

# ANÁLISIS DEL PROYECTO Entrega 2 Diseño

Daniel Manzanera, 202411443, [d.manzanerat@uniandes.edu.co](mailto:d.manzanerat@uniandes.edu.co)

Santiago Gomez, 202221885, [s.gomezp2@uniandes.edu.co](mailto:s.gomezp2@uniandes.edu.co)

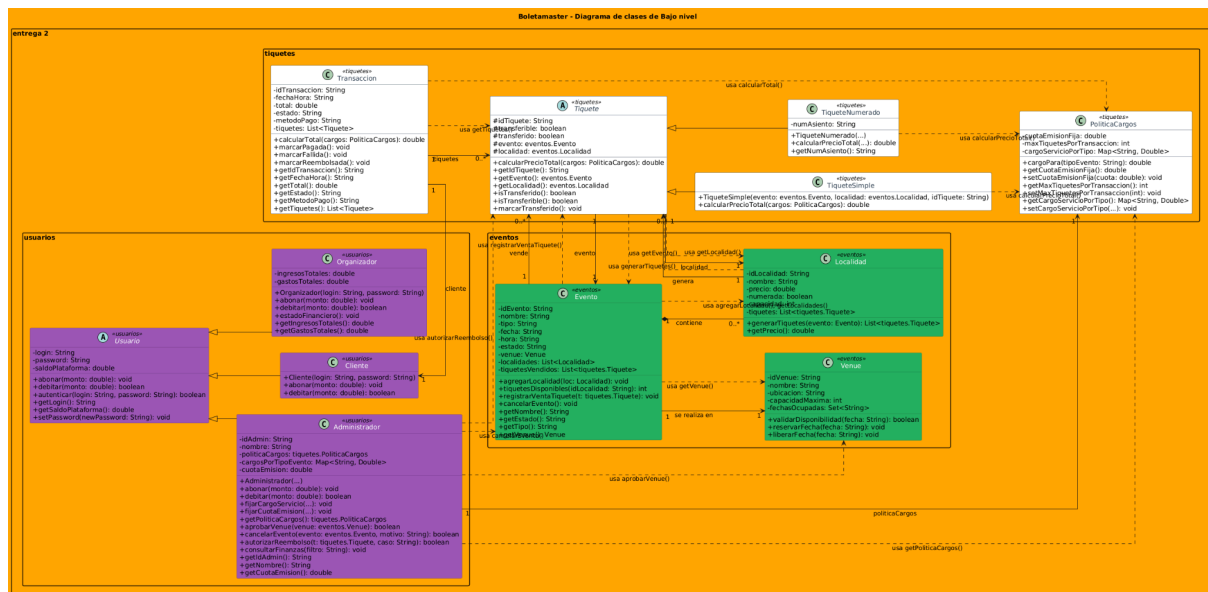
Santiago Pinilla, 202315246, [s.pinillap2@uniandes.edu.co](mailto:s.pinillap2@uniandes.edu.co)

## Índice

Índice	1
Diseño general del sistema	1
Diagrama de de alto nivel	2
Diagrama de bajo nivel	2
Diagramas de secuencia	3
Diagramas de casos de Uso	5
Justificación	8
Organización Lógica y Claridad	9
Uso de Herencia y Polimorfismo	9
Encapsulamiento y Control de Errores	9
Gestión de Reglas de Negocio	10
Diseño Orientado a Casos de Uso	10
Extensibilidad y Mantenibilidad	10

