

Proyecto 1 DPOO - Boletmaster (Actualizado)

Contexto del problema:

La aplicación Boletmaster tiene como propósito simular el funcionamiento de una empresa dedicada a la venta de tickets para eventos de diversa índole (musicales, culturales, religiosos, deportivos, etc.). Esta plataforma sirve como base de operaciones de la empresa, encargándose de la gestión de eventos, organización de localidades, emisión de tickets y administración general del sistema.

Usuarios:

- Cliente:

Usuario principal de la aplicación. Puede comprar tickets, almacenarlos en su cuenta y consultarlos según su estado (vigentes o no vigentes).

- Organizador:

Usuario que hereda las características del Cliente. Puede crear, administrar y consultar los eventos que organiza, así como revisar su desempeño financiero.

- Administrador:

Usuario con privilegios especiales. Puede aprobar, cancelar eventos y gestionar la información del sistema. Es responsable de la administración general de la plataforma y no puede comprar tickets.

Requerimientos funcionales:

1. La aplicación debe permitir a los organizadores registrar nuevos eventos con su respectiva información (fecha, hora, venue, capacidad, precio base, localidades, etc.).
2. Los clientes deben poder comprar tickets para eventos vigentes, los cuales se almacenan en su cuenta.
3. Los organizadores deben poder consultar los eventos próximos y pasados.
4. El administrador debe poder cancelar eventos o aprobar venues.
5. Cada evento debe asociarse a un organizador, un venue y una lista de localidades.
6. Los tickets deben generarse con un identificador único por cliente.

Requerimientos no funcionales:

1. El sistema debe almacenar los datos de usuarios, eventos, localidades y tickets en estructuras dinámicas (ArrayList o similares).
2. La interfaz debe ser intuitiva y clara.
3. El código debe estar modularizado en paquetes: Usuarios, eventos, tickets y modelo.
4. El sistema debe asegurar la integridad de las relaciones entre entidades (organizador-evento, evento-ticket, etc.).

Restricciones funcionales:

1. Todo evento debe estar asociado a un organizador y a un venue válido.
2. El venue no puede presentar más de un evento en la misma fecha y hora.
3. No se pueden vender más tiquetes que la capacidad máxima del evento.
4. Un cliente no puede tener tiquetes duplicados para el mismo evento.
5. Solo los organizadores pueden crear o modificar eventos.
6. El administrador no puede comprar tiquetes.

Programas de Prueba:

- Programa 1: Carga inicial de datos

Objetivo: Validar que el sistema cree correctamente usuarios, eventos, localidades y tiquetes en memoria.

Entradas: Llamadas a los constructores de las clases Usuario, Evento, Tiquete, Localidad.

Salidas esperadas: Listados creados correctamente.

Requerimiento validado: Persistencia y relaciones entre entidades.

- Programa 2: Creación de evento (Organizador)

Objetivo: Validar que el organizador pueda registrar un evento con su información completa.

Entradas: Datos de evento (nombre, fecha, hora, venue, capacidad, precio base, localidades).

Salidas esperadas: Evento agregado a la lista de eventos próximos.

Requerimiento validado: Creación y gestión de eventos.

- Programa 3: Compra de tiquetes (Cliente)

Objetivo: Verificar que un cliente pueda adquirir tiquetes de eventos activos.

Entradas: ID del evento, tipo de tiquete, cantidad.

Salidas esperadas: Generación de tiquete con identificador único y registro en la lista de tiquetes vigentes.

Requerimiento validado: Compra y emisión de tiquetes.

- Programa 4: Cancelación de evento (Administrador)

Objetivo: Comprobar que el administrador pueda cancelar eventos activos.

Entradas: ID del evento.

Salidas esperadas: Evento marcado como inactivo (estatus = false) y tiquetes afectados actualizados.

Requerimiento validado: Cancelación y control de eventos.

Diagrama de clases:

A continuación se presenta el diagrama UML del sistema (estructura general de clases y relaciones).

Versión HD: https://lucid.app/lucidchart/7ccec7da-a358-4f9e-80d1-22ba0a5b8d58/edit?viewport_loc=-732%2C-2397%2C4091%2C1755%2C0_0&invitationId=inv_f12bd584-ccaf-4e8f-aa19-c26f12821d2d