ARELLANO GRANADOS ANGEL MARIANO

218123444

DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN.

ESTRUCTURAS DE DATOS I

22A D03

DAVID ALEJANDRO GÓMEZ ANAYA

REPORTE ACT. 2

ARELLANO GRANADOS ANGEL MARIANO
218123444

1. Planteamiento del problema.

Un cine un independiente requiere un sistema computacional que ayude al control de su establecimiento. El cine dispone de 7 salas, 2 grandes y 5 medianas. Las salas medianas pueden albergar hasta 56 personas en 7 filas (“A” a “G”) de 8 personas (1 a 8, de izquierda a derecha). Las salas grandes están compuestas por 10 filas de (“A” a “J”) de 14 personas (1 a 14, de izquierda a derecha). El cine cuenta también con información acerca de la película que se está exhibiendo en cada sala, esta información es: nombre, director, duración, y horarios de reproducción en la sala. Puede estar la misma película en dos salas distintas, pero no puede haber dos películas distintas en la misma sala.

Las personas que compren un boleto para alguna película deberán elegir una película, un horario, una sala y un lugar disponible. Serán registrados con su nombre, un id único generado automáticamente, se le asociará con una película, una sala, un horario específico, y un asiento (fila, columna). A una persona se le puede decir que boleto (película, sala y hora de inicio) compró si proporciona su nombre o id. Si la búsqueda se realiza por nombre, se mostrarán todas las coincidencias y se mostrarán también los id.

A una sala se les puede agregar películas solamente si tiene horario disponible, el/los horarios los propone quien registra la película en la salsa. El horario del cine es de 9:00 hrs a 22:00 hrs. Una Sala puede tener hasta 5 horarios de películas y entre cada película tiene que haber, al menos, 30 minutos para la limpieza del lugar. Puede asumir que no se pueden registrar más de 2800 personas (lleno total de las salas en los 5 horarios disponibles en cada una).

En esta fase del proyecto, no es necesario conectar el programa al horario actual. Solamente se registrarán personas a los asientos, horarios, películas y salas; las películas y los horarios a las salas. Se puede eliminar una película, de un horario específico, de una sala, y por tanto, también los clientes que se encuentren registrados en esa película desaparecerán del sistema.

2. Objetivos:

1. Comprender el paradigma de programación orientado a objetos y utilizarlo para elaborar soluciones informáticas.
    1. Abstraer las entidades Cine, Sala, Cliente y Película.
    2. Diseñar las acciones de alta, consulta y eliminación de películas en salas.
    3. Diseñar las acciones de alta, consulta y eliminación de clientes.

ARELLANO GRANADOS ANGEL MARIANO
218123444

3. Marco teórico.

1. Clases y Objetos

Class: A class in C++ is the building block that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

Object: is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.

2. Herencia

Las clases nuevas se pueden derivar de clases existentes mediante un mecanismo denominado "herencia" Las clases que se utilizan para la derivación se conocen como "clases base" de una clase derivada determinada. Una clase derivada se declara mediante la sintaxis siguiente:

```
class Derived : [virtual] [access-specifier] Base {
   // member list
};
class Derived : [virtual] [access-specifier] Base1,
   [virtual] [access-specifier] Base2, . . .
{
   // member list
};
```

Las especificaciones base pueden contener un especificador de acceso, que es una de las palabras clave public, protected o private. Estos especificadores de acceso aparecen delante del nombre de la clase base y solo se aplican a esa clase base.

3. Headers

The names of program elements such as variables, functions, classes, and so on must be declared before they can be used.

The declaration tells the compiler whether the element is an int, a double, a function, a class or some other thing. Furthermore, each name must be declared (directly or indirectly) in every .cpp file in which it is used. When you compile a program, each .cpp file is compiled independently into a compilation unit.

ARELLANO GRANADOS ANGEL MARIANO
218123444

4. Desarrollo.

Inicie creando el proyecto y añadiendo todos los headers y cpp que necesitaría para elaborar la actividad.

Ver Ilustración 1 Proyecto

Tras declarar todos los atributos que pude notar en mi atracción del problema en cada uno de los headers decidí iniciar con la opción 4 del menú, pues si no se podían agregar películas y horarios al cine ninguna de las otras opciones funcionaria.

Para esto se muestra un menú si pregunta si se quiere ingresar una película o un horario, dependiendo se llama a la misma función *agregar()* pero con diferente parámetro, esta consigue el tamaño y numero de la sala en la que se quiere agregar con una validación simple.

