

Documento de Análisis – Proyecto Boletamaster

a. Modelo de dominio

Clase	Atributos	Relaciones
Usuario (abstracta)	login: String password: String nombre: String saldo: double	Superclase de Cliente, Organizador y Administrador
Cliente	idCliente: String	El Cliente se conecta con Tiquete (boletas que adquiere) y con Transacción (operaciones de compra o reembolso).
Organizador	idOrganizador: int	El Organizador está conectado con Evento , porque es quien los crea y gestiona, y con Oferta , porque define los descuentos aplicados a localidades de sus eventos.
Administrador	idAdministrador: String	El Administrador , como subclase de Usuario , se conecta con Venue (creación/aprobación) y con Evento (cancelación).
Evento	idEvento: int nombre: String fecha: Date hora: Time tipoEvento: TipoEvento	El Evento está ligado por composición con Venue (se realiza en un lugar específico), se conecta con Tiquete (define las entradas disponibles) y con TiqueteTemporada (puede formar parte de pases múltiples), además de relacionarse con el Organizador , que lo crea y gestiona.

Venue	idVenue: int nombre: String ubicacion: String capacidad: int restriccionesUso: String	El Venue se relaciona con el Administrador (puede crearlo o aprobarlo), tiene composición con Evento (los eventos ocurren en un lugar específico) y se conecta con Localidad , que define la distribución interna del espacio.
Localidad	nombre: String precioBase: double numerada: boolean numAsientos: int	La Localidad forma parte por composición del Venue (secciones del lugar), se conecta con Tiquete (las boletas dependen de la localidad) y con Oferta , que aplica descuentos sobre ella.
Tiquete (abstracta)	- idTiquete: int - precio: double - cargoServicio: double - cargoEmision: double - estado: String “Como el enunciado dice que la fecha y hora del tiquete dependen del evento, no es un atributo propio del Tiquete, sino que viene del Evento.”	El Tiquete se relaciona con el Cliente (quien lo adquiere), con el Evento (al que da acceso), con la Localidad (que determina su precio y características) y con la Transacción (obligatoria para su compra). Además, es clase padre de TiqueteBásico , TiqueteMúltiple , TiqueteTemporada y PaqueteDeluxe .
TiqueteBasico	numeroAsiento: int localidadNumerada : bool	Hereda de Tiquete
TiqueteNumerado	numeroAsiento: int	Hereda de Tiquete
TiqueteMultiple	cantidadEntradas: int precioTotal: double	Compone Tiquetes Hereda de Tiquete
PaqueteDeluxe	beneficios: String tiquetesIncluidos: List<Tiquete>	Hereda Tiquetes Agregacion Tiquetes

Transaccion	idTransaccion: int fecha: Date monto: double	La Transacción está vinculada al Cliente que la realiza y tiene una relación de obligatoriedad con el Tiquete , ya que toda compra de boletas debe quedar registrada en una transacción.
Oferta	descuento: double fechaInicio: Date fechaFin: Date	La Oferta es creada por un Organizador y aplica sobre una Localidad , estableciendo descuentos en sus tiquetes durante un tiempo definido.

b. Restricciones del proyecto

- Cada tiquete debe tener un identificador único.
- Cada tiquete tiene un precio base, un cargo porcentual por servicio y una cuota fija de emisión.
- Puede existir un número máximo de tiquetes por transacción.
- Un cliente puede transferir un tiquete a otro, validando login y contraseña.
- Los tiquetes Deluxe no se pueden transferir.
- Los tiquetes múltiples pueden transferirse de forma individual o completa, pero no como paquete completo si ya hay vencidos o transferidos.
- Los tiquetes múltiples no pueden venderse individualmente.
- En los tiquetes múltiples y de temporada, el tope de compra aplica al paquete, no a los tiquetes internos.
- El precio de los paquetes puede ser distinto a la suma de los tiquetes que incluye.
- Todo evento debe tener un organizador (no puede existir evento sin promotor).
- Todo evento debe tener un venue asociado.
- Un venue no puede tener dos eventos el mismo día.
- Las localidades no son inherentes al venue (se configuran según el evento).
- Todas las boletas de una misma localidad tienen el mismo precio.
- Si una localidad es numerada, cada tiquete debe tener un asiento único en esa localidad.
- Un organizador puede generar ofertas sobre localidades de un evento.
- Una oferta aplica un descuento porcentual en una ventana de tiempo definida.
- Los venues sugeridos por promotores deben ser aprobados por el administrador antes de usarse.
- El administrador fija el cargo por servicio y la cuota fija de emisión.

