



Sede Regional Brunca

Campus Pérez Zeledón

Proyecto #1

Curso: Programación I

Profesor: Josías Chaves M.

III Ciclo - Grupo 80

Estudiantes:

Abarca Pérez Jefferson David

Granados Vargas Adrián

15 de Enero del 2025

Tabla de contenidos

Tabla de contenidos. 2

Planteamiento del problema: 3

Solución adaptada: 4

Diseño: 4

- ☐ Interfaz del sistema: 4
- ☐ Gestión de datos:5
- ☐ Validaciones:5

Funcionalidades y estructuras en c ++: 6

Clases: 6

- ☐ Client: 6
- ☐ ConfingEvent: 6
- ☐ DateProcessorTool:6
- ☐ Segment: 7

Anexos:8

Referencias Bibliográficas: 11

Planteamiento del Problema

La organización de eventos masivos es un proceso complejo que requiere precisión, coordinación y herramientas tecnológicas efectivas. Sin embargo, en la actualidad, tanto organizadores como los usuarios finales enfrentan desafíos significativos que dificultan la ejecución de una manera correcta y a su vez que se disfrute de estos eventos. Los problemas más comunes incluyen la falta de control sobre la asignación de asientos, errores en el proceso de venta de entradas y la imposibilidad de personalizar eventos según las necesidades específicas de los organizadores.

Para los organizadores, estas limitaciones representan barreras que les impiden diseñar eventos que se alineen con su visión de una manera fácil y sencilla. La ausencia de un sistema flexible y adaptable limita la creación de sectores diferenciados, la asignación de precios variados y la implementación de estrategias de descuentos. Estas carencias dificultan la planificación eficiente, generando desorganización y afectando la calidad del evento en sí.

Desde la perspectiva de los usuarios finales, los problemas se reflejan en la falta de claridad y accesibilidad en el proceso de la compra de entradas para estos eventos. Según datos proporcionados por la Organización de Consumidores y Usuarios de Chile (ODECU), muchos usuarios enfrentan incertidumbre sobre cómo adquirir boletos y seleccionar asientos, lo que limita su acceso a eventos de interés y disminuye su satisfacción. Este escenario genera una desconexión entre las expectativas de los asistentes y la realidad operativa de los eventos (ODECU, 2019).

Ante estas dificultades, surge la necesidad de un sistema tecnológico innovador que centralice la configuración, gestión y venta de boletos para eventos masivos. Este sistema debe permitir a los organizadores personalizar cada aspecto del evento, desde la creación de sectores hasta la configuración de precios y descuentos, mientras que a los usuarios les ofrece un entorno intuitivo para seleccionar asientos y realizar compras de manera sencilla y confiable. La implementación de esta solución no solo mejoraría la experiencia del usuario, sino que también optimizaría los procesos organizativos, garantizando eventos más eficientes y satisfactorios para todas las partes involucradas.

Solución adoptada:

De acuerdo con lo mencionado anteriormente, para la solución del problema presente el cual se basa en crear un programa en C++ donde se permita gestionar eventos masivos tanto como a los usuarios como a los diseñadores. Este programa tiene como objetivo registrar y organizar a los participantes en los eventos masivos, así como generar informes y estadísticas útiles para los organizadores. La solución se estructura en componentes claramente definidos para garantizar la escalabilidad, la eficiencia y la claridad del código.

Diseño:

El diseño del sistema se basa en una arquitectura la cual es capaz de organizar las funcionalidades principales del programa en distintas clases y métodos. Esto permite que el desarrollo sea más comprensible y que las futuras modificaciones o ampliaciones puedan implementarse de manera aún más eficiente, esto para facilitar la organización de los eventos masivos y la realización del mismo.

Algunos elementos necesarios para la elaboración del programa son:

Interfaz del sistema:

La interacción del programa con el organizador o usuario se realiza a través de un menú en consola que ofrece las siguientes funcionalidades:

1. Configurar/Crear evento.
2. Configurar descuento.
3. Comprar entradas.
4. Observar disponibilidad de asientos.
5. Acerca de.
6. Salir del programa.

Este menú guía al usuario a través de las diferentes opciones de la manera lo más intuitiva posible hacia otros menús los cuales le permiten hacer la gestión de lo que es realizar el eventos, esto mediante mensajes claros que facilitan el uso del mismo para todo tipo de usuario, este menú está basado en un boceto el cual fue proporcionado por Josías Chaves(J.Chaves, 2025).Las funcionalidades de los menús a los cuales derivan este son:

1. Al escoger la opción número uno de configuración del evento deriva en el un menú el cual permite crear un evento, ver la información del evento actual y finalizar el mismo.
2. En la opción de configurar descuento permite al usuario obtener un código de descuento si este lo desea o necesita, esto también depende de la disponibilidad de los mismos, ya que son limitados a gusto del organizador.
3. Al momento de seleccionar la opción número tres el usuario accede al apartado donde puede comprar una entrada, a su vez que puede observar la disponibilidad de los asientos a tiempo real.
4. La opción cuatro está únicamente destinada a observar lo que es la disponibilidad de todos los segmentos del evento, a tiempo real.
5. Opción que deriva únicamente en conocer la información de los creadores del programa.
6. Para salir del programa y finalizar el evento.

