

UNIVERSIDAD MAYOR DE SAN ANDRÉS

CARRERA DE INFORMÁTICA

PROGRAMACIÓN II

INF-121



PROYECTO : 10

Cine: Boletería y Funciones

Integrantes:

CI

Sandoval Calderon Kevin Oscar

9904973

Docente : Lic. Marcelo Aruquipa

Fecha : 13 de noviembre de 2025

LA PAZ - BOLIVIA

2025

Introducción

El proyecto se basa en la creación de un sistema de boletería para un cine. La idea principal es poder gestionar las funciones de películas tanto en salas 2D como 3D, controlar los cupos disponibles y facilitar la venta de entradas de manera más organizada. También queremos incluir el cálculo automático de precios según las tarifas o promociones, el registro de clientes, el registro de ventas y ocupación. En sí lo que se busca es que el sistema ayude a que todo el proceso del cine sea más ordenado y eficiente.

Análisis del contexto y la problemática

En muchos cines aún se manejan las funciones y la venta de entradas de forma manual o sistemas poco eficientes lo que genera errores, confusiones con los horarios y perder el control sobre los cupos disponibles. Además, el calcular precios según promociones, el registro de ventas y clientes se vuelven un caos si no se usa una herramienta informática que ayude con esto.

El proyecto busca resolver este problema de organización y control dentro del cine. Con el sistema de boletería que se quiere implementar, se puede administrar de manera más práctica la cartelera, los cupos y los precios, reduciendo errores. Es una forma de modernizar el proceso, para los clientes y los trabajadores.

Objetivos

Objetivo General

Desarrollar un sistema de boletería para un cine que permita gestionar funciones, salas y ventas de entradas, aplicando los principios de la Programación Orientada a Objetos

Objetivos específicos:

- Diseñar e implementar las clases principales del sistema (Película, Sala, Sala2D, Sala3D, Función, Cliente, Boleto, Venta y Reportes).
- Aplicar herencia y polimorfismo para manejar distintos tipos de salas (2D y 3D) con precios y características diferentes.
- Implementar composición y agregación en las relaciones entre clase.

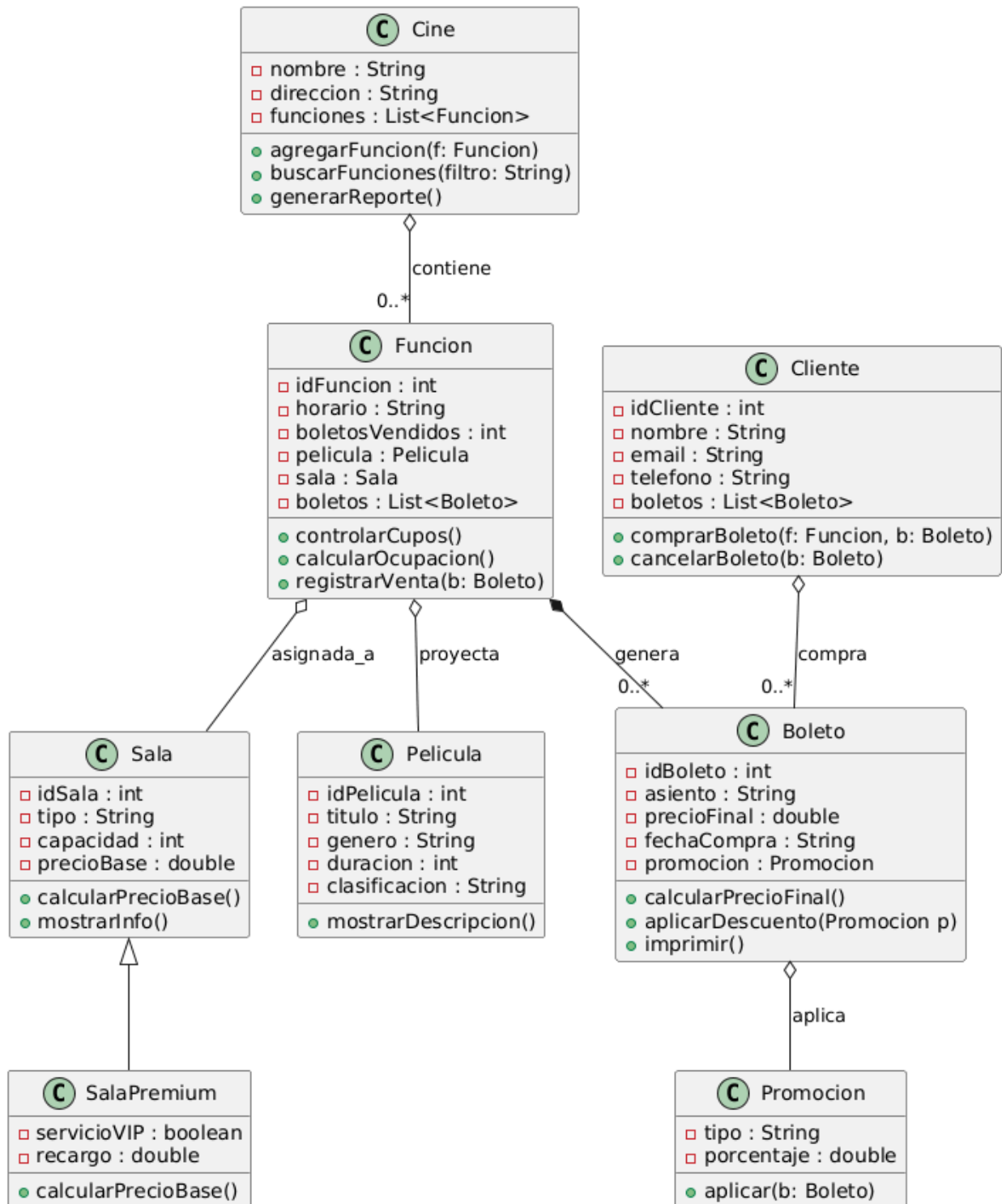
- Desarrollar el registro y venta de boletos con control automático de cupos disponibles por función.
- Implementar persistencia de objetos mediante archivos para guardar y recuperar datos de películas, funciones y ventas.
- Generar reportes diarios de ventas y ocupación de salas dentro del sistema principal (Reportes).