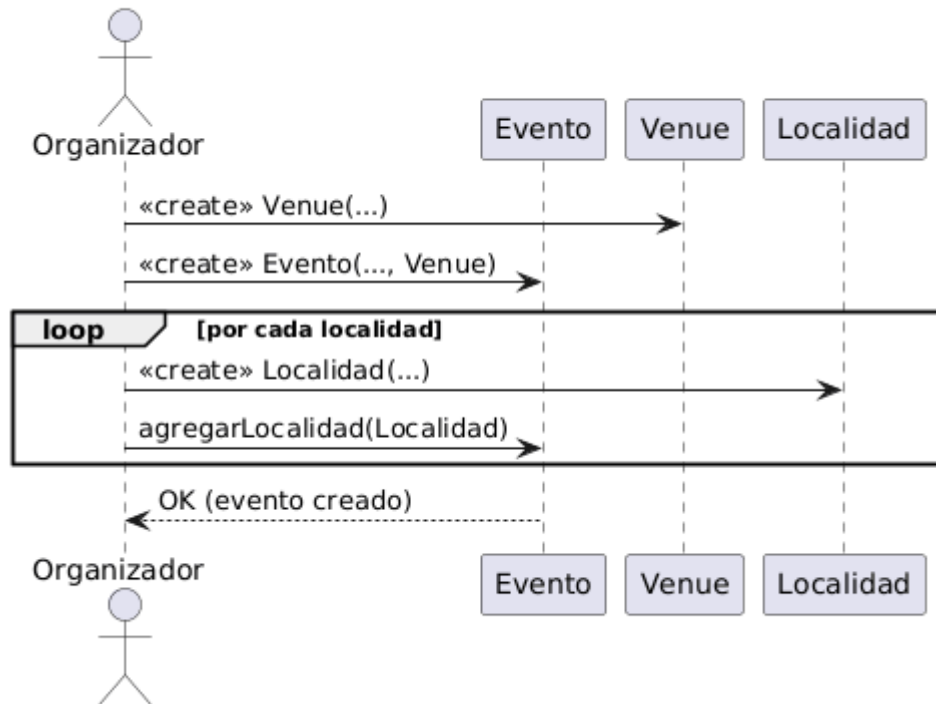
## Diseño general del sistema

// En el repo se encuentran los diagramas más claros para su mejor visualización.

