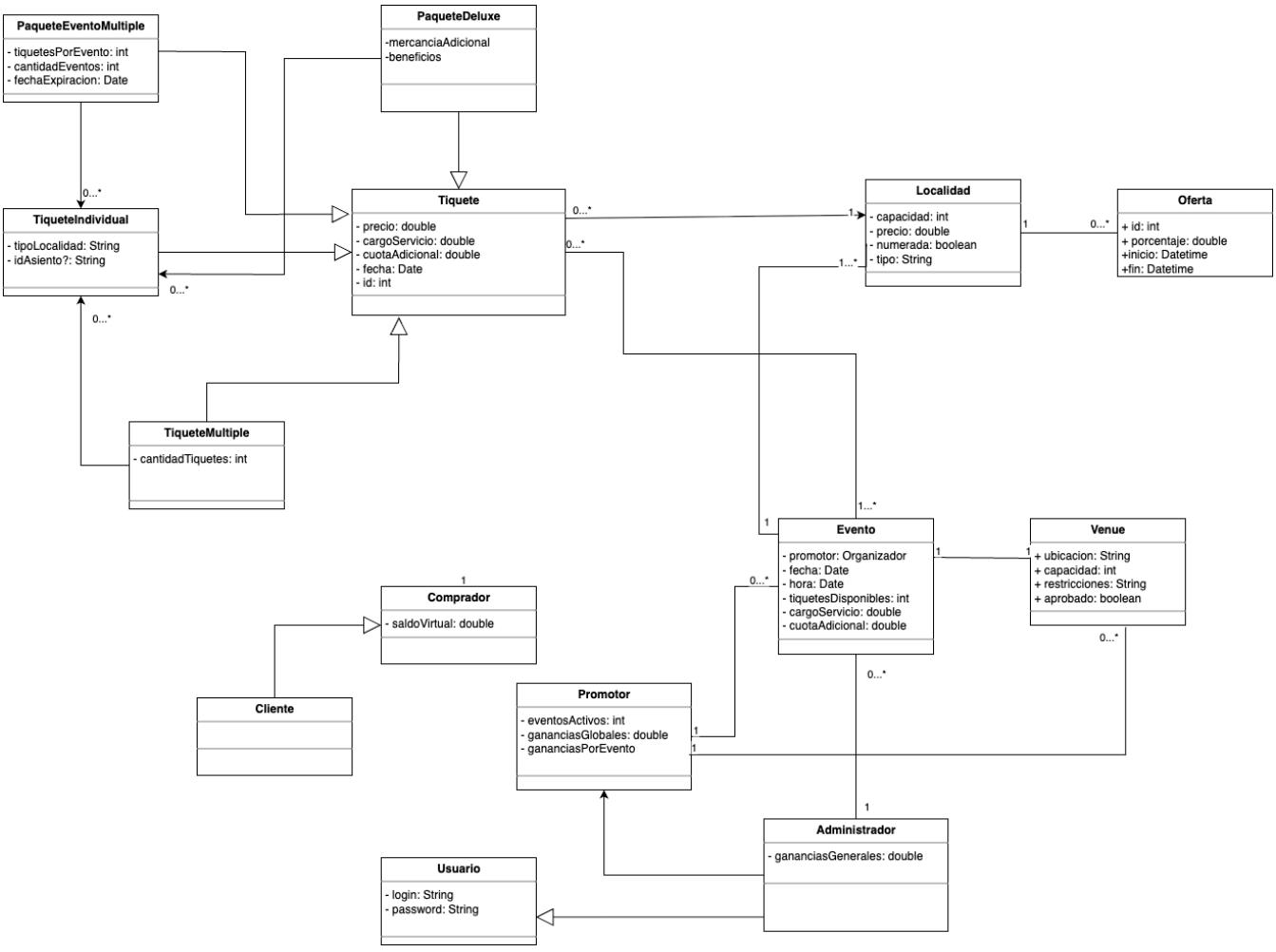
Proyecto 1 — Entrega 1 Análisis

Santiago Serrano – 202421756

Juliana Hidalgo - 202122727

José Manuel Pardo – 202411790

1. Modelo de dominio



Entidades (Clases)

- **Usuario:** login:String, password:String.

- **Comprador** ***(hereda Usuario):*** saldoVirtual:double. (Usuario que compra/transfiere tiquetes)

- **Cliente** ***(hereda Comprador)***.

- **Promotor** ***(hereda Usuario)*****:** eventosActivos:int, gananciasGlobales:double, gananciasPorEvento: Conjunto de ganancias generadas en cada evento.

- **Administrador** ***(hereda Usuario)*****:** gananciasGenerales:double.

- **Venue**: ubicacion:String, capacidad:int, restricciones:String, aprobado:boolean.

