

Ingeniería en ciencias de la computación



Proyecto Bimestral

Introducción a la programación

Integrantes:

Quito Ordoñez Jorge Luis

Gonzales Hurtado David Israel

Ing. Irene Robalino Pedro Daniel

Periodo octubre 2024 – febrero 2025

Problemática:

Sistema de gestión de entradas para la Feria internacional de Loja

Desarrollar un sistema de gestión de entradas para la feria internacional de Loja, el cual permitirá facturar boletos de entrada a la feria, considerando dos categorías: entradas normales y entradas para funciones especiales, estas últimas relacionadas con presentaciones de artistas nacionales e internacionales, que se realizarán los jueves, viernes y sábados en horario de 5 pm a 2 am.

Características a considerar:

- **Gestión de Boletos:** Crear métodos o funciones para la compra de boletos, donde se puedan adquirir entradas normales y entradas para funciones especiales, considerando las fechas específicas de la feria del 30 de agosto al 8 de septiembre de 2024.
- **Facturación de Boletos:** Implementar un sistema que calcule el monto a pagar por cada tipo de entrada, teniendo en cuenta las tarifas establecidas y las posibles promociones para determinadas funciones en días específicos.
- **Registro de Asistencia:** Desarrollar funciones que registren la asistencia a las funciones especiales, almacenando la cantidad de personas que han asistido a cada evento.
- **Generación de Estadísticas:** Crear métodos que permitan generar estadísticas finales de la feria, incluyendo la afluencia total de visitantes, ganancias generadas y la asistencia por día, semana o función.

Análisis:

Tras analizar los requisitos solicitados para la resolución de la problemática planteada, se ha decidido tomar un enfoque 100% modular al momento de la codificación, es decir nuestra solución va a estar dividida en diferentes archivos java class para reutilizar bloques de código específicos en caso de ser necesario, en base a esta estructura escogida realizamos el análisis de los diferentes bloques de código que vamos a utilizar, además, de la entrada de datos, y los resultados esperados.

1. Entrada de datos:

Ya que nuestro proyecto se trata de la feria internacional de Loja, debemos tomar en cuenta que este evento se realiza en fechas específicas y que el flujo normal en este evento va en el rango de docenas de miles de datos por día, además que se venden dos tipos de entradas dependiendo de los eventos celebrados dentro de esta feria. Con estos requerimientos podemos definir los pasos a seguir para la construcción del módulo ingreso de datos de nuestro programa.

1.1.Verificar fechas:

Teniendo en cuenta que las fechas de actividad de esta feria son fijas, creemos que el primer paso primordial para la solución es verificar que la compra de las entradas está dentro del rango de las fechas de celebración de la feria. Esto se va a lograr solicitando al usuario el ingreso del mes de la compra y el día específico para poder analizar el dato y dar un mensaje de alerta en caso que esta fecha se salga del rango permitido, tenemos pensado una serie de condicionales previos al cuerpo principal del programa.

Se trabajará en la siguiente estructura:

```
1| << "Dime para que día es la entrada:"
2| >> día
3| << "Dime para que mes es la entrada:"
3| >> mes
4| si (mes == "xxxxx")
5|     si (día == "1,2,3..."
6|         // Cuerpo principal //
7|     si no
8|         << "Mensaje de alerta"
```

Resultado final de la sección en código:

```
//Se solicita al usuario fecha y hora para comprobar que tipo de entrada se puede vender
System.out.print("Dime para que día es la entrada: ");
dia = teclado.nextInt();
teclado.nextLine();
System.out.print("Dime para que mes quieres la entrada: ");
mes = teclado.nextLine();
```

1.2.Verificar tipo de entrada:

Ya que en la Feria se venden dos tipos de entradas dependiendo del día, y que tipo de eventos se celebran, debemos verificar si el usuario desea comprar una entrada convencional o una entrada que le permite entrar a los eventos del día, para esto de igual forma se tiene pensado una estructura de condicionales para comprobar y registrar los datos, esto se va a verificar únicamente con una condición extra la cual va a ser la hora, ya que el día exacto ya se solicitó previamente, solo se debe verificar los horarios en caso la venta de la entrada de eventos sea obligatoria.

```

1 | Mientras seguir Hacer
2 |     Si mes es igual a "agosto" o mes es igual a "septiembre" Entonces
3 |         Segun dia Hacer
4 |             Caso 30, 31, 5, 6, 7:
5 |                 Escribir "Dime la hora a la que vas a ir: "
6 |                 Leer hora
7 |                 Si hora >= 5.00 y hora <= 23.59 Entonces
8 |                     var2 <- var2 + 1
9 |                 Sino
10 |                     var1 <- var1 + 1
11 |             FinSi
12 |         FinSegun
13 |     FinSi
14 | FinMientras

```

Resultado final de la sección en código:

```

while (seguir) {
    if (mes.equalsIgnoreCase("agosto") || mes.equalsIgnoreCase("septiembre")) {
        //Comprobar los días de eventos y la que no lo son
        switch (dia) {
            case 30: case 31: case 5: case 6: case 7: //Días de eventos
                //Comprobar la hora para determinar si es evento o no
                System.out.print("Dime la hora a la que vas ir(Recuerda que el horario se maneja de 24hr no 12hr por lo cual 1pm = 13pm): ");
                hora = teclado.nextFloat();
                //Comprobar la hora de la compra de la entrada, para determinar si son eventos o no
                if (hora >= 5.00 && hora <= 23.59) {
                    var2++;
                } else {
                    var1++;
                }
            }
        }
    }
}

```

2. Procesos

En base a la estructura principal de la problemática nosotros solo debemos usar un único proceso de calculo dentro del programa, el cual seria el cálculo de precios

según la entrada comprada y la cantidad de las mismas, generando diferentes precios para una correcta estructura estadística para facilitar la comprensión del usuario. Con esto en mente hemos decidido definir estos pasos en un procedimiento a parte para que este algoritmo se pueda reutilizar en otros bloques de código de ser necesario, mejorando así la eficiencia del programa.

2.1.Definición de los precios:

Primero para realizar el calculo debemos realizar la definición de los precios, de forma general, para esto investigamos en fuentes oficiales de la Feria internacional de Loja, para trabajar con precios que son utilizados en la celebración de la feria, los cuales son:

- Precio de entrada común: 2.50\$
- Precio de entrada para el ingreso a eventos: 5.00\$

2.2.Procesamiento de los datos:

Una vez obtenidos los valores generales de la entrada podemos continuar con el procesamiento de los datos, como sabemos el calculo de la venta de las entradas se hace según el día de la venta y la hora de manera automática, estos datos se guardan en una variable dentro del proceso, lo que quiere decir que para procesar los datos dentro de otro bloque de código estas variables deben ingresar como parámetros, por lo cual, el primer paso es declarar los parámetros a ingresar, una vez hecho esto debemos multiplicar el precio de las entradas comunes por la cantidad de entradas comunes vendidas y se realiza lo mismo con las entradas para el ingreso de eventos, teniendo la siguiente estructura:

```

1 | Procedimiento precio(var1, var2)
2 |     Definir precio_general, precio_eventos, total_general, total_eventos,
    precio_total Como Real
3 |     precio_general <- 2.50
4 |     precio_eventos <- 5.00
5 |     total_general <- 0
6 |     total_eventos <- 0
7 |     precio_total <- 0
8 |     total_general <- precio_general * var1
9 |     total_eventos <- precio_eventos * var2
10 |    precio_total <- total_general + total_eventos
11 |    Escribir "Precio por entradas generales: ", total_general, "$"
12 |    Escribir "Precio por entradas de eventos: ", total_eventos, "$"
13 |    Escribir "Precio total: ", precio_total, "$"
14 | FinProcedimiento

```

Resultado final de la sección en código:

```

public static void precio(int var1, int var2){
    //Declaracion de las variables temporales que se van a utilizar en este metodo
    double precio_general = 2.50, precio_eventos = 5.00, total_general = 0, total_eventos = 0, precio_total = 0;
    //Calculo del precio total de las entradas
    total_general = precio_general * var1;
    //Calculo del precio total de las entradas de eventos
    total_eventos = precio_eventos * var2;
    //Calculo del precio total de las entradas
    precio_total = total_general + total_eventos;
    //Impresion de los datos obtenidos
    System.out.println("Precio por entradas generales: " + total_general+"$");
    System.out.println("Precio por entradas de eventos: " + total_eventos+"$");
    System.out.println("Precio total: "+(precio_total)+"$");
}

```

Ya que trabajamos el procesamiento de datos en un procedimiento y no en una función es necesario generar la impresión de los datos dentro del mismo cuerpo del procedimiento.

