

Proyecto #1 Boleta Master (Fase diseño e implementación)

David Felipe Alvarez Rodriguez-202410841

Julian Mauricio Sanchez Florez-202412119

A)Diagrama de clases alto nivel	2
B)Diagrama de clases de diseño detallado	3
C)Descripción de Restricciones	3
D)Diagrama Secuencia	4
E)Requerimientos Funcionales	5
1.1 Cliente	5
1.2 Administrador	8
1.3 Organizador de Eventos	9
F)Pruebas funcionales	11
G) Mecanismo persistencia	12

En esta segunda entrega se presenta el diseño detallado del sistema BoletaMaster, incluyendo el diagrama de clases de diseño, los diagramas de secuencia correspondientes a las funcionalidades críticas y una descripción de las principales decisiones arquitectónicas adoptadas.

El objetivo de esta fase es garantizar la coherencia entre el análisis previo y la futura implementación funcional del sistema.

Se definió una clase central llamada BoletaMaster, la cual actúa como fachada principal del sistema, encargada de orquestar las operaciones más relevantes como la compra, transferencia y cancelación de tiquetes.

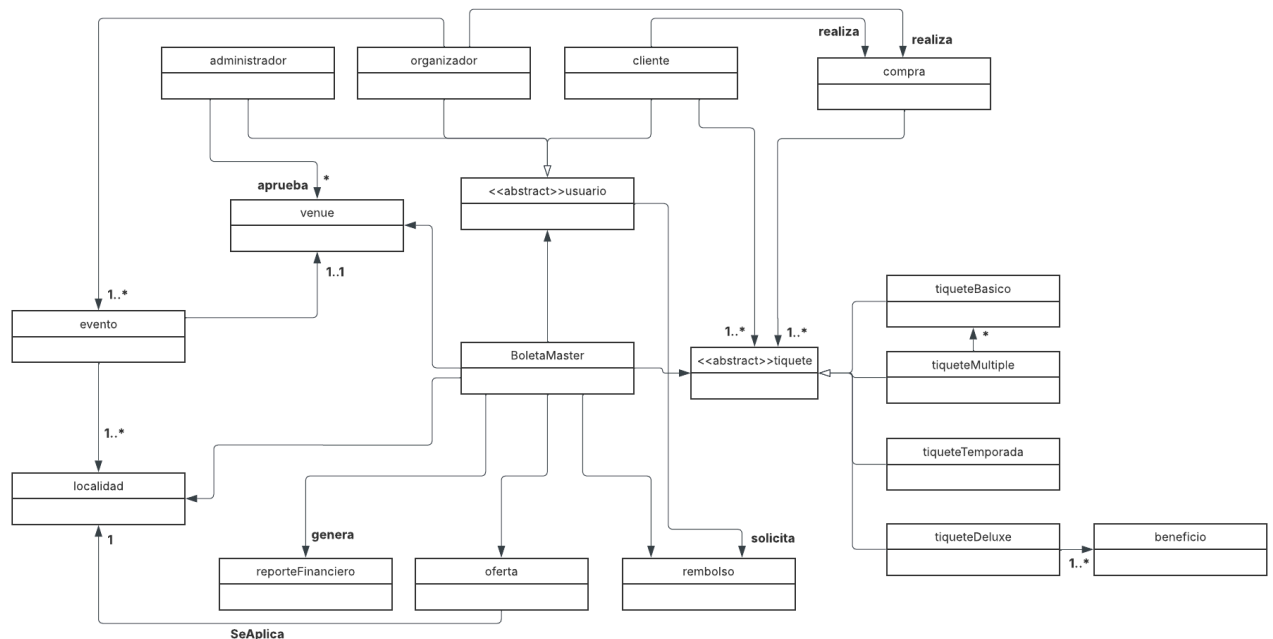
La persistencia de la información se maneja mediante la clase PersistenciaDatos, responsable de guardar y cargar los datos en archivos de texto plano, permitiendo así la recuperación del estado del sistema entre ejecuciones.

Las clases Usuario, Evento, Venue, Localidad y Tiquete representan las principales entidades del dominio, capturando la estructura y comportamiento esencial de las demás clases involucradas en el proceso de venta de boletas.

A continuación, se presenta el diagrama de clases de alto nivel, el cual omite los atributos y métodos con el fin de resaltar las relaciones y dependencias entre las clases principales, brindando una visión global del sistema.