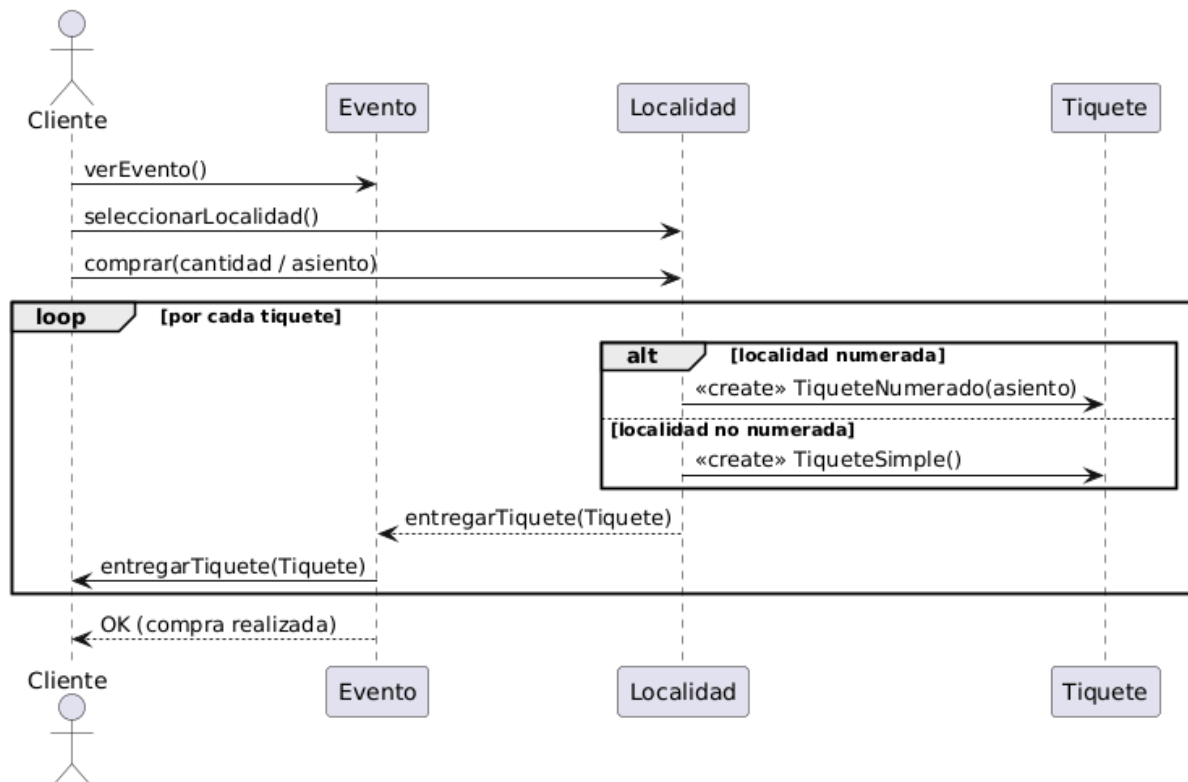


## Diagramas de secuencia

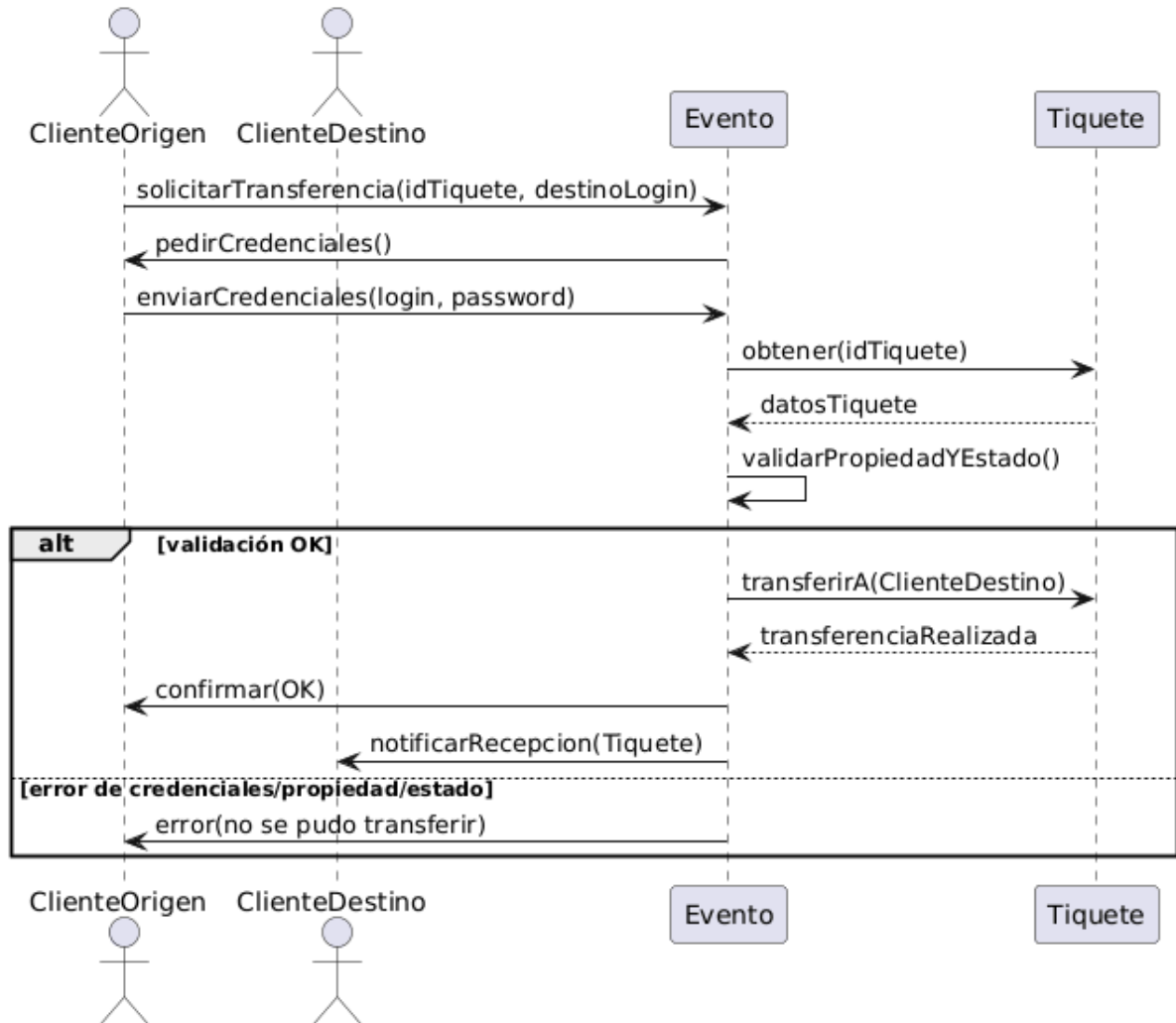
