



Sede Regional Brunca

Campus Pérez Zeledón

Sistema de ventas y creación de un evento

Brenda Aguilar Fonseca y Erick Fonseca Mata

Primer proyecto - Programación I

Ingeniería en Sistemas

Universidad Nacional

PROF. MSC. Jeison Vargas Miranda

15 de Enero 2025

Tabla de Contenidos

Descripción del Proyecto	3
Clase Segmento.....	3
Clase Descuento.....	4
Clase Cliente.....	4
Clase Evento.....	5

Descripción del proyecto:

Para el desarrollo de este proyecto en c++ se utilizó la POO, por lo que se crearon 5 clases y en ellas se utilizó abstracción y encapsulamiento, cada una de las clases recibirá su respectiva explicación para abarcar todas sus funciones y su uso.

Clase Segmento

Esta clase brinda las bases para el proyecto, puesto a que en esta se crea el “molde” de todo lo que debe tener el segmento, como la cantidad de espacios, el precio, etc. Sus principales funciones son las siguientes:

inicializarMatriz(): En esta función se crea la matriz de espacio que se utilizará en cada segmento, y estos se inicializan en “D” para indicar que no se ha comprado ninguno hasta el momento.

preguntarDatos(): En esta se preguntan los datos fundamentales para la creación del segmento como la cantidad de filas y columnas, y el precio, luego se llama la función de **inicializarMatriz()** para usar los datos ingresados por el usuario.

verificarEstadodeEntradas(): La Función principal de este método es saber si dicho segmento cuenta aún con entradas disponibles para su venta, si no cuenta con entradas entonces no se pueden realizar más compras en dicho segmento.

seleccionarEspacio(): La principal funcionalidad de este método es que el usuario digite la letra (que se tiene que pasar a número para buscar el espacio en la matriz) de la fila y el número de columna que desea comprar y marcarla como “V” (vendida), además esta tiene algunas restricciones para asegurarse de que el espacio seleccionado exista y esté disponible.

mostrarEspacios(): Imprime en pantalla el estado de los asientos, “V” para los vendidos y “D” para los disponibles.

```
class Segmento {
private:
    int fila;
    int columna;
    char** espacios;
    float precio;
    int entradasVendidas;
    int entradasPorUsuario;

public:
    Segmento();
    ~Segmento();

    void setPrecio(float precio);
    void setFila(int fila);
    void setColumna(int columna);
    int getFila();
    int getColumna();
    float getPrecio();
    int getEntradasVendidas();
    int getEntradasPorUsuario();
    void inicializarEntradasPorUsuario();

    void inicializarMatriz();
    void preguntarDatos();
    bool verificarEstadodeEntradas();
    void seleccionarEspacio();
    void mostrarEspacios();
};
```

Clase Descuento

Su creación cumple con el objetivo de aplicar descuentos en las entradas compradas por los usuarios, con una contraseña para una mayor seguridad en cada compra que el usuario quiera aplicar un descuento.

Entre sus principales métodos y atributos para el cumplimiento de los objetivos planteados en la creación del descuento son:

Se crea el atributo “**string* vecContrasenia**”, para guardar las contraseñas de forma dinámica en cada cupón que se digite para aplicar el descuento, esto se cumple en el método “**getContrasenia**” donde se va incrementando para el uso de todas las contraseñas que tiene cada cupón.

Otro método importante es “**generarContrasenia**”, en ella se crean 3 for para contar con el alfabeto de forma minúscula y mayúscula, el tercero contará con el manejo de los números, también se manejan caracteres especiales, todo esto se combina para crear una contraseña usando un random que tomará diferentes caracteres y números para la creación de dicha contraseña.

Para aplicar el descuento se cuenta con dos métodos los cuales uno para confirmar el descuento y otro para aplicar dicho descuento. El primer método para confirmar el descuento se trabaja de forma parametrizable tal cual se pedía en el enunciado del proyecto, en él se asignará el precio que contiene el segmento seleccionado en el también se llama el método que aplica el descuento. El método para aplicar el descuento trabaja con el porcentaje y la cantidad de boletos que se han comprado, sumará todos los boletos comprados para obtener el total a pagar y les aplicará su respectivo descuento con el porcentaje que se está trabajando en dicho momento.

```
class Descuento
{
private:
    int porcentaje;
    int cantidad;
    string contrasenia;
    float precioActual;

    string* vecContrasenia;
    int contadorContrasenia = 0;

public:
    Descuento();

    void setCantidad(int cantidad);
    void setContrasenia(string contrasenia);

    int getCantidad();
    string getContrasenia(bool estadoContrasenia);

    string generarContrasenia();
    void confirmarDescuento(float precioSegmento);
    float aplicarDescuento();
    void IngresarDatosdelEvento();
};
```