Proyecto #1 Boleta Master (Fase diseño e implementación)

A)Diagrama de clases alto nivel



Con base en el diagrama anterior, se puede identificar la forma en que las clases se relacionan y colaboran entre sí para cumplir con los objetivos del sistema. A continuación, se detallan las principales decisiones de diseño y la justificación de las relaciones establecidas.

Explicación del diagrama y decisiones de diseño

- BoletaMaster centraliza las operaciones del sistema y mantiene las listas de usuarios, eventos, venues y compras. Esta decisión permite un punto de acceso único, simplificando la coordinación entre módulos.
- La clase abstracta Usuario se especializa en Administrador, Organizador y Cliente, lo cual facilita la extensión de comportamientos específicos según el rol.
- Un Organizador crea y gestiona múltiples Eventos, mientras que un Administrador aprueba los Venue donde se realizarán.
- Cada Evento se asocia a un único Venue y se compone de varias Localidades, representando las diferentes zonas o sectores disponibles.
- Las Localidades contienen múltiples Tiquetes, los cuales heredan de la clase abstracta Tiquete para soportar diferentes tipos (básico, múltiple, temporada, deluxe).
- Los Clientes pueden realizar varias Compras, cada una asociada a uno o más Tiquetes.

Ahora, una vez comprendida la estructura general del sistema, se mostrará el diagrama de clases de diseño detallado, el cual incluye los atributos y métodos principales de cada clase.

Este diagrama complementa al de alto nivel, permitiendo observar con mayor precisión la lógica interna, la responsabilidad de cada clase y la forma en que se comunican entre sí durante la ejecución del sistema.

Proyecto #1 Boleta Master (Fase diseño e implementación)

10. Los tickets de temporada o múltiples eventos deben respetar las fechas de vigencia del paquete adquirido.
11. Debe existir un tope máximo de tickets por transacción definido por el sistema.
12. Las ofertas o descuentos sólo pueden aplicarse a una localidad específica y dentro del tiempo de vigencia definido.
13. Los reembolsos sólo pueden generarse en los casos de cancelación o calamidad establecidos por el sistema.
14. Todos los reembolsos se acreditan al saldo virtual del usuario.

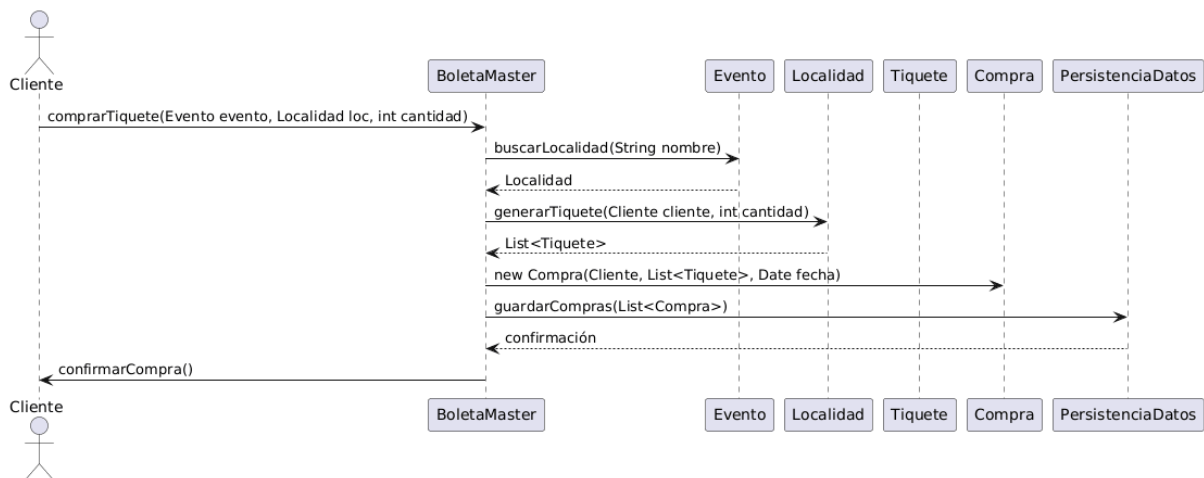
D)Diagrama Secuencia

En esta sección se presentan los diagramas de secuencia de las funcionalidades críticas del sistema.

Estos diagramas ilustran la interacción entre los objetos principales del dominio, destacando el rol de la clase BoletaMaster como coordinador de la lógica.