Después se llama al método *Registrar()* de la clase cine que asi misma llamara a los métodos *agregarPelicula()* o *agregarHorario* de la clase sala dentro del arreglo de salas de cine, que pedirán datos para llamar a un constructor con sobre carga y agregara una película y para los horarios tiene una validación del horario para poder agregar un máximo de 5 funciones en puntos no lineales pero con un tiempo entre estas.

Ver Ilustración 2 Constructor con sobrecarga

Ver Ilustración 3 Validación Horario

Tras esto solo tenia que hacer que se mostraran las películas, salas y horarios de manera organizada para la opción 1 del menú, para esto cree un método de la clase cine *mostrarTodo()* que se conecta con método de la clase sala para mostrar con el siguiente formato *Película -> Sala -> Horarios-* .

Para la opción 5 eliminar decidí hacer una eliminación “Lógica” donde todos los datos seguirían ahí pero las banderas y contadores se reiniciarían para que al volver a hacer otro ciclo el programa piense que en esos espacios de memoria no hay nada y sobrescriba los datos.

Pasando a la parte más difícil en mi opinión es la opción 2 vender boleto donde en un método de la clase SalaGrande y SalaMediana se imprime una matriz para que el usuario pueda visualizar que asiento quiere, donde si esta disponible aparecerá la coordenada del asiento y si está ocupado aparecerá un “X” de la siguiente manera.

Ver Ilustración 4 Ejemplo de Matriz

Al finalizar el proceso se mostrar una confirmación de los daros usando una stringstream para concatenar todos los datos del asiento.

Por último esta la búsqueda que pude ser por ID o por nombre ambas usan el mismo código solo cambia el elemento que comparan (nombre o ID), el algoritmo se basa en una serie de validaciones que confirman que allá una película registrada, horarios registrados y al menos un cliente registrado, tras validad iterara en las matices de 3 dimensiones solo si hay un cliente en alguna (pero revisa todas) y comparara el ID o nombre dado e imprimirá todas las coincidencias.

Ver Ilustración 5 Algoritmo de Búsqueda

5. Pruebas y resultados.

Para la prueba de resultados use una inyección de datos que agrega una película a todas las salas y agrega 2 horarios no lineales a cada sala.

```
        MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
4
        1. Agregar Pelicula a Sala
        2. Agregar Horario a Pelicula
1
A que tipo de sala?
        1) Grande
        2) Mediana
1
En cual sala?
1) SALA 1
2) SALA 2
1
Nombre de la pelicula:
red
Director de la pelicula:
disney
Duracion de la pelicula (Minutos):
120
Registro exitoso
```

Aquí podemos ver un ejemplo de cómo se registra una película en una sala.

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
       MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
4
       1. Agregar Pelicula a Sala
       2. Agregar Horario a Pelicula
2
A que tipo de sala?
       1) Grande
       2) Mediana
1
En cual sala?
1) SALA 1
2) SALA 2
2
A que hora va a iniciar la funcion? (30 min = 0.5, ejemplo 9:30 = 9.5)
10.5
Registro exitoso
```

Y aquí un ejemplo de como se registra un horario a una película

```
       MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
1
Pelicula -> Sala -> Horarios
Salas Grandes:
red -> G1 -> 9.5 - 17 -

