

ANÁLISIS PROYECTO

DEL

Daniel Manzanera, 202411443, d.mazanerat@uniandes.edu.co

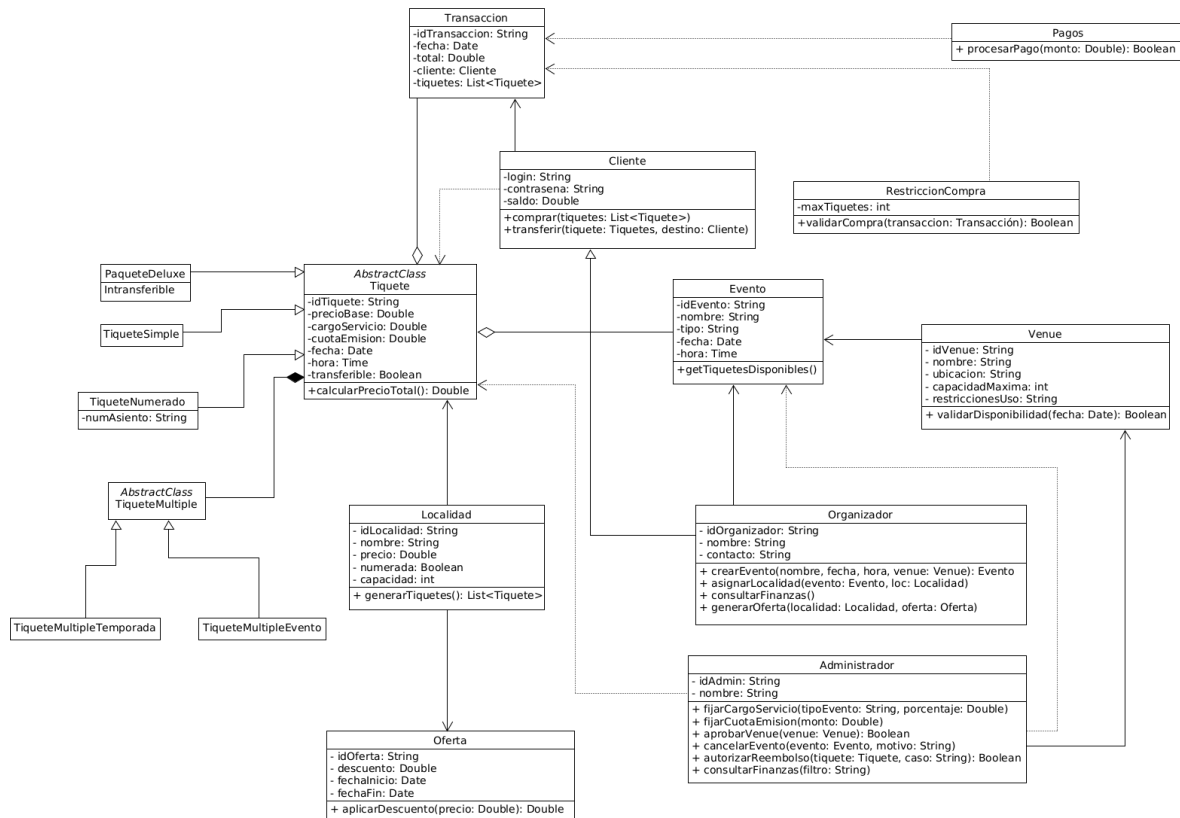
Santiago Gomez, 202221885, s.gomezp2@uniandes.edu.co

Santiago Pinilla, 202315246, s.pinillap2@uniandes.edu.co

Índice

Índice	1
Modelo de dominio	2
Descripción Restricciones del proyecto	2
Restricciones funcionales	2
• Cada cliente debe tener un login y una contraseña válidos	2
• Todo objeto relevante del sistema debe poseer un identificador único	2
• Existe un máximo de tiquetes por transacción	3
• Cada evento está asociado a una fecha y hora	3
• Las localidades deben respetar su capacidad máxima	3
• Un venue no puede albergar más de un evento en la misma fecha	3
• Los venues propuestos por organizadores deben ser aprobados por el administrador	3
• La gestión de ofertas está limitada por fechas de vigencia	4
Restricciones técnicas	4
Descripción de los programas de prueba	4
Criterios de selección de programas de prueba	4
Programas de prueba definidos	5
• Programa 1: Creación de evento y asignación de localidades	5
• Programa 2: Compra de tiquetes	5
• Programa 3: Transferencia de tiquetes entre usuarios	5
• Programa 4: Cancelación de evento y reembolsos	5