1.Compra de ticket:

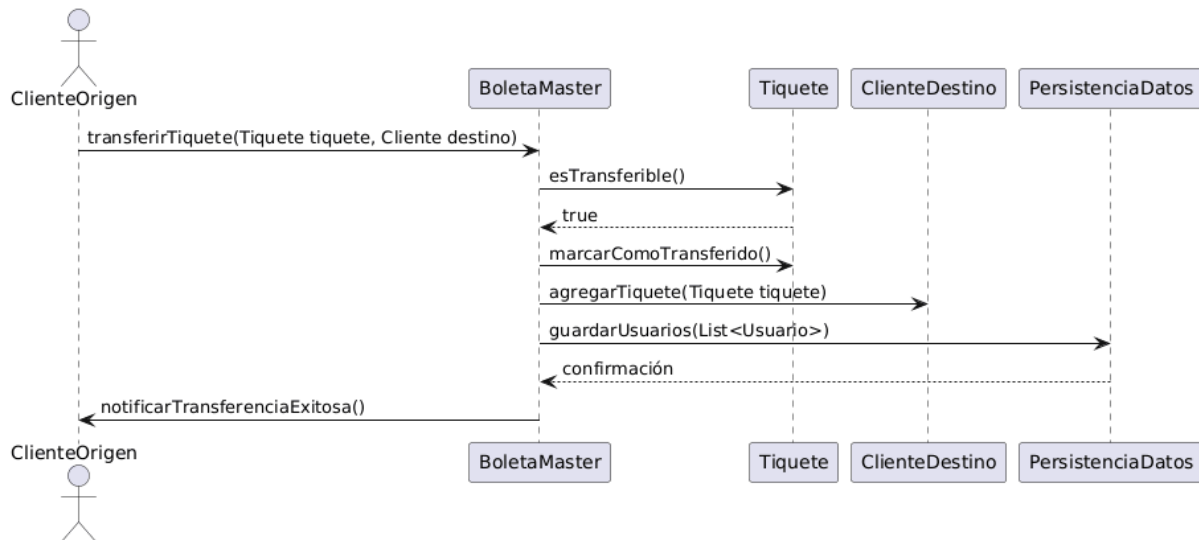
Este diagrama ilustra cómo el sistema gestiona la compra de tickets desde la solicitud del cliente hasta el registro en persistencia.



2.Transferir ticket:

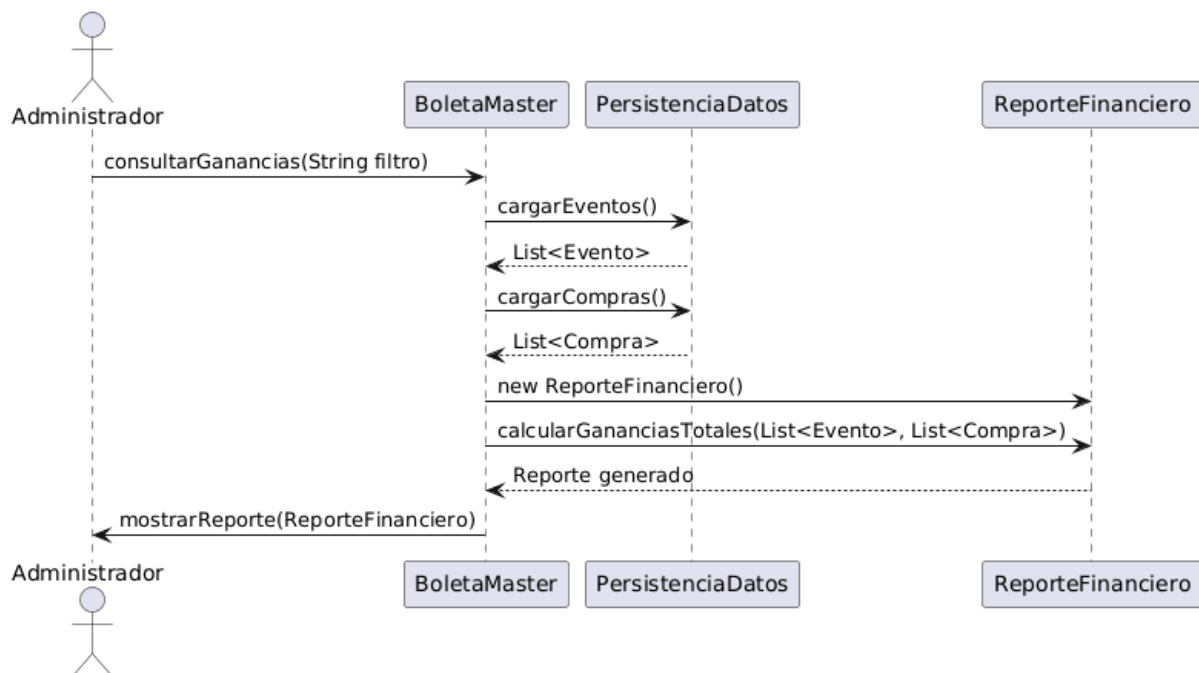
Muestra cómo el cliente puede transferir un ticket a otro usuario, verificando condiciones de transferencia y actualizando la persistencia.

