



Sede Regional Brunca

Campus Pérez Zeledón

Proyecto #2

Curso: Programación I

Profesor: Josías Chaves M.

III Ciclo - Grupo 80

Estudiantes:

Abarca Pérez Jefferson David

Granados Vargas Adrián

12 de Febrero del 2025

Tabla de contenidos.....	2
1. Planteamiento del Problema.....	3
2. Solución Adoptada.....	4
3. Diseño del Sistema.....	5
○ Interfaz del Sistema.....	5
○ Gestión de Datos.....	6
○ Validaciones.....	6
4. Funcionalidades y Estructuras en C++.....	7
○ Clases.....	7
5. Anexos.....	8
6. Referencias Bibliográficas.....	10

Planteamiento del Problema

El desarrollo y organización de eventos masivos requiere diversas herramientas tecnológicas para poder optimizar y mejorar la asignación de asientos, la gestión de boletos y también la experiencia del usuario final. En la primera versión del proyecto, se abordaron los problemas fundamentales tales como la compra de boletos y asignación de asientos. Sin embargo, se identificaron nuevas necesidades y algunos problemas como lo fueron la integración de una interfaz gráfica más intuitiva, validaciones más estrictas y un mejor manejo de reportes y segmentación de eventos.

Como se muestra a través de perspectiva de los usuarios finales de los datos proporcionados por la ODECU, los problemas se reflejan en la falta de claridad e accesibilidad en todo lo que conlleva el proceso de la compra de entradas para estos eventos y también a la hora de diseñar el mismo.

Los proporcionados por la Organización de Consumidores y Usuarios de Chile (ODECU), indica que muchos usuarios enfrentan trabas al 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).

Para abordar estos retos, el proyecto se expande de una manera que permita integrar una interfaz basada en SFML que permita una experiencia visual interactiva en la selección de asientos y la administración de eventos.

Solución Adoptada

Para resolver los problemas mencionados, se ha desarrollado un programa en C++ utilizando SFML para la interfaz gráfica. Las principales mejoras implementadas incluyen:

- **Interfaz visual interactiva:** Los asientos se representan como botones en una ventana gráfica de SFML en lugar de utilizar una interfaz de consola.
- **Validaciones mejoradas:** Se han implementado verificaciones estrictas para la cédula de los clientes, selección de asientos y fechas de eventos.
- **Generación de reportes automatizada:** Se generan archivos de texto con detalles de ventas, información del cliente y disponibilidad de asientos.
- **Aplicación de descuentos:** Se gestionan códigos de descuento con validación y aplicación automática.
- **Redimensión de los segmentos:** El programa permite añadir y eliminar filas o columnas a petición del desarrollador.
- **Cargar datos:** Se guardan los datos del evento en una lista, la cual guarda a su vez tiene la lista de los segmentos y esta a su vez tiene la información específica de la misma.

Diseño

Interfaz del sistema

La parte más importante de este proyecto fue diseñar la interfaz de este sistema la cual ha sido creada de una manera muy intuitiva para que funcione y pueda proporcionar una experiencia confortable y principalmente fácil de entender al usuario final de la manera más eficiente posible.

Esto se pudo lograr gracias a que se utilizó la librería SFML, desarrollado una interfaz interactiva con un tema en específico escogido estratégicamente el cual que permite navegar por las distintas opciones del programa a través de botones interactivos.

Las opciones principales del sistema incluyen:

1. Configurar evento: Permite ingresar datos como nombre, fecha y cantidad de segmentos, además de visualizar la configuración actual.
2. Configurar Código: Genera códigos de descuento y muestra los disponibles para su uso.
3. Comprar entradas: Facilita la selección de asientos a través de botones interactivos en una representación gráfica del recinto.
4. Disponibilidad: Muestra en tiempo real los asientos reservados y disponibles dentro del evento.
5. Rembolsar entrada: La cual permite al diseñador del evento eliminar una venta en específico y esta se elimina del archivo de texto.
6. Cargar evento: Carga un evento totalmente funcional el cual se tuvo que almacenar previamente en el archivo de texto.
7. Editar Evento: Permite modificar un segmento es específico, eliminarlo o también permite eliminar algún código de descuento.
8. Acerca de: Muestra una foto de los diseñadores del programa.
9. Salir del programa: Cierra la aplicación de forma segura.

Gestión de Datos

El sistema organiza la información utilizando estructuras de datos eficientes para garantizar un rendimiento óptimo. Las principales áreas de gestión de datos incluyen:

1. **Lista de clientes:** Se mantiene un registro bien detallado de los clientes, incluyendo su identificación, nombre y las entradas compradas.
2. **Segmentación de asientos:** Cada evento está dividido en segmentos que son totalmente configurables, donde cada segmento tiene una cantidad específica de filas, columnas y un precio por asiento, estos mismos son modificables en tamaño.
3. **Historial de ventas y descuentos:** Se lleva un seguimiento total de las transacciones realizadas, incluyendo tanto como la aplicación de códigos de descuento y las reservas de los asientos.
4. **Archivos de reportes:** Se genera un informe detallado sobre la ocupación del evento y las ventas realizadas del mismo, almacenados en archivos de texto para su análisis y posible utilización en un futuro.