- **Evento**: promotor:Promotor, fecha:Date, hora:Date, tiquetesDisponibles:int, cargoServicio:double *(% servicio del evento)*, cuotaAdicional:double *(costo fijo por tiquete)*.

- **Localidad**: capacidad:int, precio:double *(precio base)*, numerada:boolean, tipo:String.

- **Oferta**: id:int, porcentaje:double, inicio:DateTime, fin:DateTime.

- **Tiquete**: id:int, fecha:Date, precio:double *(final)*, cargoServicio:double, cuotaAdicional:double.

- **TiqueteIndividual** ***(hereda Tiquete):*** tipoLocalidad:String, idAsiento?:int.

- **TiqueteMultiple** ***(hereda Tiquete)*****:** cantidadTiquetes:int.

- **PaqueteEventoMultiple** ***(hereda Tiquete)*****:** tiquetesPorEvento:int, cantidadEventos:int, fechaExpiracion:Date.

- **PaqueteDeluxe (hereda Tiquete):** mercanciaAdicional:String, beneficios:String.

2. Relaciones (asociaciones y herencia)

<u>Herencias</u>

Usuario: Usuario ⟵ Administrador, Comprador ⟵ Cliente, Promotor.

Tiquete: Tiquete ← TiqueteIndividual, TiqueteMultiple, PaqueteEventoMultiple, TiqueteMultiple, PaqueteDeluxe;

Asociaciones

Promotor 1 → 0...* Evento

Evento 1 → 1 Venue

Venue 1 → 0...* Evento.

Evento 1 → 1...* Localidad

Localidad 1 → 1 Evento

Localidad 1 → 0...* Oferta

Oferta 1 → 1 Localidad

Localidad 1 → 0...* Tiquete

Tiquete 1 → 1 Localidad

Usuario 1 → 0...* Tiquete

Tiquete 1 → 1 Usuario

3. Restricciones del proyecto

- Para un Venue, no puede haber dos Evento con la misma fecha.
- PrecioFinal = precio + cuotaAdicional + (precio × cargoServicio).
- Para TiqueteMultiple, el tope aplica al número de paquetes, no a la suma de boletas internas.
- Se puede transferir cualquier Tiquete salvo PaqueteDeluxe.
- Un TiqueteMultiple puede transferirse completo o parcial.
- No se puede transferir un paquete completo si alguno de sus tiquetes está vencido o ya fue transferido.
- Los Venue sugeridos por organizadores deben ser aprobados por el administrador para poder usarse.

- Todas las boletas de una misma Localidad comparten precio y características.

- Si la localidad es numerada, el asiento debe tener un identificador único dentro de la localidad.

- Oferta: descuento porcentual activo solo dentro de un rango de fechas específico.

- El administrador puede cancelar un evento a su discreción, en este caso se reembolsará el precio de los tiquetes menos el costo de emisión.

- Si un promotor solicita la cancelación de un evento y esta es autorizada por el administrador, se reembolsará solo el costo base de los tiquetes.

- Si un promotor compra tiquetes, se registran como cortesías: no generan ingresos para el promotor ni la plataforma.

- Reembolso por calamidad hacía los clientes se dará a discreción del administrador.

- El administrador no puede comprar ni transferir tiquetes.

- Los reembolsos van al saldo virtual del usuario.

- El precio de los paquetes múltiples es independiente a la suma de sus tiquetes individuales.

- Las ganancias del administrador provienen de los sobrecargos fijados por el mismo.

- Las localidades dentro de un Venue pueden variar dependiendo del evento.

- Los organizadores se encargan de crear los eventos, así como fijar fecha y hora de este.

- Pueden existir restricciones acerca del limíte de tiquetes máximos por transacción, lo cual limita la cantidad de tiquetes que un usuario comprador pueda comprar.

4. Reglas del dominio

1. **Unicidad:** Tiquete.id es único en toda la plataforma.

2. **Usuarios:** Ya sean clientes, promotores o administradores, deben contar con un login y un password.

3. **Agenda de venue:** un Venue no aloja dos Evento en la misma fecha/hora.

4. **Precio base por localidad:** el precio parte de Localidad.precio al momento de emitir el tiquete.

5. **Oferta por vigencia:** una Oferta aplica solo si inicio ≤ fechaCompra ≤ fin.

6. **Fórmula del precio final (por tiquete):** precioFinal = Localidad.precio * (1 - Oferta.porcentaje) + Tiquete.cargoServicio + Tiquete.cuotaAdicional.

7. **Numeradas con exclusividad:** el mismo asiento no se puede vender dos veces para el mismo evento/localidad.

8. **No numeradas por cupo:** TiqueteNoNumerado reduce Localidad.capacidad sin asiento.

9. **Múltiple descuenta todo:** TiqueteMultiple descuenta los cupos de todas las entradas que representa.

10. **Propiedad:** todo Tiquete pertenece a un Usuario (inicialmente el cliente que compra).