Proyecto #1 Boleta Master (Fase diseño e implementación)



3. Generación reporte financiero:

Representa el proceso de consulta, carga de datos y cálculo de ganancias realizado por el administrador.



Ahora se definirán los requerimientos funcionales, estos requerimientos reflejan el comportamiento esperado del sistema en situaciones reales y sirven como base para el desarrollo de las pruebas y la implementación de cada módulo.

E) Requerimientos Funcionales

1.1 Cliente

Nombre	RF1. Consulta de Catálogo de Eventos
--------	--------------------------------------

Proyecto #1 Boleta Master (Fase diseño e implementación)

Resumen	El sistema debe permitir al cliente visualizar el catálogo completo de conciertos y eventos culturales con sus horarios, disponibilidad y condiciones de asistencia.
Entradas	<ul style="list-style-type: none">• Solicitud de consulta de catálogo.• Filtros de búsqueda por tipo de evento, fecha o ubicación.
Resultados	Demuestra visualización del catálogo de eventos, horarios y condiciones de acceso.

Nombre	RF2. Compra de Tiquetes
Resumen	El sistema debe permitir al cliente realizar la compra de uno o varios tiquetes, mostrando los precios totales, cargos adicionales por servicio e impresión.
Entradas	<ul style="list-style-type: none">• Selección de evento y tipo de tiquete.• Método de pago.• Confirmación de cantidad.
Resultados	Demuestra proceso de compra con cálculo de precios y cargos adicionales.

Nombre	RF3. Transferencia de Tiquetes
Resumen	El sistema debe permitir al cliente transferir tiquetes individuales o paquetes grupales completos a otro usuario de su elección.
Entradas	<ul style="list-style-type: none">• Identificación del tiquete.• Datos del usuario receptor.• Confirmación de transferencia.
Resultados	Demuestra gestión de transferencias de tiquetes individuales o múltiples.

Proyecto #1 Boleta Master (Fase diseño e implementación)

Nombre	RF4. Consulta de Tiquetes Activos
Resumen	El sistema debe permitir al cliente iniciar sesión y consultar los tiquetes activos que posee, junto con su tipo y estado.
Entradas	<ul style="list-style-type: none">● Login y autenticación del cliente.● Consulta de perfil de usuario.
Resultados	Demuestra autenticación del cliente y visualización de tiquetes activos.

Nombre	RF5. Revisión de Condiciones y Restricciones
Resumen	El sistema debe permitir al cliente revisar los detalles de los tiquetes adquiridos, incluyendo restricciones, condiciones de uso, fechas y horarios.
Entradas	<ul style="list-style-type: none">● Selección de tiquete.● Solicitud de información detallada.
Resultados	Demuestra acceso a información detallada de cada tiquete, restricciones y condiciones.

Nombre	RF6. Consulta de Saldo Virtual
Resumen	El sistema debe permitir al cliente acceder a su saldo virtual y revisar el balance disponible en la aplicación.
Entradas	<ul style="list-style-type: none">● Login del cliente.● Solicitud de consulta de saldo.
Resultados	Demuestra visualización del saldo virtual y actualizaciones tras compras o reembolsos.

Proyecto #1 Boleta Master (Fase diseño e implementación)

1.2 Administrador

Nombre	RF1. Modificación de Porcentajes Adicionales
Resumen	El sistema debe permitir al administrador modificar los porcentajes de cargos adicionales aplicables a cada tipo de tickete.
Entradas	<ul style="list-style-type: none">● Selección de tipo de tickete.● Ingreso del nuevo porcentaje de cargo.
Resultados	Demuestra actualización de cargos adicionales y su efecto en precios de venta.