### Crear evento y asignar localidades

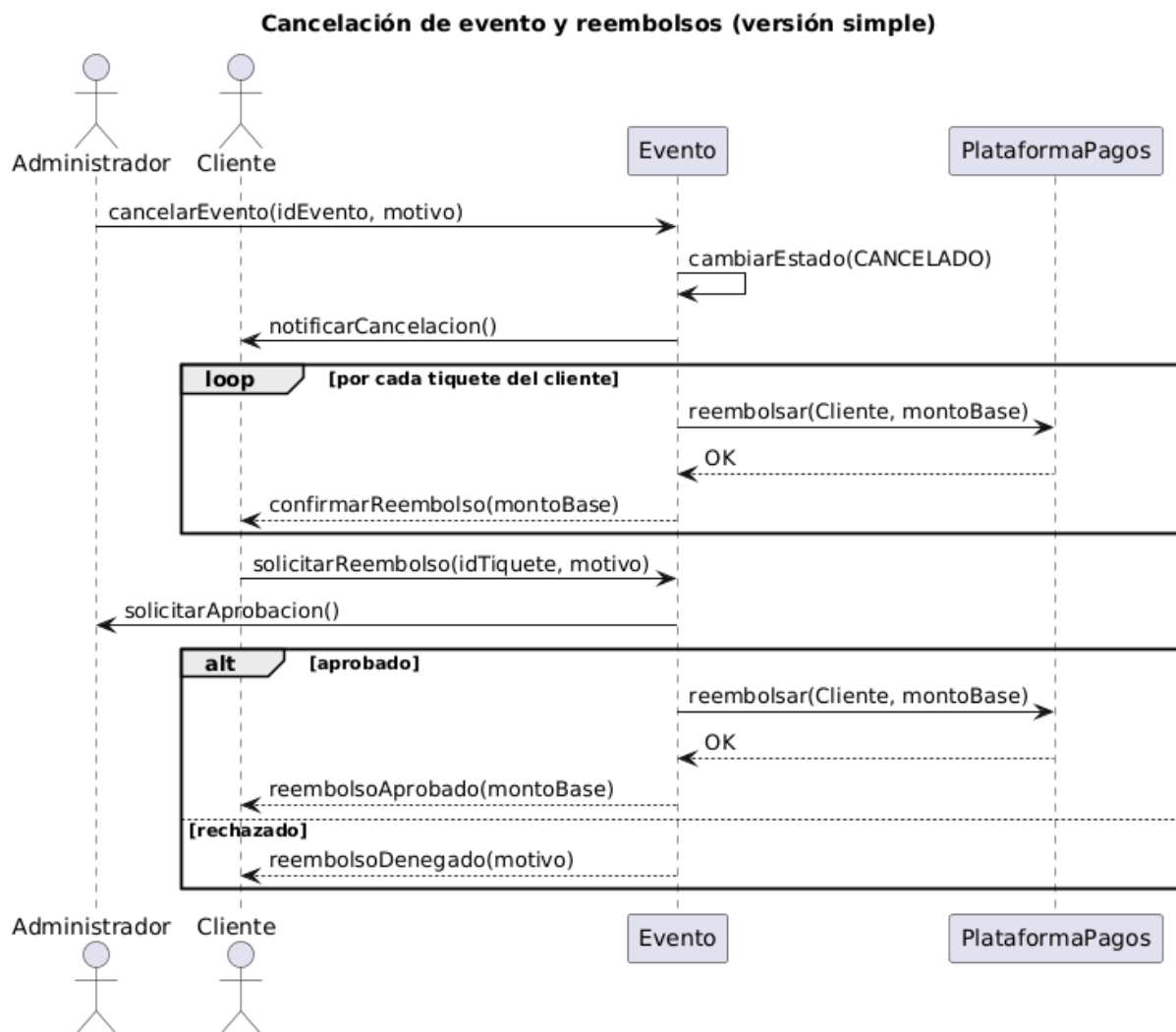