11. **Saldo virtual:** reembolsos/ajustes se abonan a Usuario.saldoVirtual (reutilizable en compras futuras).

5. Requerimientos

Requerimientos Funcionales (RF)

RF-01. Autenticación con login y password para todo usuario.

RF-02. Crear/gestionar Venues; aprobación previa por Administrador si son sugeridos por organizadores.

RF-03. Crear eventos asociados a un Venue, con fecha/hora; evitar choque de agenda.

RF-04. Configurar Localidades por evento (precio base, numerada/no numerada, cupo).
RF-05. Gestionar Ofertas por Localidad con vigencia (inicio–fin).

RF-06. Emitir tiquetes con id único; aplicar límites por transacción si existen.
RF-07. Calcular precio final por tiquete con oferta/cargo/emitido.

RF-08. Vender TiqueteIndividual y TiqueteMultiple; en múltiples no vender entradas Individuales.

RF-09. Vender PaqueteEventoMultiple (pases de varios eventos) y PaqueteDeluxe (con beneficios).

RF-10. Transferir tiquetes (excepto Deluxe); en múltiples permitir transferencia total o parcial bajo restricciones.

RF-11. Cancelación de eventos: por Admin (reembolso precio menos emisión) o por solicitud del Organizador (reembolso solo base si se autoriza).

RF-12. Registrar compras de promotor como cortesías (sin ingresos).

RF-13. Abonar reembolsos por calamidad a discreción del Administrador.

RF-14. Reportes: Promotor (ganancias sin recargos y % ventas por evento/localidad); Admin (ganancias generales, por fecha, evento, organizador).

RF-15. Persistir todo el estado en archivos fuera de la carpeta del código fuente.

Requerimientos No Funcionales (RNF)

RNF-01. Implementación en Java.

RNF-02. Persistencia en archivos: cargar al iniciar; guardar tras compras, transferencias y cancelaciones.

RNF-03. Integridad: unicidad de Tiquete.id; no duplicar asiento numerado.

RNF-4. Concurrencia: evitar sobreventa de cupos (reservas/escrituras atómicas).

RNF-05. Seguridad por roles: Admin no compra ni transfiere.

RNF-06. Auditoría: historial de compras, transferencias, cancelaciones y reembolsos.

RNF-07. Respaldo de la carpeta de datos para recuperación ante fallos.

6. Programas de Prueba.

1. **Choque de agenda en Venue**

   Objetivo: impedir dos eventos en el mismo venue y misma fecha/hora.
   Pasos:

   - Crear promotor y venue aprobado.

   - Crear Evento A en Venue X el 2025-11-10 20:00.

   - Intentar crear Evento B en Venue X el 2025-11-10 20:00.

   - Crear Evento C en Venue X el 2025-13-10 20:00.

   Verifica: rechazo por conflicto; Evento B no se crea; el estado persistente conserva solo Evento A.

2. **Aprobación previa de Venue sugerido**

   Objetivo: un venue sugerido no puede usarse sin aprobación del admin.
   Pasos:

   - Crear admin.

   - Promotor sugiere Venue Y (no aprobado).

   - Intentar crear un evento en Venue Y → debe fallar.

- Admin aprueba Venue Y.

- Reintentar crear el evento → debe funcionar.

Verifica: validación de “aprobado”.

3. **Precio base uniforme por Localidad**

Objetivo: todas las boletas de la misma localidad comparten precio/atributos.
Pasos:

- Crear Localidad “Platea” (precio base P, numerada).

- Crear 1 cliente.

- Realizar compra de N tiquetes de “Platea”.

- Consultar cada tiquete emitido.

Verifica: todos muestran el mismo precio base y mismas características de la Localidad.

4. **Asientos numerados únicos**

Objetivo: no duplicar asiento en misma localidad/evento.
Pasos:

- Crear Localidad numerada “VIP” con cupo.

- Crear cliente.

- comprar tiquete en asiento A-12 → OK.

- Intentar comprar otro tiquete asiento A-12 (mismo evento/localidad) → debe fallar.

Verifica: bloqueo por duplicado; el segundo tiquete no se crea ni consume cupo.

5. **Vigencia de Oferta**

Objetivo: aplicar descuento solo dentro de la ventana [inicio, fin].
Pasos:

- Crear Oferta del 10% del 2025-10-01 al 2025-10-05 para “General”.
- Simular compra el 2025-09-30 → sin descuento.
- Simular compra el 2025-10-03 → con 10% descuento.
- Simular compra el 2025-10-06 → sin descuento.

Verifica: descuento solo dentro de la ventana, incluidos bordes (inicio y fin).