Gestión de datos

El manejo de datos es uno de los pilares fundamentales del sistema, el cual se organiza en las siguientes áreas:

1. Registros de participantes : Cada participante se representa como un objeto con atributos clave tales como el nombre, fecha de nacimiento y cédula de identificación y las validaciones necesarias para cada uno.
2. División de los segmentos: Cada evento permite ingresar x cantidad de segmentos los cuales se pueden diseñar y modificar lo que es el tamaño de cada uno y los precios de cada asiento en estos segmentos.
3. Estadísticas de eventos : El sistema muestra datos relevantes como lo son los asientos que están vendidos y los que no, también el desglose de lo que es cada venta con información detallada en un recibo.
4. Informes : Se generan listados detallados en formato legible, organizados por segmentos y por venta para la transparencia entre la información, los organizadores y los usuarios.

Validaciones

Para garantizar que no se pierdan nada de los datos, el programa implementa validaciones en todas las entradas del usuario. Por ejemplo:

- Verificación de que los cupos en el evento no estén ocupados antes de registrar nuevos participantes.
- Validación de formatos de entrada como nombres, fechas de nacimiento o cédula de identidad para evitar inconsistencias o problemas en los registros.
- Validación al definir la fecha del evento, que no se pueda realizar en fechas inválidas tomando en cuenta el año bisiesto.
- Validación para que elementos como códigos de descuento no se puedan utilizar más de una vez.
- Validaciones para no poder ingresar elementos en blanco.

Funcionalidades y estructuras en c++

En esta etapa al momento de aplicar las funcionalidades en el lenguaje madre de este proyecto utilizando las librerías `iostream`, `string`, `omanip` y `ctime` para poder realizar el programa correctamente. algunos puntos importantes a tomar en cuenta en el uso de este lenguaje para realizar este gestor de eventos masivos son:

Clases:

Las clases como permiten estructurarse de una manera más exacta se crearon con el fin de hacer objetos diferentes y sus diferentes funcionalidades para poder realizar las validaciones y operaciones necesarias .

Algunas de las clases las cuales fueron creadas son:

- ❖ **Client:** Esta clase permite gestionar la información personal de cada client, como la identificación, el nombre y la fecha de nacimiento, asegurando que los datos ingresados sean válidos y únicos. Además, maneja las reservas de asientos realizadas por el cliente, llevando un registro de los mismos, y calcula información relacionada con las facturas, incluyendo descuentos aplicables y el monto total.
- ❖ **ConfinEvent:** En esta clase se captura y valida la información básica del evento como nombre, fecha, número de segmentos y datos relacionados con los asientos para enfrentar un error en específico como lo es el ingreso de números decimales se utilizó el método `static_cast<int>` para poder redondear dichos números (IBM, 2024) , también se genera y administra códigos de descuento para las entradas, muestra el estado actual de los segmentos y asientos a su vez permite reservar asientos y verificar su disponibilidad, y también realiza funcionalidades como la compra de boletos, aplicando descuentos si se utilizan códigos válidos, también genera facturas detalladas que incluyen información del evento, del cliente, precios totales y descuentos aplicados utilizando arreglos dinámicos para manejar los segmentos y asientos, ajustando el espacio según sea necesario.
- ❖ **DateProcessorTool:** Esta clase está diseñada para ser una herramienta auxiliar en sistemas donde la gestión de fechas es crucial, como en el control de eventos, validación de datos personales o registro de usuarios. Proporcionar robustez al garantizar que las fechas sean válidas, estén en el formato adecuado y cumplan con las reglas específicas del contexto en que se utilizan.
- ❖ **EventController:** Se encarga de gestionar las distintas opciones de menú para configurar y administrar un evento, como la compra de entradas, la visualización de disponibilidad de asientos, y la configuración de descuentos, entre otros. El propósito general de esta clase es servir como un controlador central para gestionar todas las operaciones relacionadas con un evento, desde su creación y configuración hasta la venta de entradas, brindando un flujo de trabajo claro y organizado para el usuario.