Validaciones

El sistema implementa diversa cantidad validaciones para asegurar lo que es integridad y consistencia de los datos dados por los usuarios. Entre las validaciones más importantes se encuentran:

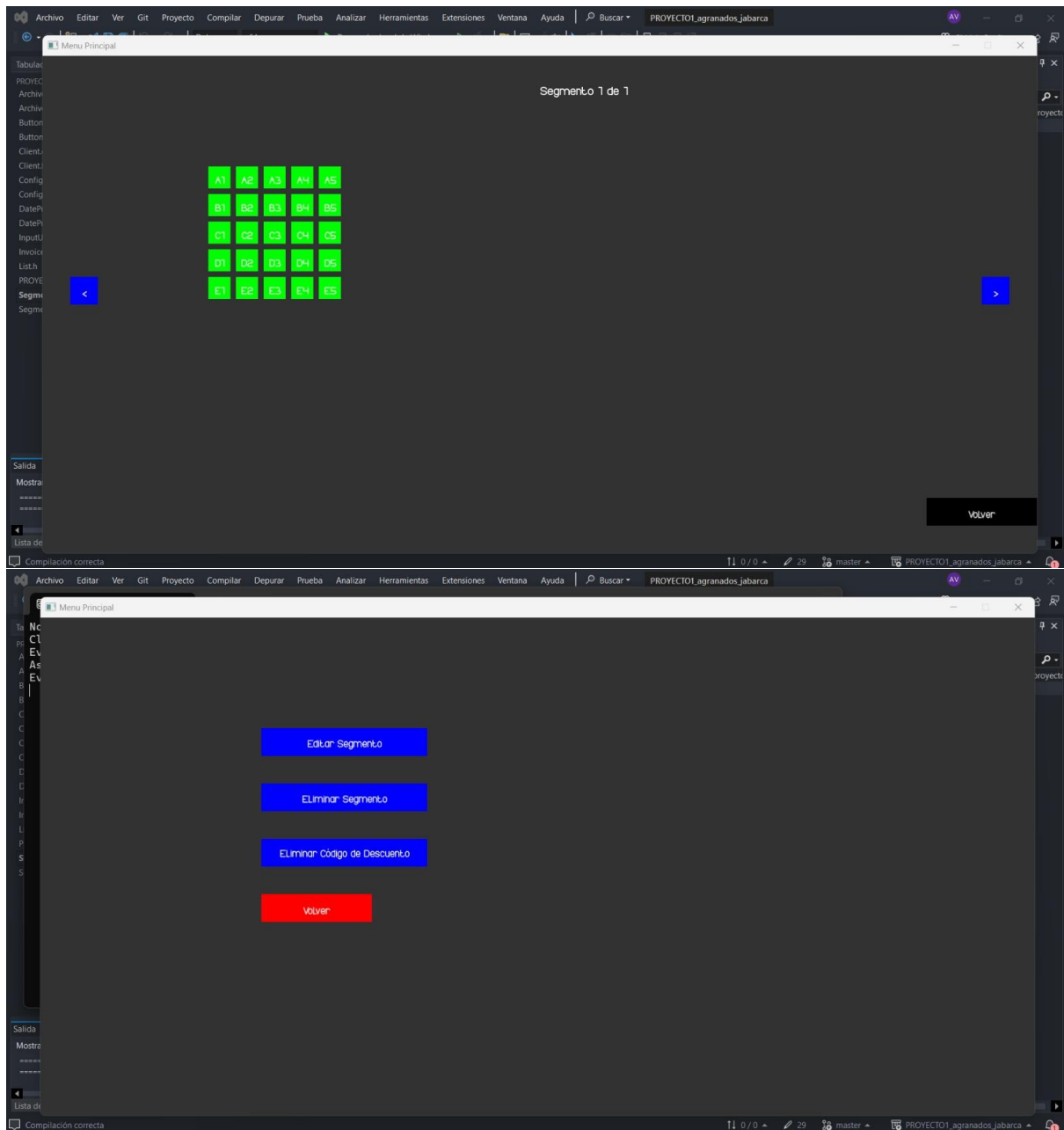
- **Verificación de cédula:** Se asegura que cada cédula ingresada tenga el formato correcto y no se pueda ingresar 2 veces con la misma cédula en el sistema.
- **Restricción de asientos ocupados:** Un asiento ya reservado no va a poder ser seleccionado nuevamente.
- **Validación de fechas de eventos:** Se verifica que los eventos solo puedan programarse únicamente para en fechas futuras y con el formato adecuado, también verifica que los clientes sean mayores de edad para adquirir entradas.
- **Aplicación de códigos de descuento:** Cada código de descuento solo puede ser utilizado una vez y también este debe estar dentro del rango de descuentos permitidos.