batman -> G2 -> 10.5 - 16.25 -

Salas Medianas:
encanto -> M1 -> 18.5 - 9 -

Avergers -> M2 -> 12 - 20 -

Saw -> M3 -> 19.5 - 9.5 -

Soul -> M4 -> 10.25 - 16.5 -

moonknight -> M5 -> 21 - 11.25 -
```

Asi se muestra las películas de cada sala con sus horarios

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
Seleccione una opcion
2
Pelicula -> Sala -> Horarios
Salas Grandes:
red -> G1 -> 9.5 - 17 -

batman -> G2 -> 10.5 - 16.25 -

Salas Medianas:
encanto -> M1 -> 18.5 - 9 -

Avergers -> M2 -> 12 - 20 -

Saw -> M3 -> 19.5 - 9.5 -

Soul -> M4 -> 10.25 - 16.5 -

moonknight -> M5 -> 21 - 11.25 -

A que tipo de sala?
        1) Grande
        2) Mediana
1
En cual sala?
1) SALA 1
2) SALA 2
1
Pelicula: red
Seleciona un horario
Horario 1 : 9.5
Horario 2 : 17
2
|-----------------------------------------------|
|A1|A2|A3|A4|A5|A6|A7|A8|A9|A10|A11|A12|A13|A14|
|B1|B2|B3|B4|B5|B6|B7|B8|B9|B10|B11|B12|B13|B14|
|C1|C2|C3|C4|C5|C6|C7|C8|C9|C10|C11|C12|C13|C14|
|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|
|E1|E2|E3|E4|E5|E6|E7|E8|E9|E10|E11|E12|E13|E14|
|F1|F2|F3|F4|F5|F6|F7|F8|F9|F10|F11|F12|F13|F14|
|G1|G2|G3|G4|G5|G6|G7|G8|G9|G10|G11|G12|G13|G14|
|H1|H2|H3|H4|H5|H6|H7|H8|H9|H10|H11|H12|H13|H14|
|I1|I2|I3|I4|I5|I6|I7|I8|I9|I10|I11|I12|I13|I14|
|J1|J2|J3|J4|J5|J6|J7|J8|J9|J10|J11|J12|J13|J14|
|-----------------------------------------------|
Seleccione un acciento disponible
Seleccione la fila (Letra):   C
Seleccione la columna (Numero):   9
A que nombre quedaria el asiento:   angel

Detalles Del Boleto:
Nombre: angel
ID: 42
Sala: G1
Asiento: C9
Horario: 17

Registro exitoso
```

Este es el proceso donde se compran los boletos, si intentamos comprar otro boleto en la misma sala la matriz muestra el asiento ocupado de la siguiente manera.

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
A que tipo de sala?
        1) Grande
        2) Mediana
1
En cual sala?
1) SALA 1
2) SALA 2
1
Pelicula: red
Seleciona un horario
Horario 1 : 9.5
Horario 2 : 17
2
|-----------------------------------------------|
|A1|A2|A3|A4|A5|A6|A7|A8|A9|A10|A11|A12|A13|A14|
|B1|B2|B3|B4|B5|B6|B7|B8|B9|B10|B11|B12|B13|B14|
|C1|C2|C3|C4|C5|C6|C7|C8|X |C10|C11|C12|C13|C14|
|D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14|
|E1|E2|E3|E4|E5|E6|E7|E8|E9|E10|E11|E12|E13|E14|
|F1|F2|F3|F4|F5|F6|F7|F8|F9|F10|F11|F12|F13|F14|
|G1|G2|G3|G4|G5|G6|G7|G8|G9|G10|G11|G12|G13|G14|
|H1|H2|H3|H4|H5|H6|H7|H8|H9|H10|H11|H12|H13|H14|
|I1|I2|I3|I4|I5|I6|I7|I8|I9|I10|I11|I12|I13|I14|
|J1|J2|J3|J4|J5|J6|J7|J8|J9|J10|J11|J12|J13|J14|
|-----------------------------------------------|
Seleccione un acciento disponible
Seleccione la fila (Letra):   A
Seleccione la columna (Numero):   4
A que nombre quedaria el asiento:   angel

Detalles Del Boleto:
Nombre: angel
ID: 468
Sala: G1
Asiento: A4
Horario: 17

Registro exitoso
```

Al comprar ambos boletos con el nombre de “angel” al buscar por nombre nos mostrara ambos resultados.

```
        MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
3
        1. Buscar por id
        2. Buscar por nombre
        0. Salir
2
Ingresa un Nombre:
angel
Detalles Del Boleto:
Nombre: angel
ID: 468
Sala: G1
Asiento: A4
Horario: 17

Detalles Del Boleto:
Nombre: angel
ID: 42
Sala: G1
Asiento: C9
Horario: 17
```

También si buscamos el ID 42 mostrara el segundo de los boletos que registramos.

```
        MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
3
        1. Buscar por id
        2. Buscar por nombre
        0. Salir
1
        Ingresa un ID:
42
Detalles Del Boleto:
Nombre: angel
ID: 42
Sala: G1
Asiento: C9
Horario: 17
```

Por último podemos eliminar la película de la sala grande 1 y al intentar mostrar las películas a perecerá que no hay nada registrado en esa sala.

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
        MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
5
A que tipo de sala?
        1) Grande
        2) Mediana
1
En cual sala?
1) SALA 1
2) SALA 2
1
Se a eliminado la sala con exito
        MENU
1. Mostrar catalogo
2. Vender un ticket
3. Buscar y mostrar cliente
4. Agregar Pelicula o Horario a Sala
5. Eliminar Pelicula de una sala
0. Salir
Seleccione una opcion
1
Pelicula -> Sala -> Horarios
Salas Grandes:
la sala no tiene una pelicula registrada
batman -> G2 -> 10.5 - 16.25 -

Salas Medianas:
encanto -> M1 -> 18.5 - 9 -

Avergers -> M2 -> 12 - 20 -

Saw -> M3 -> 19.5 - 9.5 -

Soul -> M4 -> 10.25 - 16.5 -

moonknight -> M5 -> 21 - 11.25 -
```

6. Conclusiones.

Con esta práctica fui capaz de organizar y aprovechar un proyecto donde existen más de un archivo pues le di uso a varios Headers y cpp’s a la hora de declarar e implementar clases para este programa, también logré adecuarme bien a la filosofía de hacer un programa usando la Programación Orientada a Objetos, que fue un gran cambio comparado a lo que ya estaba acostumbrado a hacer.