Nombre	RF2. Fijación de Costo Fijo por Emisión
Resumen	El sistema debe permitir al administrador establecer un costo fijo por la emisión o impresión de cada tickete.
Entradas	<ul style="list-style-type: none">● Ingreso del valor de costo fijo.● Confirmación de registro del valor.
Resultados	Demuestra configuración de costo fijo de emisión aplicado a todos los tickets.

Nombre	RF3. Cancelación de Eventos y Reembolsos
Resumen	El sistema debe permitir al administrador cancelar eventos y generar reembolsos automáticos según la causa de cancelación.
Entradas	<ul style="list-style-type: none">● Selección de evento.● Causa de cancelación.● Confirmación del proceso.
Resultados	Demuestra cancelación de evento y cálculo de reembolsos automáticos.

Proyecto #1 Boleta Master (Fase diseño e implementación)

Nombre	RF4. Consulta de Ganancias Generales
Resumen	El sistema debe permitir al administrador acceder a reportes de ganancias por fecha, evento y organizador.
Entradas	<ul style="list-style-type: none">● Filtro de consulta por fecha o categoría.● Solicitud de generación de reporte.
Resultados	Demuestra generación de reportes de ganancias por diferentes criterios.

1.3 Organizador de Eventos

Nombre	RF1. Propuesta de Venue
Resumen	El sistema debe permitir al organizador proponer venues al administrador para su evaluación y aprobación.
Entradas	<ul style="list-style-type: none">● Datos del venue (nombre, ubicación, capacidad).● Solicitud de aprobación.
Resultados	Demuestra flujo de propuesta y validación de venues por parte del administrador.

Nombre	RF2. Creación de Eventos
Resumen	El sistema debe permitir al organizador crear nuevos eventos asignando fechas, localidades, tipos y cantidad de tiquetes disponibles.
Entradas	<ul style="list-style-type: none">● Datos del evento.● Asignación de localidades y tipos de tiquete.
Resultados	Demuestra creación y configuración completa de eventos por el organizador.

Proyecto #1 Boleta Master (Fase diseño e implementación)

Nombre	RF3. Consulta de Estados Financieros
Resumen	El sistema debe permitir al organizador visualizar sus estados financieros sin incluir recargos, con detalle por localidad y evento.
Entradas	<ul style="list-style-type: none">● Selección de evento.● Solicitud de consulta de resultados financieros.
Resultados	Demuestra generación de reportes financieros de ventas por evento y localidad.

Nombre	RF4. Creación de Ofertas Temporales
Resumen	El sistema debe permitir al organizador crear ofertas con descuentos limitados en tiempo para incentivar ventas en ciertas localidades o tiquetes.
Entradas	<ul style="list-style-type: none">● Selección de localidad o tipo de tiquete.● Definición de descuento y duración.
Resultados	Demuestra gestión de ofertas temporales aplicadas a localidades o tipos de tiquete.

Nombre	RF5. Funciones de Cliente
Resumen	El sistema debe permitir al organizador acceder a las funciones básicas de cliente, como la compra y revisión de tiquetes.
Entradas	<ul style="list-style-type: none">● Login del organizador.● Selección de evento y tipo de tiquete.
Resultados	Demuestra compatibilidad del organizador con las funcionalidades de cliente.

Proyecto #1 Boleta Master (Fase diseño e implementación)

Posteriormente, con el propósito de validar que el diseño y las funcionalidades cumplen con los requerimientos definidos, se diseñó un conjunto de programas de prueba. Cada prueba busca demostrar el funcionamiento correcto de una característica clave del sistema y garantizar la coherencia entre el modelo de diseño y la implementación final.

F) Pruebas funcionales

Requerimiento Funcional	Nombre del Programa de Prueba	¿Qué demuestra?
RF1. Consulta de Catálogo de Eventos	TestConsultaCatalogo	Carga datos del sistema y muestra la lista de eventos con su tipo, fecha, lugar y disponibilidad.
RF2. Compra de Tiquetes	TestCompraTiquetes	Cálculo del valor total, cargo de servicio y costo fijo de emisión.
RF3. Transferencia de Tiquetes	TestTransferenciaTiquetes	Permite probar la transferencia de tiquetes individuales o paquetes completos entre usuarios.
TestAdminYOrganizador	TestAdminYOrganizador	Propuesta de venue, aprobación por admin, creación de evento y generación de reporte financiero.

G) Mecanismo persistencia

El sistema implementa persistencia a través de un patrón de diseño basado en interfaces que definen las operaciones de lectura y escritura de datos, como se muestra en el siguiente diagrama.