- El administrador puede establecer tarifas distintas según tipo de evento.
- El administrador puede cancelar un evento y definir el tipo de reembolso.
- Un cliente puede pedir reembolso por calamidad, a discreción del administrador.
- Todos los usuarios tienen un saldo virtual inicial en 0, donde se cargan los reembolsos.
- El administrador no puede comprar boletos.

c. Programas de prueba

- **Prueba de Cliente–Transacción–Boleto**
Crear un cliente, simular que compra un boleto y verificar que queda registrada una transacción.
- **Prueba de Organizador–Evento–Localidad**
Un organizador crea un evento, lo asocia a un venue y define localidades.
Mostrar que cada localidad tiene un precio y puede generar boletos.
- **Prueba de Boleto Básico Numerado**
Crear una localidad numerada y generar boletos con asiento único.
Verificar que no se repiten los números de asiento.
- **Prueba de Boleto Múltiple / Temporada**
Crear un pase múltiple (ej: palco o pase de temporada).
Confirmar que se asocian varios boletos y que el precio no es solo la suma.
- **Prueba de Paquete Deluxe**
Crear un paquete con boletos + beneficios adicionales.
Mostrar que no puede transferirse.
- **Prueba de Administrador–Venue–Evento**
Administrador aprueba un venue.
Administrador cancela un evento y se genera reembolso al saldo del cliente.
- **Prueba de Oferta**
Organizador crea una oferta sobre una localidad.
Verificar que el precio del boleto cambia si la fecha está dentro de la ventana de la oferta.

Historias de Usuario:

Cliente 1.1

Campo	Descripción
Nombre	RF1. Registrarse como cliente
Resumen	El cliente crea una nueva cuenta en la plataforma para poder usar el sistema.
Entradas	Nombre, login, password
Resultados	Nuevo usuario registrado

Campo	Descripción
Nombre	RF2. Reservar tiquete
Resumen	El cliente reserva un tiquete para un evento específico.
Entradas	Evento, tipo de tiquete, pago
Resultados	Tiquete reservado

Campo	Descripción
Nombre	RF3. Pagar tiquete
Resumen	El cliente realiza el pago del tiquete mediante saldo en cuenta u otro medio.
Entradas	Saldo de cuenta
Resultados	Aprobación de compra de tiquete

Organizador 1.1

Campo	Descripción
Nombre	RF4. Crear evento
Resumen	El organizador crea un nuevo evento para la venta de tiquetes.
Entradas	Fecha, Venue, número de tiquetes
Resultados	Nuevo evento creado

Campo	Descripción
Nombre	RF5. Consultar ganancias
Resumen	El organizador consulta las ganancias de sus eventos, sin incluir cargos de la tiquetera.
Entradas	Ninguna
Resultados	Ganancia por evento

Campo	Descripción

Nombre	RF6. Generar oferta
Resumen	El organizador crea una oferta con descuento sobre una localidad de un evento.
Entradas	Localidad, porcentaje de descuento, ventana de tiempo
Resultados	Oferta creada y aplicada a la localidad

Administrador 1.1

Campo	Descripción
Nombre	RF7. Ver ganancias
Resumen	El administrador consulta las ganancias netas de la tiquetera.
Entradas	Ninguna
Resultados	Ganancias calculadas

Campo	Descripción
Nombre	RF8. Fijar cargo fijo
Resumen	El administrador define un cargo adicional fijo para cada ticket vendido.
Entradas	Cargo fijo
Resultados	Cargo aplicado a los tickets

Campo	Descripción
Nombre	RF9. Fijar cargo porcentual
Resumen	El administrador define el porcentaje de cargo por servicio para los tickets según tipo de evento.
Entradas	Cargo porcentual
Resultados	Cargo aplicado a los tickets

Campo	Descripción
Nombre	RF10. Reembolso de ticket
Resumen	El administrador decide si reembolsar o no un ticket bajo ciertas circunstancias.
Entradas	Ticket para reembolsar
Resultados	Monto reembolsado o rechazo del reembolso