Entrega 2
Texto de decision

Enfoque general:

Diseñamos el sistema priorizando el flujo real de venta y gestión de tiquetes: precio base del organizador, sobrecargos del administrador y reglas de paquetes (palco, temporada y deluxe). Esa realidad y la estructura que definimos determinó que las entidades Evento, Venue, Localidad, Tiquete y Paquetes quedaran muy visibles y que los roles operativos (Administrador, Organizador y Cliente) concentraran acciones de alto nivel como crear, aprobar, comprar, transferir, cancelar y reembolsar. Este alineamiento con los casos de uso es consistente con las restricciones y programas de prueba definidos en el documento (precios multicapa, límites por transacción, ofertas por localidad, transferencias con verificación y reportes por rol).

Decisiones clave de modelo

Localidades y eventos:

Modelamos Localidad como una clase que se relaciona y es configurable de un Venue para cada Evento, porque el contexto exige reorganizar localidades por evento y ofrecer descuentos por localidad en ventanas de tiempo. Con esto evitamos acoplar la "plantilla" del Venue con la "configuración" de cada show.

Tiquetes:

Definimos Tiquete con identificador, fecha,hora, precio base, transferibilidad y estado de venta. Esto responde a la trazabilidad requerida para transferencias/reembolsos y a la existencia de localidades numeradas (caso del asiento). Incorporamos la subclase TiqueteVendidoNumerado para capturar el número de asiento cuando la localidad es numerada.

Paquetes

Creamos paquetes explícitos porque:

- el precio del paquete puede diferir de la suma de tiquetes,
- el límite por transacción aplica al paquete completo,
- Deluxe incluye beneficios adicionales y no es transferible.
  Esto justifica que el paquete tenga precio propio y reglas de transferencia distintas a las de un tiquete suelto.

tiquetemultiTemporada como multi-evento

Para los pases de temporada optamos por MultiEvento (clase que contiene relaciones) a fin de poder vender la temporada como un "evento contenedor", manteniendo coherencia con el evento

Decisiones sobre roles y por qué quedan un poco cargados

Administrador:

Entrega 2
Texto de decision

El administrador registra/aprueba venues, fija sobrecargos y cuota fija por evento, cancela y aprueba reembolsos, y consulta ganancias por sobrecargos. Decidimos colocar estos métodos en la clase del rol para que los programas de prueba llamen directamente a la acción esperada sin intermediación innecesaria en esta primera entrega.

Organizador de eventos:

El organizador crea eventos, define localidades y ofertas, genera tiquetes y paquetes y consulta sus ingresos por precio base. Además, puede comprar como cliente, pero si compra de su propio evento, se considera cortesía (no ingresa dinero). Ubicamos estos métodos en el rol para que el flujo sea directo y fiel a la especificación.

Cliente :

El cliente compra tiquetes/paquetes, transfiere (con login del receptor y password del emisor), solicita reembolsos y consulta saldo. Centralizamos estas operaciones en Cliente para simplificar las pruebas de aceptación de compra, transferencia y reembolso, y poder verificar límites por transacción y aplicación de ofertas por localidad.

En resumen, sí dejamos a Administrador, Organizador y Cliente con muchos métodos porque eso acorta la distancia entre requerimiento y código durante la primera iteración y permite ejecutar los 7 programas de prueba tal como están enumerados.

Ventajas del diseño actual:

1. **Trazabilidad directa con casos de uso:** cada acción de los programas de prueba existe como un método del rol correspondiente, lo que agiliza la verificación funcional temprana.
2. **Dominio explícito:** entidades y atributos esenciales (precios, recargos, asiento, transferible, paquetes, temporada) están modelados de forma clara, facilitando discusiones y ajustes de reglas.
3. **Reglas complejas encapsuladas en objetos del negocio:** paquetes con precio propio y políticas de transferencia están representados como clases, lo que nos permite diferenciar **Deluxe** de **Múltiple** y **Temporada** sin ramificaciones frágiles.
4. **Compras y transferencias realistas:** integramos autenticación en las transferencias y consideramos cortesías del organizador, coherente con las reglas del documento.

Conclusión:

En Esta estructura privilegiamos la claridad operativa: los roles están “cargados” a propósito para que los flujos del documento se mapeen con métodos y los programas de prueba se ejecuten sin problemass. La estructura elegida nos dio rapidez de entrega y alineación con la situacion (precios multicapa, paquetes, temporada y transferencias verificadas).

Entrega 2
Texto de decision