Modelo de dominio



Descripción Restricciones del proyecto

Restricciones funcionales

- Cada cliente debe tener un login y una contraseña válidos

Esta restricción se implementa en la clase Cliente, la cual se encarga de validar que el usuario tenga una cuenta activa y correctamente autenticada. Solo los clientes que hayan ingresado con credenciales válidas podrán realizar operaciones en el sistema, tales como la compra o la transferencia de tiquetes. Esto garantiza seguridad y evita accesos no autorizados.

- Todo objeto relevante del sistema debe poseer un identificador único

Para evitar fraudes o duplicaciones, se define que las clases Evento, Transacción,

Tiquete, Localidad, Oferta, Organizador y Venue deben contener un atributo identificador único (ID). En el caso de los tiquetes, este ID es fundamental para prevenir falsificaciones; en los eventos y transacciones, asegura trazabilidad y control de la información dentro de la plataforma.

- **Existe un máximo de tiquetes por transacción**

Con el fin de regular la reventa y evitar abusos en el sistema, se establece un límite máximo de tiquetes que un cliente puede adquirir en una sola transacción. Este límite es un parámetro configurable por el Administrador de la plataforma. De esta manera, se asegura que la compra sea válida únicamente si cumple con la cantidad de tiquetes permitidos.

- **Cada evento está asociado a una fecha y hora**

La clase Evento contiene los atributos de fecha y hora, los cuales son verificados en cada transacción. Al momento de la compra, el sistema comprueba que el evento exista y que aún tenga tiquetes disponibles. Si la fecha del evento ya expiró o no quedan localidades habilitadas, la compra será rechazada. Esto garantiza la coherencia de la información y evita ventas inválidas.

- **Las localidades deben respetar su capacidad máxima**

La clase Localidad define el número máximo de tiquetes que pueden generarse para una sección específica de un evento. Bajo ninguna circunstancia se podrán crear más tiquetes que la capacidad definida. En el caso de localidades numeradas, cada tiquete debe tener un número de asiento único dentro de la misma localidad.

- **Un venue no puede albergar más de un evento en la misma fecha**

La clase Venue valida la disponibilidad de un escenario verificando que no existan eventos programados en la misma fecha. Además, debe cumplir con las restricciones de uso definidas en el atributo restriccionesUso.

- **Los venues propuestos por organizadores deben ser aprobados por el administrador**

Cuando un organizador solicita el uso de un nuevo Venue, este debe ser validado y aprobado por el Administrador antes de que pueda ser asignado a un evento. Esto garantiza

que el escenario cumpla con las condiciones necesarias para su utilización.

- **La gestión de ofertas está limitada por fechas de vigencia**

La clase Oferta establece un rango de fechas (fechaInicio y fechaFin) durante el cual puede aplicarse un descuento a una localidad de un evento. Fuera de este rango, la oferta no es válida. Además, el valor del descuento nunca puede superar el 100 % del precio base del tiquete.

Restricciones técnicas

- “Toda la información debe ser persistente. La información debe almacenarse en archivos (pueden ser planos, en el formato que ustedes definan, o binarios), dentro de una carpeta y se puede suponer que sólo la aplicación va a escribir y leer de esa carpeta (ningún usuario malicioso va a modificar los archivos que ahí se encuentren sin utilizar la aplicación). La carpeta no puede ser la misma carpeta donde se encuentre el código fuente de la aplicación” (Proyecto 1 Boletamaster, s. f., p. 4).
- “La aplicación debe estar hecha en Java” (Proyecto 1 Boletamaster, s. f., p. 4).
- “La persistencia no necesariamente debe hacerse en un solo archivo: diseñe con cuidado cuántos archivos habrá y cómo van a estar estructurados” (Proyecto 1 Boletamaster, s. f., p. 4).

