Proyecto 1 DPOO

- **App venta de tiquetes -**

Nombres: Orlando Barrios Vergara (202422923), José Luis (), Camilo Escobar Ospina (202323442)

Sección: 2

Profesor: Camilo Ortiz

Introducción

El proyecto 1 tiene como finalidad el análisis y creación de un uml para una aplicación de ventas de tiquetes de eventos importantes, como conciertos, eventos deportivos, culturales y religiosos

En esta primera entrega, se debe identificar las Clases y atributos, además sus relaciones y así mismo representar esta información en un diagrama UML.

Este análisis ayudara a hacer un planteamiento en base a un análisis del problema a resolver, para así tener seguir unos lineamientos con poca probabilidad de desviación, llegando de una forma más rápida y segura a la solución del problema.

Uml

Clases y Atributos

Usuario	Cliente	Organizador	Administrador
login: String Contraseña: String nombre: String saldo: int			

Venue	Evento	Localidad	Descuento

id: String nombre: String ubicacion: String maximaCapacidad: int aprobado: boolean	id: String nombre: String fechaYHora: DateTime tipo: TipoEvento estado: EstadoEvento	id: String nombre: String numerada: boolean precio: Money cupoVoletas: int cargoServicio: double	id: String porcentaje: double desde: DateTime hasta: DateTime

Tiquetes

TiqueteSimple/ Tiquete	TiqueteNumerado	TiqueteMultiple	PaqueteDeluxe	Transferencia
id: String codigoUnico: String precioNormal: Money cargoFijoEmision: Money estado: EstadoTiquete	asiento: Int	nombrePaquete: String totalEntradas: int	nombre: String descripcion: String beneficios: String	id: String fecha: DateTime desdeLogin: String haciaLogin: String contraseña: int

Compras y pagos

Carrito	ItemCarrito	Transaccion	MetodoPago	PasarelaExterna	SaldoPlataforma
id: String	id: String cantidad: int	id: String			

creadaEn: DateTime maxTiquetesPorTransaccion: int	precio: Money listaProductors(Lista): Productos	Fecha: DateTime totalAntesDeCargos: Money TotalCargosServicio: Money totalPagado: Money TransaccionAprovada: boolean			

Cancelaciones y reembolsos

CancelacionEvento	Reembolso
id: String comentarios: String fecha: DateTime tipo: TipoCancelacion	id: String fecha: DateTime monto: Money comentarios: String

Algunos valores

TipoEvento: Musical, Cultural, Deportivo, Religioso,

Estado Evento: Activo, Cancelado

EstadoTiquete: Disponible, Vendido, Cancelado

TipoCancelacion: Problemas para asistir, Compra equivocada.

Herencia

Usuario	**Tiquete (abstracta)**	**«interface» MetodoPago**

Subclases: • Cliente • Organizador • Administrador	Subclases: • TiqueteSimple (no numerado) • TiqueteNumerado • TiqueteMultiple	Implementaciones: • PasarelaExterna • SaldoPlataforma

Restricciones del proyecto

1) Toda la información del sistema debe almacenarse en archivos dentro de una carpeta especial, distinta a la del código fuente.

2) Todos los usuarios deben estar registrado con credenciales para autenticación.

3) Un evento solo puede realizarse en un venue aprobado, No pueden existir dos eventos el mismo día en el mismo venue.

4) Todo tiquete tiene un precio base. Cada tiquete debe tener un identificador único para evitar falsificaciones.

5) Tiquetes numerados deben tener un número de asiento único dentro de su localidad. Tiquetes múltiples se venden como conjunto y el límite aplica sobre el paquete. Paquetes Deluxe incluyen beneficios adicionales y no son transferibles.

6) Un cliente puede transferir un tiquete indicando el login del destino y validando su contraseña. No se pueden transferir tiquetes de Paquetes Deluxe.

7) Los descuentos solo aplican a una localidad específica de un evento. Tienen un porcentaje de descuento y una ventana de tiempo definida.

