

<!--GRUPAL-->

Sistema de gestión
de entradas para la
Feria internacional
de Loja {

<Por=
"Leandro Saquisari"
"Anthony Romero"/>



Contenidos

01

Introducción

02

Problemática

03

Solución

04

Resultados

05

Conclusiones

Introducción {

El proyecto "Sistema de gestión de entradas para la Feria Internacional de Loja", tiene como objetivo modernizar la venta y control de boletos para este evento anual. Este sistema, desarrollado en Java, permite gestionar entradas normales y para funciones especiales, realizadas los jueves, viernes y sábados, del 30 de agosto al 8 de septiembre de 2024.

Las principales características incluyen:

- Gestión de boletos: Compra de entradas normales y especiales según las fechas de la feria.
- Facturación: Cálculo del monto a pagar, considerando tarifas y promociones.
- Registro de asistencia: Almacenamiento de la cantidad de asistentes a cada evento.
- Generación de estadísticas: Datos sobre afluencia, ganancias y asistencia para decisiones informadas.

Este sistema mejora la experiencia del usuario y proporciona a los organizadores herramientas para una gestión eficiente. }

1. Gestión de boletos:

- Problema: Modelar adecuadamente la compra de boletos, diferenciando entre entradas normales y especiales.
- Solución: La clase Ticket modela los boletos con atributos para tipo, fecha, precio, nombre y cédula. La clase TicketManager maneja la venta y almacenamiento de boletos. La lógica para la compra de boletos se implementa en el método sellTicket en TicketController, que verifica si el tipo es "Especial" y si la fecha es válida para eventos especiales.

```
public class Ticket implements Serializable {  
    private static final long serialVersionUID = 1L;  
    private String type;  
    private LocalDate date;  
    private double price;  
    private String nombre;  
    private String cedula;
```

```
public void sellTicket(Ticket ticket) {  
    soldTickets.add(ticket);  
    Event event = getEventByDate(ticket.getDate());  
    if (event != null) {  
        event.addTicket(ticket);  
    }  
}
```

```
public void sellTicket(String type, LocalDate date, String nombre, String cedula) {  
    if (type.equals("Especial") && !isSpecialEventAllowed(date)) {  
        System.out.println("Las funciones especiales solo se pueden reservar los jueves,  
        return;  
    }  
    double price = ticketManager.calculateTicketPrice(type, date);  
    Ticket ticket = new Ticket(type, date, price, nombre, cedula);  
    ticketManager.sellTicket(ticket);  
}
```

2. Facturación de boletos:

- Problema: Calcular el monto a pagar por cada tipo de entrada, considerando tarifas y promociones.
- Solución: El método `calculateTicketPrice` en `TicketManager` determina el precio del boleto, aplicando un descuento del 10% para eventos especiales los viernes, sábados y domingos.

```
public double calculateTicketPrice(String type, LocalDate date) {  
    double price = type.equals("Especial") ? SPECIAL_TICKET_PRICE : NORMAL_TICKET_PRICE;  
    // Aplicar promociones para días específicos si es necesario  
    if (type.equals("Especial") && (date.getDayOfWeek().getValue() == 5 || date.getDayOfWeek().getValue() == 6 || date.getDayOfWeek().getValue() == 7)) {  
        price *= 0.9; // 10% de descuento los viernes, sábados y domingos  
    }  
    return price;  
}
```

3. Registro de asistencia:

- Problema: Registrar la asistencia a funciones especiales.
- Solución: El método `getAttendanceByEvent` en `TicketManager` cuenta la cantidad de boletos vendidos para un evento específico, almacenados en la lista de boletos de cada evento.

```
public int getAttendanceByEvent(String eventName) {  
    for (Event event : events) {  
        if (event.getName().equals(eventName)) {  
            return event.getTickets().size();  
        }  
    }  
    return 0;  
}
```

4. Generación de estadísticas:

- Problema: Generar estadísticas de la feria como la afluencia total de visitantes, ganancias, y asistencia por día o semana.
- Solución: El método `displayStatistics` en `TicketController` calcula y muestra las ganancias totales, asistencia por evento, día y semana utilizando las colecciones y métodos de agrupamiento.

```
public void displayStatistics() {
    double totalEarnings = ticketManager.calculateTotalEarnings();
    System.out.println("Ganancias Totales: $" + totalEarnings);

    List<Event> events = ticketManager.getEvents();
    for (Event event : events) {
        int attendance = event.getTickets().size();
        System.out.println("Evento: " + event.getName() + ", Fecha: " + event.getDate() + ", Asistencia: " + attendance + " personas.");
    }

    Map<LocalDate, Long> attendanceByDate = ticketManager.getSoldTickets().stream()
        .collect(Collectors.groupingBy(Ticket::getDate, Collectors.counting()));
    System.out.println("Asistencia por día:");
    attendanceByDate.forEach((date, count) -> System.out.println(date + ": " + count + " personas"));

    WeekFields weekFields = WeekFields.of(Locale.getDefault());
    Map<Integer, Long> attendanceByWeek = ticketManager.getSoldTickets().stream()
        .collect(Collectors.groupingBy(ticket -> ticket.getDate().get(weekFields.weekOfWeekBasedYear()), Collectors.counting()));
    System.out.println("Asistencia por semana:");
    attendanceByWeek.forEach((week, count) -> System.out.println("Semana " + week + ": " + count + " personas"));
}
```

5. Persistencia de Datos:

- Descripción: Los datos deben ser guardados y cargados para no perder la información entre sesiones.
- Solución Aplicada: Se implementaron métodos para guardar (saveToFile) y cargar (loadFromFile) el estado del TicketManager utilizando serialización

```
public void saveToFile(String filename) throws IOException {  
    try (ObjectOutputStream oos = new ObjectOutputStream(new FileOutputStream(filename))) {  
        oos.writeObject(this);  
    }  
}  
  
public static TicketManager loadFromFile(String filename) throws IOException, ClassNotFoundException {  
    try (ObjectInputStream ois = new ObjectInputStream(new FileInputStream(filename))) {  
        return (TicketManager) ois.readObject();  
    }  
}
```


Resultados del Sistema de Gestión de Entradas

- **Gestión de Boletos:** Permite la compra de boletos para eventos normales y funciones especiales, con validación de fechas y horarios específicos para funciones especiales.
- **Facturación:** Calcula correctamente los precios de los boletos, aplicando descuentos del 10% para funciones especiales los viernes, sábados y domingos.
- **Registro de Asistencia:** Los boletos vendidos se asocian correctamente a los eventos, actualizando el número de boletos vendidos y la asistencia a cada evento.
- **Estadísticas:** Genera estadísticas detalladas, incluyendo afluencia total de visitantes, ganancias generadas y asistencia por día y semana, facilitando la toma de decisiones.
- **Persistencia de Datos:** Guarda y carga datos de manera confiable, asegurando que la información se mantenga entre sesiones.

Conclusión {

El sistema de gestión de entradas para la Feria Internacional de Loja es funcional y cumple con los requisitos básicos de compra, facturación y registro de asistencia de boletos. La implementación de descuentos y la generación de estadísticas proporcionan un valor añadido significativo, facilitando una mejor gestión y toma de decisiones. Sin embargo, se debe seguir mejorando y testeando el sistema para garantizar su robustez y fiabilidad en un entorno real.

}

<!--GRUPAL-->

Gracias {

}