Clase Cliente

Esta clase se encarga de preguntar los datos personales del cliente como el nombre, número de cédula y fecha de nacimiento.

```
class Cliente
{
private:
    string nombreCliente;
    string cedulaCliente;
    string fechaNacimiento;
    int dia, mes, anio;

public:
    string getNombreCliente();
    string getCedulaCliente();
    string getFechaNacimiento();
    void preguntarDatos();
};
```

Clase Evento

Clase principal, donde se combinan las demás clases para cumplir todos los objetivos propuestos en el enunciado del proyecto. En esta clase se trabajan los métodos tanto para la configuración del evento como para la venta de las entradas, los métodos principales para cumplir este objetivo son:

configurarEvento(): En este método se le pregunta al usuario los datos fundamentales para la creación del evento como el nombre y la cantidad de segmentos que tendrá. Teniendo estos datos se crea un arreglo dinámico de segmentos y luego se preguntan los datos de cada uno.

venderEntradas(): Es la función encargada de unir todos los datos que se necesitan para la venta de las entradas.

gestionarCompra(): La función de este método es manejar el control de la venta de los segmentos en ella se puede verificar la disponibilidad de los segmentos(cuántos espacios están disponibles llamando la función de **seleccionarSegmento**), también cuenta con la restricción de las 5 entradas para la venta, además se llama el método de **imprimirEstado de Venta** para tener en cuenta que entradas están disponibles o saber si algún segmento ya no cuenta con entradas. Una vez la persona ya decide no comprar más se hace una condicional para saber si sus entradas cuentan con descuento(esteste proceso se hace en el método **procesarDescuento**, una vez se le aplique o no el descuento se imprimirá la factura de su compra.

seleccionarSegmento(): En este método se valida la elección del segmento en el cual se va a guardar el segmento seleccionado para poder aplicar la compra, también se

realiza una condicional donde se llama el método de **verificarEstadoEntradas**, para saber si ese segmento ya se encuentra lleno o no, si se encuentra lleno existe la posibilidad de elegir otro, en el caso que sea solo uno y este lleno no se realizará más ventas y el comprador se le preguntará si quiere aplicar descuento a los boletos que ya había comprado previamente.

procesarDescuento(): El método cumple con la función de consultar al comprador si desea aplicar descuento a sus entradas, para esto se declara una variable llamada **contrasenna** para que se guarde la contraseña y la pueda mostrar para que el usuario pueda escribirla y aplicar su debido descuento, en este método se utilizan dos variables booleanas una llamada **contraseniaAceptada** su función es reconocer la contraseña ingresada, para ver si ingresa una contraseña correcta, la otra variable llamada **contraseniaCondicion** se crea para que se pueda usar una contraseña de cada cupón cada vez que el usuario quiera aplicar el descuento, es decir que pase a la siguiente contraseña del siguiente cupón que se quiera digitar. Los descuento se mostrarán al final en su factura.

imprimirInformacionEvento(): Se muestra un resumen de toda la información del evento creado, tales como el nombre, cantidad de segmentos, la cantidad de espacios que tiene cada segmento y su respectivo precio.

generarFactura(): Aquí se muestra toda la información de la compra realizada, un desglose de las entradas con su respectivos precios y el total con y sin descuento aplicado.

imprimirEstadoDeVentas(): Este método se utiliza para dar una representación gráfica de las entradas que están vendidas y las que no. Este se utiliza en el caso de que el usuario desee seguir comprando, con la finalidad de que pueda buscar bien los asientos. También se puede revisar el estado de ventas con la misma.

infoEstudiantes(): Muestra la información de los estudiantes que realizaron el programa.

mostrarLogo(): Muestra el logo de la empresa de eventos creada.

menu(): Fundamental para el proyecto, permite que el usuario decida la opción del programa que desea utilizar.

```
class Evento
{
private:

    Segmento* segmentos;
    int numeroSegmento;
    string nombreEvento;
    Cliente cliente;
    Descuento descuento;
    int cantidadPersonas;
    int segmentoSeleccionado;
    bool descuentoAceptado = false;
    bool noHaySegmentos = false;
    bool contraseniaCondicion = true;

public:

    Evento();
    ~Evento();

    string getNombreEvento();

    void configurarEvento();
    void venderEntradas();
    void gestionarCompra();
    int seleccionarSegmento();
    bool procesarDescuento();
    void imprimirInformacionEvento();
    void generarFactura();
    void imprimirEstadoDeVentas();
    void infoEstudiantes();
    void mostrarLogo();
    void menu();
};
```