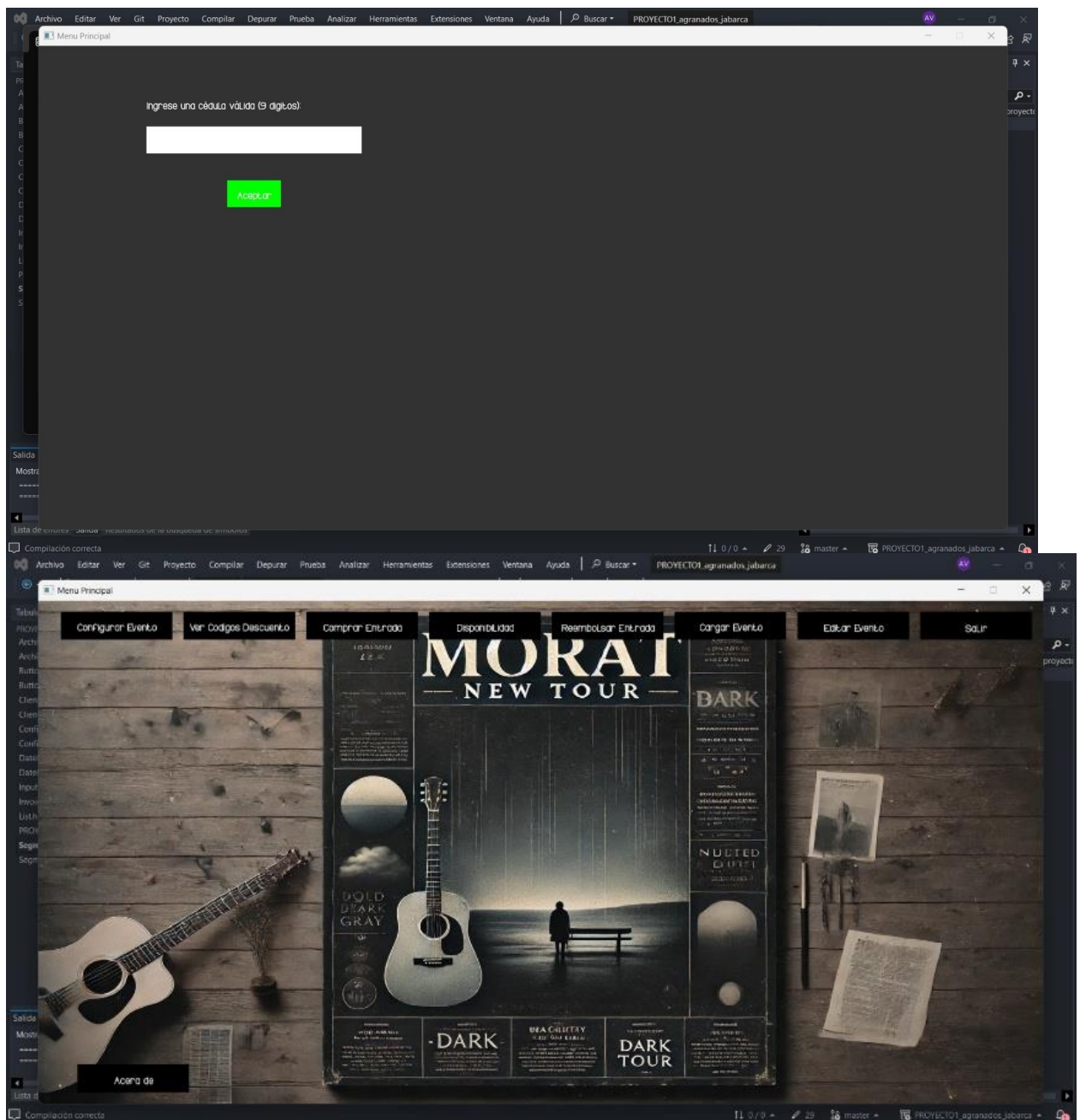
Funcionalidades y Estructuras en C++

Clases

- **Client:** Administra la información del cliente, validando su cédula y registrando sus compras. También maneja la aplicación de descuentos y el cálculo del total a pagar.
- **ConfigEvent:** Permite configurar eventos, estableciendo su nombre, la fecha, sus segmentos, precios, descuentos y demás funciones.
- **DateProcessorTool:** Gestiona la validación de fechas, asegurando que los eventos se programen en fechas futuras y verificando la mayoría de edad de los clientes y también el formato correcto de las mismas. .
- **Segment:** Representa un segmento dentro del evento, administrando la disponibilidad de asientos y su visualización en la interfaz gráfica específicamente con algunas funciones para inicializar, dibujar, reservar, liberar y verificar el estado de los asientos.
- **Button:** Implementa la representación gráfica de botones interactivos en SFML, permitiendo la navegación en el sistema y la selección de asientos.
- **ReportViewer:** Genera reportes detallados sobre la venta de entradas y la ocupación de asientos dentro del evento para luego mostrarlos.
- **List:** Es una lista simplemente enlazada la cual permite almacenar elementos de cualquier tipo, y proporciona métodos para agregar, eliminar, buscar y acceder a los elementos de la lista.
- **InputUtils:** Permite que el usuario al ingresar un valor alfanumérico mediante la interfaz de sfml en consola.

Anexos:





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>