### Compra de tiquetes



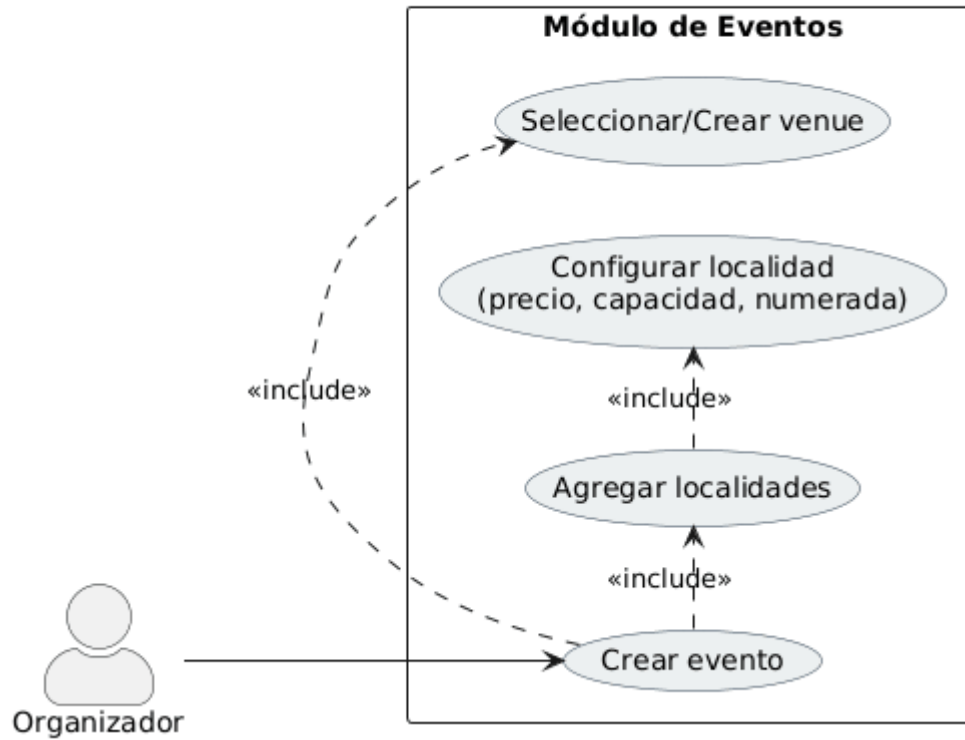
### Transferencia de tickets entre usuarios