8) Si un evento se cancela por el administrador se devuelve el dinero menos el de los cargos. Si se cancela por solicitud del organizador solo se devuelve el precio base.

9) El administrador no puede comprar tiquetes y puede aprobar venues, fijar tarifas y cancelar eventos.

Programas de prueba

Prueba 1: Carga y Validación de datos

- Propósito: Verificar que se cargue la información desde la carpeta de datos y validar coherencia básica (usuarios, venues, eventos, localidades y tiquetes).
- Entradas mínimas: Ruta de la carpeta de datos.
- Salida esperada: Conteo de objetos cargados y listado de inconsistencias (si existen) o mensaje de validación correcta.

Prueba 2: Creacion de evento y aprobación de Venue

- Propósito: Mostrar que un organizador puede crear un evento y que el administrador aprueba el Venue antes de usarlo.
- Entradas mínimas: Carpeta de datos, identificador de organizador y Venue.
- Salida esperada: Evento creado y asociado a un Venue aprobado. Se impide programar dos eventos en el mismo Venue el mismo día.

Prueba 3: Configuración de localidades y tiquetes

- Propósito: Validar la creación de localidades (numeradas y no numeradas) y la generación de tiquetes correspondientes.
- Entradas mínimas: Carpeta de datos, identificador de evento y definición de localidades.
- Salida esperada: Listado de localidades y tiquetes generados, con verificación de cupos y asientos únicos.

Prueba 4: Venta básica de tiquetes simples

- Propósito: Comprobar la venta de tiquetes simples, aplicando cargos porcentuales y fijos de emisión.
- Entradas mínimas: Carpeta de datos, usuario cliente, evento, localidad y cantidad.

- Salida esperada: Transacción aprobada con cálculo correcto de totales y actualización del inventario.

Prueba 5: Venta de tiquete numerado

- Propósito: Probar la compra en localidades numeradas con asignación de asiento único.

- Entradas mínimas: Carpeta de datos, cliente, evento, localidad y asiento (opcional).

- Salida esperada: Venta registrada y asiento marcado como ocupado. Error si ya estaba vendido.

Prueba 6: Venta de tiquetes múltiples y Parque Deluxe

- Propósito: Validar la compra de paquetes múltiples y del Paquete Deluxe, aplicando las reglas especiales de cada uno.

- Entradas mínimas: Carpeta de datos, cliente, evento y paquete.

- Salida esperada: Venta registrada con detalle de entradas incluidas y beneficios, marcando como no transferible el Paquete Deluxe.

Prueba 7: Transferencia de Tiquetes

- Propósito: Verificar que un usuario pueda transferir tiquetes, excepto los de Paquete Deluxe.
- Entradas mínimas: Carpeta de datos, usuario origen, usuario destino e identificador del tiquete.
- Salida esperada: Cambio de propietario en el tiquete o mensaje de error si no es transferible.

Prueba 8: Descuentos temporales por localidad

- Propósito: Probar aplicación de descuentos porcentuales en un rango de fechas específico.

- Entradas mínimas: Carpeta de datos, evento, localidad, porcentaje y fechas de vigencia.
- Salida esperada: Venta dentro del rango aplica descuento; fuera del rango no lo aplica.

Prueba 9: Cancelación de evento y reembolso

- Propósito: Validar el cálculo de reembolsos cuando un evento es cancelado.
- Entradas mínimas: Carpeta de datos, evento y tipo de cancelación.
- Salida esperada: Reembolsos generados según la causa de cancelación y actualización de saldos virtuales de los usuarios.

Prueba 10: Reportes de ventas y finanzas

- Propósito: Mostrar reportes de ventas para organizadores y ganancias por cargos para la tiquetera.
- Entradas mínimas: Carpeta de datos y filtros opcionales (evento, organizador, fechas).
- Salida esperada: Tablas de ventas y ganancias consistentes con las transacciones previas.

Proyecto 1 Boletamaster