

## **Definición general**

En el siguiente documento se presenta una propuesta de desarrollo para un sistema de venta de boletos.

El sistema debe permitir el registro de usuarios, donde podrán proporcionar la información necesaria para crear una cuenta. Una vez registrados, los usuarios podrán iniciar sesión en el sistema para comprar boletos, administrar su información personal, así como acceder a su historial de compras. Además, el sistema les debe permitir obtener información sobre los próximos eventos.

Una política organizacional indica que una vez adquirido un boleto, el usuario tendrá la capacidad de cancelar o modificar sus compras con un límite máximo de 48 horas.

## **Desarrollo**

El sistema fue desarrollado en las siguientes condiciones

- SO debian 12
- Lenguaje de programación: C++ version 12.2.0
- IDE Sublime Text

## **Planificación**

El desarrollo se dividió en cuatro subproblemas:

- USUARIO
- BOLETO
- EVENTO
- SISTEMA

En usuario se determinan las siguientes características:

Un usuario tiene nombre, contraseña y un historial de compras; el cual se puede mostrar, puede cancelar una compra, y puede modificar una compra siempre y cuando cumpla con la condición de no exceder los dos días de tolerancia.

En boleto tenemos:

Cada boleto tiene un ID, un número de asiento, el evento al que pertenece y una fecha de compra; un boleto solo puede mostrar y editar cada una de esas características.

Un evento se caracteriza por:

Contiene un ID, un lugar donde sucederá, el total de asientos, el nombre del evento, la fecha en que sucederá. En un evento se pueden comprar boletos, liberar un asiento, editar un asiento si el usuario decidió cambiar de lugar, mostrar los asientos disponibles, las características mismas del evento, y además editarlas en caso de ser necesario.

El sistema, siendo la interfaz con la que se comunicara el usuario, se define por:

Un menú de acceso, para que pueda ingresar el administrador o el usuario:

El administrador puede crear un evento, crear un usuario, mostrar los eventos, mostrar los usuarios.

Un usuario puede ver su historial, ver los eventos disponibles, comprar un boleto, editar alguna compra, eliminar una compra, editar su perfil, o cerrar sesión.

Además, se generó una clase especial para la fecha, que contiene día, mes, año y un comparador de fecha, para poder identificar si la fecha excede la tolerancia según las políticas de administración.

## **Justificación**

Al ser un problema complejo en cuanto a elementos, se decidió separar en esas cuatro características debido a que implicaría mejor facilidad para manejar el problema. También, al dividirlo en usuario, boleto, sistema y evento, es más sencillo agregar funcionalidades y nuevas características a cada uno de los elementos por separado.

En cuanto al lenguaje de programación, se escogió c++ debido a que es un lenguaje orientado a objetos, y que manejamos directamente con memoria y apuntadores, por lo cual, el editar los elementos que se contienen dentro de otros, por ejemplo, un boleto tiene un evento, y un usuario tiene un boleto, es más eficaz manejar

directamente el contenido de cada objeto, y así evitar copias de información innecesarias y simplificar el acceso a la información.

Con ello, realizamos un programa más eficiente para sistemas ligeros y también, debido a las características del lenguaje de programación, lo hacemos más seguro al darnos la libertad de únicamente darle al usuario el archivo ejecutable sin darle así el código y evitar problemas de seguridad

### **Problemáticas**

Al tener un tiempo reducido, el sistema no toma en cuenta todas las posibles fallas que pueda cometer un usuario, como ingresar letras en lugar de números u otras entradas que podrían hacer que el sistema falle. Por tanto, requiere más modificaciones en cuanto a regular las entradas del teclado. A pesar de ello, se intentó hacer lo más posible en cuanto a detectar errores. Sin embargo, hay que tener en cuenta esto, para posibles pruebas.

También tenemos en cuenta las capacidades que tiene, por tanto, es posible realizarle algunas modificaciones para que pueda acceder a bases de datos, o a sistemas distribuidos, para que se pueda acceder desde cualquier lugar.

Otro inconveniente que puede tener, es el entorno gráfico, se maneja todo desde la terminal y con entradas del teclado, como resultado, es algo a lo que no estamos acostumbrados hoy en día, sin embargo, esto también nos da acceso a computadoras que no tengan esas capacidades y además, podemos implementar un sistema gráfico desde cualquier otro lenguaje y realizar llamadas a estas clases y métodos, sin necesidad de modificar tan profundamente el funcionamiento del programa.

### **Uso**

Como el sistema fue desarrollado en C++, es necesario tener su compilador, puede instalar MinGW desde la página oficial:

- <https://www.mingw-w64.org/downloads/#msys2>

Para compilar y ejecutar desde terminal:

- Linux: `g++ main.cpp -o main && ./main`
- Windows: `g++ -o main.exe main.cpp`

## **Código**

Está disponible en:

- <https://github.com/LuisQuintanar29/pruebaTecnicaSS>