3. Salida de datos

Una vez procesados los datos tenemos la necesidad de una impresión exacta de los datos y de fácil legibilidad para el usuario, teniendo en cuenta que también se solicita un enfoque estadístico en la impresión de los datos como una de las problemáticas a resolver, en base a estas solicitudes hemos decidido crear otro archivo de tipo java class para la impresión de los datos de forma estadística. Los datos que vamos a imprimir son:

- El día en el que se percibió mayor flujo de dinero
- El día en el que se percibió menor flujo de dinero
- Las ventas realizadas por día
- El total económico por la venta de todas las entradas
- El total económico por la venta de entradas generales
- El total económico por la venta de entradas para el ingreso de eventos.

3.1. Obtener datos extras:

Ya que debíamos tomar un enfoque estadístico nos vimos en la obligación de incluir variables que nos permitan verificar las ventas por día de la feria, por lo cual ya que solicitamos el día exacto de la venta podemos incluir 7 variables que coincidan con los días de la semana en los cuales la feria se encuentra activa, de esta forma incluimos una serie de condicionales para comprobar los datos y en caso de ser necesario guardarlos, la estructura se planea manejarla de la siguiente manera:

```
1 | Si dia == 5 Entonces
2 |   jueves <- jueves + 1
3 | Sino Si dia == 30 O dia == 6 Entonces
4 |   viernes <- viernes + 1
```


5 | Sino Si dia == 31 O dia == 7 Entonces

6 | sabado <- sabado + 1

7 | FinSi

8 |

9 | Si dia == 1 O dia == 8 Entonces

10 | domingo <- domingo + 1

11 | Sino Si dia == 2 Entonces

12 | lunes <- lunes + 1

13 | Sino Si dia == 3 Entonces

14 | martes <- martes + 1

15 | Sino Si dia == 4 Entonces

16 | miercoles <- miercoles + 1

17 | FinSi

Resultado final de la sección en código:

```
//Comprobar que dia exactamente se realizo la compra de las entradas, para generar estadística
if(dia == 5)
    jueves++;
else if(dia == 30 || dia == 6)
    viernes++;
else if(dia == 31 || dia == 7)
    sabado++;
```

```
//Comprobar el dia exacto durante los dias sin eventos
if(dia == 1 || dia == 8)
    domingo++;
else if(dia == 2)
    lunes++;
else if(dia == 3)
    martes++;
else if(dia == 4)
    miercoles++;
```

3.2.Impresión de los datos:

Una vez obtenidos los datos faltantes para la impresión de datos se va a proceder únicamente con la impresión de todos los datos obtenidos, con ayuda de los procedimientos antes realizados, así generando una mejor eficiencia al momento de la realización del código.

3.2.1. Calculo del día de mayor y menor ventas

Para el calculo de los valores máximos y mínimos vamos a ocupar una función de la categoría *Math.* de forma anidada, las funciones que vamos a utilizar son:

- *Math.max*: Permite calcular el valor máximo entre dos números.
- *Math.min*: Permite calcular el valor mínimo entre dos números.