Descripción de los programas de prueba

Criterios de selección de programas de prueba

Los criterios para el programa de pruebas se tomaron de las necesidades del proyecto y de las funcionalidades más importantes que deben verificarse para evaluar la lógica y su cumplimiento. En esta etapa temprana no es necesario probar todos los casos posibles, sino aquellos que son esenciales para el funcionamiento del sistema. Por ello se definieron pruebas sobre la creación de eventos, la venta de tiquetes, la transferencia entre usuarios y la cancelación de eventos, ya que cada una constituye un núcleo fundamental del proyecto.

Dado que la entrega 1 no exige una interfaz compleja, las pruebas deben ser sencillas, solicitando la mínima cantidad de información posible y mostrando salidas claras que permitan revisar

la lógica implementada. Además, es importante que estas pruebas reflejen el cumplimiento de las restricciones establecidas previamente. Finalmente, la entrega 1 debe servir como base para el desarrollo de las siguientes fases del proyecto (entregas 2 y 3).

Programas de prueba definidos

- **Programa 1: Creación de evento y asignación de localidades**

El principal objetivo de este programa es comprobar que un organizador pueda crear un evento con un venue relacionado, cumpliendo las restricciones mencionadas en el apartado anterior. En la prueba se debe verificar que sea posible registrar el evento con su nombre, fecha y hora, revisando siempre que no exista otro evento el mismo día en ese lugar. Además, el organizador debe poder crear localidades, cada una con una capacidad máxima y un precio definido, y en caso de ser numeradas, asegurarse de que los asientos no se repitan. Cada localidad es la que dicta el precio y la característica de la boleta, de modo que todas las boletas de una misma localidad deben costar lo mismo y, si son numeradas, cada asiento debe ser único dentro de esa localidad (aunque puede repetirse en otra). También es posible que un evento tenga una sola localidad y un único tipo de boleta. En conclusión, este programa de prueba debe permitir armar un evento completo de manera coherente y válida dentro del sistema.

- **Programa 2: Compra de tiquetes**

El objetivo de la compra de tiquetes es, valga la redundancia, poder adquirir tiquetes, y en la prueba se debe constatar que esto sea posible con diferentes tipos: normales, múltiples o paquetes deluxe. A la hora de la compra, el sistema debe tener en cuenta el valor base de la boleta, el cargo porcentual por servicio y el costo fijo de emisión, y además verificar que no se supere el máximo de tiquetes permitidos por persona. Cada tipo de tiquete tiene sus propias características: los normales corresponden a una localidad específica, los múltiples pueden ser palcos o pases de temporada que agrupan varias entradas, y los deluxe incluyen beneficios adicionales. En conclusión, se debe poder comprar tiquetes y verificar su tipo para asegurar que se cumplan las restricciones establecidas.

- **Programa 3: Transferencia de tiquetes entre usuarios**

Con este programa se busca lograr la transferencia de tiquetes a otros usuarios. Para esto, las pruebas deben verificar las diferentes restricciones que existen a la hora de transferir. En el caso de un tiquete normal, el sistema debe pedir las credenciales del dueño original antes de realizar la operación y registrar al usuario que lo recibe. En el caso de los tiquetes múltiples, estos solo se pueden transferir de forma individual mientras no hayan caducado y no se hayan transferido más de una vez. Por otro lado, los tiquetes deluxe no se pueden transferir en ninguna circunstancia. En conclusión, la prueba debe lograr hacer una transferencia de tiquetes exitosa, cumpliendo todos los requisitos establecidos.

- **Programa 4: Cancelación de evento y reembolsos**

Para el último programa se busca poder cancelar un evento y dar su respectivo reembolso. El sistema debe permitir la cancelación y realizar la devolución del dinero sin incluir el costo fijo de emisión. El administrador puede cancelar un evento por decisión propia, pero en ese caso solo se reembolsa el precio base de la boleta. También el usuario puede pedir el reembolso de una boleta por calamidad, aunque este solo se hará efectivo si el administrador lo aprueba. En conclusión, la prueba debe demostrar que se puede realizar la cancelación de un evento con su respectivo reembolso cumpliendo las reglas establecidas.