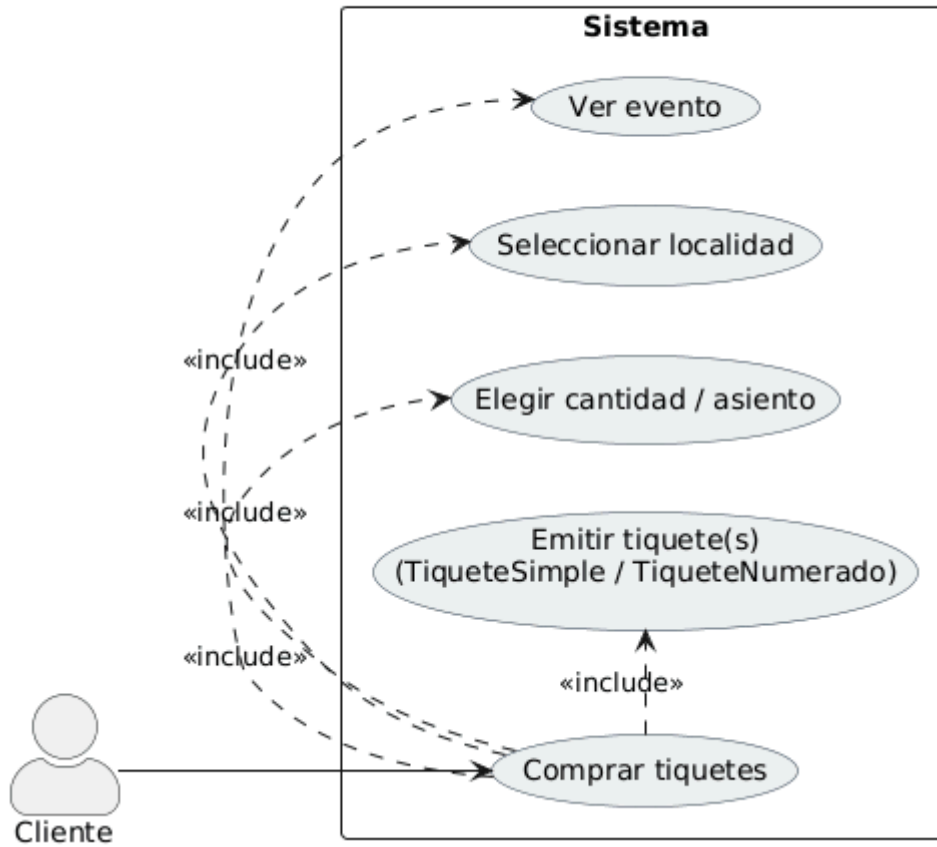


## Diagramas de casos de Uso

### Crear evento y asignar localidades



### Compra de tiquetes



## Justificación



## Organización Lógica y Claridad

El sistema fue diseñado con una arquitectura modular basada en paquetes (eventos, tiquetes, usuarios), lo que promueve una separación clara de responsabilidades y facilita la mantenibilidad del código.

Cada paquete agrupa clases relacionadas con un mismo dominio funcional:

- usuarios: gestiona los distintos tipos de usuarios y sus operaciones.
- eventos: modela la estructura de los eventos, localidades y venues.
- tiquetes: maneja las reglas de negocio asociadas a las políticas de cobro y transacciones.

Esta organización lógica favorece la escalabilidad del sistema y permite integrar nuevas funcionalidades sin afectar las existentes.

## Uso de Herencia y Polimorfismo

La clase abstracta Usuario agrupa atributos y comportamientos comunes.

A partir de ella se extienden tres subclases:

- Cliente: usuario que compra y transfiere tiquetes.
- Organizador: usuario que gestiona ingresos y gastos asociados a eventos.
- Administrador: usuario con permisos de gestión, configuración de políticas y control financiero.

El uso de herencia y polimorfismo facilita la extensión del sistema y evita duplicación de código, permitiendo que cada rol redefina o especialice los métodos según sus necesidades.

## Encapsulamiento y Control de Errores

Se aplicó un estricto control de acceso a los atributos mediante encapsulamiento, garantizando la integridad de los datos.

Además, se incorporaron validaciones en los constructores y métodos para evitar inconsistencias, como descuentos fuera del rango permitido o montos negativos en transacciones.

## Gestión de Reglas de Negocio

La clase `PoliticaCargos` centraliza las reglas de cobro por tipo de evento, permitiendo que el administrador defina cargos variables según la categoría del evento y una cuota de emisión fija.

Esto facilita la actualización de políticas sin alterar otras partes del sistema.

La clase `Transaccion` consolida las compras y reembolsos de tiquetes, manteniendo el estado y los totales calculados según las políticas vigentes.

## Diseño Orientado a Casos de Uso

El diseño responde directamente a los principales casos de uso del sistema:

- Creación y cancelación de eventos.
- Generación de localidades y tiquetes.
- Compra, transferencia y reembolso de tiquetes.
- Aplicación de descuentos y ofertas.

Cada caso de uso se encuentra representado por una clase o método que refleja su comportamiento dentro del dominio del sistema.

## Extensibilidad y Mantenibilidad

El diseño modular y el uso de interfaces hacen que el sistema sea fácilmente ampliable.

Por ejemplo:

- Pueden añadirse nuevos tipos de Usuario o Evento sin modificar la estructura existente.
- Nuevos servicios o políticas de precios pueden integrarse mediante la implementación de las interfaces del paquete servicios.

Esto asegura que el sistema pueda evolucionar con nuevas funcionalidades sin comprometer su estabilidad.