ARELLANO GRANADOS ANGEL MARIANO
218123444

7. Apéndice(s).

Ilustración 1 Proyecto.

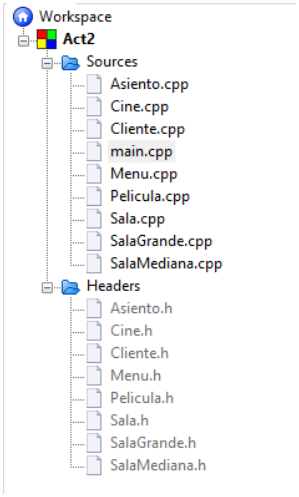

Ilustración 2 Constructor con sobrecarga

```
Pelicula::Pelicula()
{
    //ctor
}

Pelicula::Pelicula(string nombre,string director, int duracion){
    this->nombre=nombre;
    this->director=director;
    this->duracion=duracion;
}
```

Ilustración 3 Validación Horario.

```
void Sala::AgregarHorario(){
    if (uso){
        if(cont < 5){
            float horario;
            int aux,a1=0,a2=0;
            cout<<"A que hora va a iniciar la funcion? (30 min = 0.5, ejemplo 9:30 = 9.5)"<<endl;
            cin>>horario;
            aux=horario*60;//transformar a minutos
            if(aux>=540&&aux<1320){
                for(int i=0;i<cont;i++){
                    if(aux>=horarios[i]+pelicula.getDuracion()+30 || aux<=horarios[i])
                        a1++;
                    if(aux+pelicula.getDuracion()+30<=horarios[i] || aux+pelicula.getDuracion()+30>=horarios[i]+pelicula.getDuracion()+30)
                        a2++;
                }
                if(a1==cont&&a2==cont){
                    horarios[cont]=aux;
                    cont++;
                    cout<<"Registro exitoso"<<endl<<endl;
                }
                else
                    cout<<"el horario no esta disponible"<<endl;
```

Ilustración 4 Ejemplo De Matriz

ARELLANO GRANADOS ANGEL MARIANO
218123444

```
cout<<"||-----------------------------------------------||"
cout<<"||A1|A2|A3|A4|A5|A6|A7|A8|A9|A10|A11|A12|A13|A14||"
cout<<"||B1|B2|B3|B4|B5|B6|B7|B8|B9|B10|B11|B12|B13|B14||"
cout<<"||C1|C2|C3|C4|C5|C6|C7|C8|C9|C10|C11|C12|C13|C14||"
cout<<"||D1|D2|D3|D4|D5|D6|D7|D8|D9|D10|D11|D12|D13|D14||"
cout<<"||E1|E2|E3|E4|E5|E6|E7|E8|E9|E10|E11|E12|E13|E14||"
cout<<"||F1|F2|F3|F4|F5|F6|F7|F8|F9|F10|F11|F12|F13|F14||"
cout<<"||G1|G2|G3|G4|G5|G6|G7|G8|G9|G10|G11|G12|G13|G14||"
cout<<"||H1|H2|H3|H4|H5|H6|H7|H8|H9|H10|H11|H12|H13|H14||"
cout<<"||I1|I2|I3|I4|I5|I6|I7|I8|I9|I10|I11|I12|I13|I14||"
cout<<"||J1|J2|J3|J4|J5|J6|J7|J8|J9|J10|J11|J12|J13|J14||"
cout<<"||-----------------------------------------------||"
```

Ilustración 5 Algoritmo de Búsqueda

```
void SalaGrande::BuscaId(int id){
    if(getUso()){
        if(getCont()>0){
            if(contAsientos>0){
                for(int z=0;z<getCont();z++){
                    for(int x=0;x<10;x++){
                        for(int y=0;y<14;y++){
                            if(asientos[x][y][z].getId()==id){
                                cout<<asientos[x][y][z].Detalles()<<endl;
                            }
                        }
                    }
                }
            }
        }
    }
}
```