Como mencionamos esto se va a utilizar de forma anidada ya que solo permite el ingreso de dos valores numéricos.

```
mayor = Math.max(var1, Math.max(var2, Math.max(var3, Math.max(var4, Math.max(var5, Math.max(var6, var7))))));  
menor = Math.min(var1, Math.min(var2, Math.min(var3, Math.min(var4, Math.min(var5, Math.min(var6, var7))))));
```

Una vez finalizada la obtención de datos secundarios, podemos finalmente imprimir los datos recaudados durante todo el proceso con la siguiente estructura:

1	Clase obtener_estadistica
2	Procedimiento estadistica(var1, var2, var3, var4, var5, var6, var7)
3	Definir mayor, menor, total_entradas, total_general, total_eventos Como Entero
4	mayor <- Max(var1, Max(var2, Max(var3, Max(var4, Max(var5, Max(var6, var7))))))
5	menor <- Min(var1, Min(var2, Min(var3, Min(var4, Min(var5, Min(var6, var7))))))
6	total_entradas <- var1 + var2 + var3 + var4 + var5 + var6 + var7

```

7 |      total_general <- var1 + var2 + var3 + var7
8 |      total_eventos <- var4 + var5 + var6
9 |      Escribir "Valor máximo de entradas vendidas por día: ", mayor
10 |     Escribir "Valor mínimo de entradas vendidas por día: ", menor
11 |     Escribir "+-----+-----+-----+-----+-----+-----+-----+
+"
12 |     Escribir "| LUNES | MARTES | MIERCOLES | JUEVES | VIERNES |
SABADO | DOMINGO |"
13 |     Escribir "+-----+-----+-----+-----+-----+-----+-----+
+"
14 |     Escribir "| ", var1, " | ", var2, " | ", var3, " | ", var4, " | ", var5, " | ",
var6, " | ", var7, " |"
15 |     Escribir "+-----+-----+-----+-----+-----+-----+-----+
+"
16 |     Escribir "Total de entradas vendidas: ", total_entradas
17 |     Escribir "Total de entradas vendidas por eventos: ", total_eventos
18 |     Escribir "Total de entradas vendidas en días sin eventos: ", total_general
19 |     FinProcedimiento
20 | FinClase

```

Resultado final de la sección en código:

```

public class obtener_estadistica {
    //Declaración del método que se va a encargar del cálculo y impresión de las estadísticas recaudadas
    public static void estadistica(int var1,int var2,int var3,int var4,int var5,int var6,int var7){
        int mayor=0, menor=0, total_entradas=0,total_general=0,total_eventos=0;
        //Funciones de tipo Math que van a iterarse repetitivamente de derecha a izquierda comprobando todos los valores para encontrar sus y min:
        mayor = Math.max(var1, Math.max(var2, Math.max(var3, Math.max(var4, Math.max(var5, Math.max(var6, var7))))));
        menor = Math.min(var1, Math.min(var2, Math.min(var3, Math.min(var4, Math.min(var5, Math.min(var6, var7))))));
        //Para encontrar el valor de todas las entradas vendidas independientemente del día se suma todas las variables
        total_entradas = (var1+var2+var3+var4+var5+var6+var7);
        //Para el total general solo se suman los días donde solo se pueden vender entradas generales
        total_general = (var1+var2+var3+var7);
        //Para el total eventos se realiza lo mismo pero solo los días de eventos
        total_eventos = (var4+var5+var6);
        System.out.println("Valor maximo de entradas vendidas por día: " + mayor);
        System.out.println("Valor minimo de entradas vendidas por día: " + menor);
        //Se imprime en una matriz los valores obtenidos en cada día de la semana, esto se realiza en el método compra del java class compra_precio_entradas
        System.out.println("
");
        System.out.println("
+-----+-----+-----+-----+-----+-----+
| LUNES | MARTES | MIERCOLES | JUEVES | VIERNES | SABADO | DOMINGO |
+-----+-----+-----+-----+-----+-----+
");
        System.out.print("
| "+var1+" | "+var2+" | "+var3+" | "+var4+" | "+var5+" | "+var6+" | "+var7+" |
");
        System.out.print("
+-----+-----+-----+-----+-----+-----+
");
        System.out.println("Total de entradas vendidas: " + total_entradas);
        System.out.println("Total de entradas vendidas por eventos: " + total_eventos);
        System.out.println("Total de entradas vendidas en días sin eventos: " + total_general);
    }
}

```

Con esto se finaliza los requerimientos del programa, de manera eficiente, y usando una correcta programación modular para la obtención y reciclamiento de código repetitivo.