Problemas a resolver

1. Diseño de herencia para salas 2D y 3D, crear una estructura que permita manejar ambas salas con una clase base (Sala) y aplicar polimorfismo en el cálculo del precio de las entradas [**+ calcularPrecioBase()**]
2. Control de cupos por función, evitar que se vendan más boletos de los asientos disponibles en una sala [**-boletosVendidos + controlarCupos()+ registrarVenta(b)**]
3. realizar un filtrado de funciones disponibles. [**+ buscarFunciones(filtro)**]
4. Gestión de promociones o descuentos, para permitir aplicar descuentos automáticos según el tipo de sala, horario o día de la función. [**+aplicarDescuento() + calcularPrecioFinal() + aplicar(b)**]
5. Generación de reportes: calcular automáticamente las ventas totales y la ocupación por función, mostrando un resumen general del día. [**+generarReporte() + calcularOcupacion()**]

Diagrama de clases UML

Sistema de Cine - UML Final Limpio



Descripción

| Clases | Atributo | Descripción del atributo | Relaciones clave |
|--------------------|------------------------|--|---|
| Cine | nombre : String | Nombre del cine | Agrega → Funcion |
| | direccion : String | Dirección física del cine | |
| | funciones : List | Lista de funciones que se proyectan en el cine | |
| Sala | idSala : int | Identificador único de la sala | Compuesta en → Funcion |
| | tipo : String | Tipo de sala (2D, 3D, etc.) | |
| | capacidad : int | Número máximo de asientos disponibles | |
| | precioBase : double | Precio base de la entrada | |
| SalaPremium | servicioVIP : boolean | Indica si la sala tiene servicio VIP | Hereda de → Sala |
| | recargo : double | Recargo adicional por ser sala premium | |
| Funcion | idFuncion : int | Identificador único de la función | Agrega → Boleto, Agrega → Pelicula, Agrega → Sala |
| | horario : String | Hora de la función | |
| | boletosVendidos : int | Número de boletos vendidos hasta el momento | |
| | pelicula : Pelicula | Película que se proyecta en la función | |
| | sala : Sala | Sala donde se realiza la función | |
| | boletos : List | Lista de boletos vendidos para esta función | |
| Pelicula | idPelicula : int | Identificador único de la película | Compuesta en → Funcion |
| | titulo : String | Título de la película | |
| | genero : String | Género cinematográfico | |
| | duracion : int | Duración en minutos de la película | |
| | clasificacion : String | Clasificación por edades de la película | |
| Cliente | idCliente : int | Identificador único del cliente | Agrega → Boleto |
| | nombre : String | Nombre completo del cliente | |
| | email : String | Correo electrónico del cliente | |
| | telefono : String | Teléfono del cliente | |
| | boletos : List | Lista de boletos comprados por el cliente | |

| Clases | Atributo | Descripción del atributo | Relaciones clave |
|------------------|-----------------------|---|---------------------|
| Boleto | idBoleto : int | Identificador único del boleto | Agrega → Promocion |
| | asiento : String | Asiento asignado en la función | |
| | precioFinal : double | Precio final del boleto, incluyendo descuentos | |
| | fechaCompra : String | Fecha en que se compró el boleto | |
| | promocion : Promocion | Promoción aplicada al boleto (si corresponde) | |
| Promocion | tipo : String | Tipo de promoción (porcentaje, especial, día, etc.) | Aplicada a → Boleto |
| | porcentaje : double | Porcentaje de descuento que aplica la promoción | |