- ❖ Segment: Representa un segmento de asientos dentro de un evento o espacio físico, ejemplificado como un evento masivo. Su propósito es gestionar la asignación, disponibilidad y visualización de los asientos, así como mantener información relevante sobre el segmento, como su precio y dimensiones.

Para realizar algunas de estas funcionalidades específicas se tomaron elementos necesarios de fuentes confiables como lo son el `.ignore` (geeksforgeeks, 2024) para ignorar la entrada anterior información dentro de variables, también para poder pausar la pantalla en momentos específicos se utilizó el `system("pause")` y el `system("cls")` (DigitalOcean, 2022) sin dejar de darle importancia al `.empty()` para ignorar los espacios en blanco o vacíos (Lern, 2023).

Anexos:

MENU PRINCIPAL

1. Configurar Evento.
2. CrearCodigo de Descuento.
3. Comprar Entrada.
4. Observar Disponibilidad de Asientos.
5. Acerca de.
6. Finalizar.

Escoja una opcion:

CONFIGURACION DEL EVENTO

1. Crear un nuevo evento.
2. Ver evento actual.
3. Finalizar evento.
4. Volver.

Escoja una opcion: |

INFORMACION DEL EVENTO:

Nombre del evento		Grupo Firme
Fecha del evento		29/09/2025
Numero de segmentos		3
Numero de asientos		75

Presione una tecla para continuar . . . |

Segmento 1:

[_A1_]	[_A2_]	[_A3_]	[_A4_]	[_A5_]
[_B1_]	[_B2_]	[_B3_]	[_B4_]	[_B5_]
[_C1_]	[_C2_]	[_C3_]	[_C4_]	[_C5_]
[_D1_]	[_D2_]	[_D3_]	[_D4_]	[_D5_]
[_E1_]	[_E2_]	[_E3_]	[_E4_]	[_E5_]

Segmento 2:

[_A1_]	[_A2_]	[_A3_]	[_A4_]	[_A5_]
[_B1_]	[_B2_]	[_B3_]	[_B4_]	[_B5_]
[_C1_]	[_C2_]	[_C3_]	[_C4_]	[_C5_]
[_D1_]	[_D2_]	[_D3_]	[_D4_]	[_D5_]
[_E1_]	[_E2_]	[_E3_]	[_E4_]	[_E5_]

Segmento 3:

[_A1_]	[_A2_]	[_A3_]	[_A4_]	[_A5_]
[_B1_]	[_B2_]	[_B3_]	[_B4_]	[_B5_]
[_C1_]	[_C2_]	[_C3_]	[_C4_]	[_C5_]
[_D1_]	[_D2_]	[_D3_]	[_D4_]	[_D5_]
[_E1_]	[_E2_]	[_E3_]	[_E4_]	[_E5_]

Presione una tecla para continuar . . . |

Factura

Grupo Firme

Fecha del evento: 29/09/2025

Informacion personal.

Nombre del Usuario: Jefferson David Abarca Pérez

Numero de Identificacion:

Fecha de nacimiento: 22/08/2004

Informacion de compra.

Cantidad de Asientos adquiridos: 2

Precio total (sin descuento): 10.00

Cantidad del descuento aplicado: 0.00

Precio total a cancelar: 10.00

Gracias por adquirir cu entrada
con nosotros esperamos su
disfrute en este Evento

Presione una tecla para continuar . . .

Referencias Bibliográficas

ODECU. (2019). *Problemas en la compra de entradas para eventos y espectáculos masivos*.
<https://www.odecu.cl/2019/02/05/pregunta-frecuente-problemas-en-la-compra-de-entradas-para-eventos-y-espectaculos-masivos>

J.Chaves y J.Vargas. (2025). Descripción del proyecto #1
https://www.aulavirtual.una.ac.cr/pluginfile.php/82358/mod_resource/content/1/2.%20Proyecto%20PROGRA%20I%202024.pdf

IBM. (2025). El operador static_cast (sólo C++).
<https://www.ibm.com/docs/es/i/7.5?topic=expressions-static-cast-operator-c-only>

GEEKSFORGEEKS.(2024). Función cin.ignore() en C++.
<https://www.geeksforgeeks.org/cin-ignore-function-in-cpp/>

CPLUSPLUS.(2013). Programa en C++ que ingresa dos números enteros.
<https://cplusplus.com/forum/beginner/108090/>

DigitalOcean. (2022). Uso del comando system("pause") en C++.
<https://www.digitalocean.com/community/tutorials/system-pause-command-c-plus-plus>

Lern.(2023).Uso de las funciones STL stack::top y stack::empty en Visual C++
<https://learn.microsoft.com/es-es/troubleshoot/developer/visualstudio/cpp/libraries/stack-top-empty-stl-function>