6. **Fórmula de precio final por tiquete**

Objetivo: validar precioFinal = precioBase*(1−descuento) + cargoServicio + cuotaFija.
Pasos:

- Configurar cargoServicio% y cuotaFija.
- Localidad con precio base conocido; Oferta vigente conocida.
- Emitir y calcular precio final.

Verifica: el precio final coincide con la fórmula (tolerancia 0 en entero/centavos).

7. **Límite por transacción (tiquetes individuales)**

Objetivo: aplicar tope máximo por transacción.
Pasos:

- Configurar tope T (p. ej., 4).
- Intentar comprar 5 individuales en una sola transacción → rechazar.
- Comprar 4 → aceptar.

Verifica: la regla se evalúa por transacción; saldos/cupos coherentes.

8. **Límite por transacción (paquetes múltiples)**

Objetivo: en múltiples, el tope se aplica al número de paquetes, no a entradas internas.
Pasos:

- Tope T=2.
- Cada paquete incluye 6 entradas.
- Comprar 3 paquetes en una transacción → rechazar.
- Comprar 2 paquetes (12 entradas internas) → aceptar.

Verifica: el criterio se aplica a paquetes; no importa la suma de entradas internas.

9. **Transferencia permitida (no Deluxe)**

Objetivo: transferir cualquier tiquete salvo Deluxe.
Pasos:

- Comprar TiqueteIndividual, TiqueteMultiple, PaqueteDeluxe.
- Transferir el Individual → OK.
- Transferir el Múltiple → OK (ver 10).
- Intentar transferir el Deluxe → debe fallar.

Verifica: regla de no-transferencia en Deluxe; registro de transferencias exitosas.

10. Múltiple: transferencia total/parcial y bloqueo por entradas vencidas/transferidas

Objetivo: reglas de transferencia en paquetes múltiples.
Pasos:

- Emitir paquete (10 entradas mismo evento).
- Transferir parcial 3 entradas a Usuario B → OK; paquete queda con 7 activas.
- Marcar 1 entrada como vencida o transferirla a otro usuario.
- Intentar transferir el paquete completo restante → debe fallar (hay al menos una vencida/transferida).

Verifica: parcial permitida, total bloqueada si existe una entrada vencida/ya transferida.

11. El administrador no compra ni posee tiquetes

Objetivo: Admin no puede comprar/poseer/transferir.
Pasos:

- Autenticar Admin.
- Intentar comprar cualquier tiquete → rechazo.
- Intentar ser destinatario de una transferencia → rechazo.

Verifica: prohibiciones activas y trazas de intentos fallidos.

12. Cancelación por Admin → reembolso (precio−emisión)

Objetivo: reembolso correcto cuando Admin cancela.
Pasos:

- Compra un tiquete con precio final conocido (registrar precio base y cuotaFija).
- Admin cancela el evento.
- Ver saldo virtual del comprador: precio pagado − cuotaFijaEmisión.

Verifica: cálculo exacto; movimiento en "saldo virtual" y en registros financieros.

13. Cancelación solicitada por Organizador → reembolso solo base (si Admin autoriza)

Objetivo: reembolsar solo precio base cuando el organizador pide cancelación y Admin autoriza.
Pasos:

- Compra con oferta/cargos aplicados.
- Organizador solicita cancelación; Admin autoriza.
- Ver saldo virtual del comprador: solo precio base (sin cargos/emitidos).

Verifica: cargos quedan en la tiquetera; políticas reflejadas en reporte.

14. Cortesías del promotor (sin ingresos)

Objetivo: compras del propio promotor se registran como cortesías.
Pasos:

- El promotor compra tiquetes de su propio evento.
- Registrar venta y consultar finanzas del promotor y de la plataforma.

Verifica: 0 ingresos para promotor y plataforma; los tiquetes constan como cortesía.

15. Persistencia, unicidad y recuperación de estado

Objetivo: todo persiste en archivos; IDs únicos.
Pasos:

- Realizar una serie de operaciones: crear usuarios, venues, eventos, emitir/transferir, crear ofertas.
- Verificar que no existan IDs de tiquete duplicados.
- Reiniciar la app y cargar desde archivos.
- Re-verificar: eventos, ofertas, cupos, propietarios, saldos virtuales y transferencias.

Verifica: estado restaurado 1:1; sin duplicados; integridad de relaciones y saldos.

16. Consulta ganancias de administradores y promotores

Objetivo: Evidenciar cumplimiento de ingresos para promotores (precio base por tiquete) como para administradores (sobrecargos)

Pasos:

- Crear promotor, administrador, 2 eventos en fechas distintas, 2 localidades y N clientes.

- Hacer que los N clientes realicen 5 compras de tiquetes individuales para una localidad en uno de los eventos.

- Realizar el mismo proceso para la otra localidad y evento.

- Consultar ganancias de promotor por evento y ganancias generales.

- consultar ganancias de admin por evento, fecha, promotor y ganancias generales.

Verifica: congu