Captura de pantallas de ejecución

```

<terminado> Principal (26) [Aplicación Java] C:\Users\\Desktop\p2\pool\plugins\org.eclipse.justj.openj
Precio sala 2D: 15.0
Precio sala Premium: 35.0
Se aplico el descuento de 15.0 a 13.5
Se aplico el descuento de 35.0 a 28.0






=== Boleto Comprados ===
Boleto [idBoleto=1, asiento=A1, precio=13.5, fechaCompra=12/11/2025]
Boleto [idBoleto=2, asiento=A2, precio=28.0, fechaCompra=12/11/2025]
Boleto [idBoleto=3, asiento=B1, precio=35.0, fechaCompra=12/11/2025]

=== Funciones de 'Spider-Man' ===
funcion: Spider-Man 14:00
funcion: Spider-Man 21:00
no se encontro esa Spider-Man

=== Reporte de ventas y ocupación ===
Reporte del cine: Cine Plaza
Spider-Man - 14:00 /Ocupacion: 2.0%
Encanto - 18:00 /Ocupacion: 3.333333333333335%
Spider-Man - 21:00 /Ocupacion: 3.333333333333335%

=== Verificando control de cupos ===
Ocupación función 14:00: 100.0%

```

1.

2.

3.

4.

5.

```

Sala sala2D = new Sala(1, "2D", 50, 15.0);
SalaPremium salaPremium = new SalaPremium(2, 30, 25.0, true, 10.0); // Polimorfismo

System.out.println("Precio sala 2D: " + sala2D.calcularPrecioBase());
System.out.println("Precio sala Premium: " + salaPremium.calcularPrecioBase());

// ===== 20 Crear películas =====
Película película1 = new Película(1, "Spider-Man", "Acción", 130, "PG-13");
Película película2 = new Película(2, "Encanto", "Animación", 95, "G");

// ===== 30 Crear funciones =====
Función función1 = new Función(1, "14:00", película1, sala2D);
Función función2 = new Función(2, "18:00", película2, salaPremium);
Función función3 = new Función(3, "21:00", película1, salaPremium);

// ===== 40 Crear cine =====
Cine cine = new Cine("Cine Plaza", "Av. Siempre Viva 123");
cine.agregarFunción(función1);
cine.agregarFunción(función2);
cine.agregarFunción(función3);

// ===== 50 Crear promociones =====
Promoción promo10 = new Promoción("Descuento 10%", 10.0);
Promoción promo20 = new Promoción("Descuento 20%", 20.0);

// ===== 60 Crear clientes =====
Cliente cliente1 = new Cliente(1, "Kevin", "kevin@mail.com", "77777777");
Cliente cliente2 = new Cliente(2, "Maria", "maria@mail.com", "88888888");

// ===== 70 Comprar boletos (control de cupos + promociones) =====
Boleto boleto1 = new Boleto(1, "A1", sala2D.calcularPrecioBase(), "12/11/2025");
Boleto boleto2 = new Boleto(2, "A2", salaPremium.calcularPrecioBase(), "12/11/2025");
Boleto boleto3 = new Boleto(3, "B1", salaPremium.calcularPrecioBase(), "12/11/2025");

// Aplicar promociones
boleto1.aplicarDescuento(promo10); // Spider-Man 2D con 10%
boleto2.aplicarDescuento(promo20); // Encanto Premium con 20%
// boleto3 sin promoción

// Registrar venta en funciones (control de cupos)
cliente1.comprarBoleto(función1, boleto1);
cliente1.comprarBoleto(función2, boleto2);
cliente2.comprarBoleto(función3, boleto3);

// ===== Mostrar boletos comprados =====
System.out.println("\n=== Boletos Comprados ===");
for (Boleto b : cliente1.getBoletos()) {
    b.imprimir();
}
for (Boleto b : cliente2.getBoletos()) {
    b.imprimir();
}

// ===== Filtrar funciones disponibles por película =====
System.out.println("\n=== Funciones de 'Spider-Man' ===");
cine.buscarFunciones("Spider-Man");

// ===== Generar reporte final del cine =====
System.out.println("\n=== Reporte de ventas y ocupación ===");
cine.generarReporte();

// ===== 1 Verificar control de cupos =====
System.out.println("\n=== Verificando control de cupos ===");
// Intentar llenar la sala 2D
for (int i = 0; i < 60; i++) { // Capacidad 50
    Boleto b = new Boleto(100 + i, "Fila " + (i / 10 + 1) + " Asiento " + (i % 10 + 1),
        sala2D.calcularPrecioBase(), "12/11/2025");
    cliente2.comprarBoleto(función1, b);
}
System.out.println("Ocupación función 14:00: " + función1.calcularOcupacion() + "%");

```