

# Universidad de Antioquia

Departamento de Ingeniería Electrónica y Telecomunicaciones



## Parcial Informática II

7 de mayo de 2020

### 1. Instrucciones

El objetivo principal de ésta actividad es poner a prueba sus destrezas en análisis de problemas, POO y manejo del lenguaje C++. Si usted ha llevado un proceso disciplinado de aprendizaje a lo largo del semestre, ésta es una oportunidad de demostrarlo, podrá plantear una solución viable y su resultado será satisfactorio. En caso contrario, podrá identificar sus debilidades y deberá tomar medidas con el fin de poder ser apto para enfrentar situaciones como las que se plantean en la Sección 2.

Trate en lo posible de valorar muy bien la verdadera complejidad del problema planteado, no se rinda antes de intentarlo y de plantear los posibles escenarios de solución. Se dará cuenta que si bien al principio le puede parecer largo y difícil, realmente ya ha tenido la oportunidad de enfrentarse a problemas similares en las prácticas de laboratorio. Si se toma el tiempo adecuado para analizar, el código adicional no va a ser mucho.

Esperamos que disfrute del desafío propuesto. Le recomendamos que lea primero todo el documento antes de comenzar y asegúrese de entender muy bien las instrucciones antes de desarrollar ésta actividad evaluativa.

#### 1.1. Requisitos mínimos

A continuación se describen los requisitos que se deben cumplir. El incumplimiento de cualquiera de ellos implicará que su nota sea cero.

- Se debe crear un repositorio público en el cual se van a poder cargar todos los archivos relacionados a la solución planteada por usted (código fuente y anexos).
- En el repositorio se debe evidenciar el adecuado uso del mismo. Es decir, dado que la solución se va a ir construyendo, deben haber varios commits que muestren la evolución del programa desarrollado.
- El programa debe ser una aplicación de consola desarrollada en el entorno de desarrollo Qt Creator.

- La solución propuesta debe hacerse utilizando POO.
- La aplicación debe ser implementada en lenguaje C++.
- El código fuente debe estar debidamente comentado.
- El código fuente debe compilar.
- Se debe adjuntar un enlace de youtube a un video que debe incluir lo siguiente:
  - Presentación de la solución planteada. Análisis realizado y explicación de la arquitectura del programa (4 minutos máximo).
  - Demostración de funcionamiento del programa. Explicar como funcionan los módulos. En tiempo de ejecución no el código (4 minutos máximo).
  - Explicación del código fuente. Tenga en cuenta que debe justificar la elección de los contenedores usados. Por qué eligió uno u otro contenedor en cada caso particular y que ventaja ofrecen estos en comparación de otros contenedores que también podrían haber sido usados (7 minutos máximo).
- La duración total del video no debe exceder 15 minutos.
- Una vez se cumpla el plazo de entrega, no se deberá hacer modificación alguna al repositorio.

## 2. Planteamiento del problema

Realice la gestión del sistema de atención de un cine para automatizar el proceso de compra de asientos para una película (Figura 1).

Usted tiene el encargo de realizar una aplicación que cumpla con el siguiente conjunto de requerimientos.

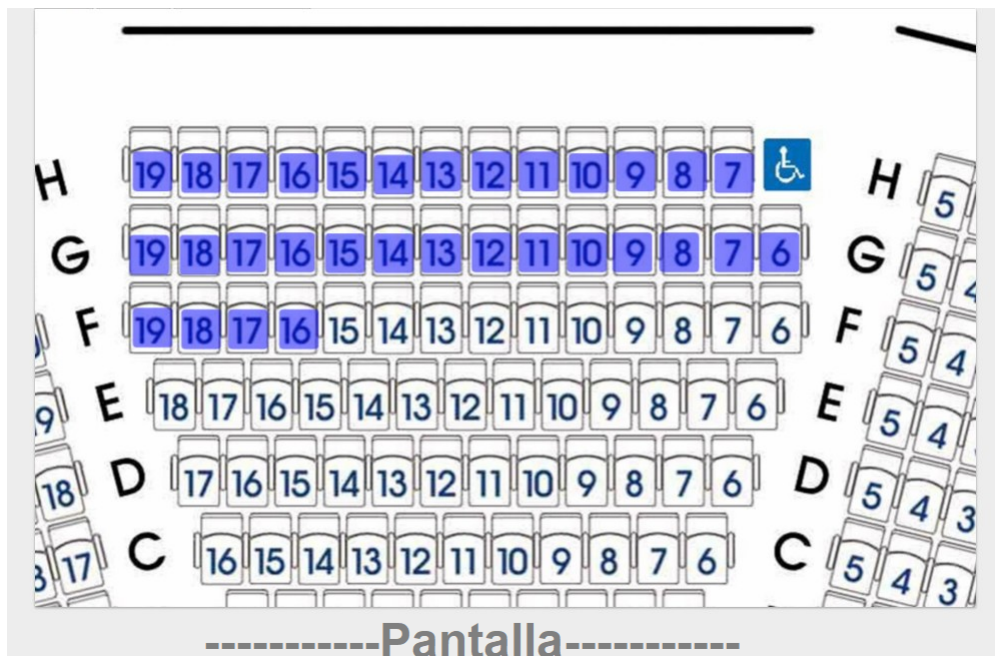


Figura 1: Sala de cine con asientos numerados.

## 2.1. Evaluación

- [0.5] Tener una interfaz de usuario clara (usando la consola como en la práctica 3) que permita desarrollar todo el proceso de compra y venta de la forma más amigable posible.

**Nota 1:** Recuerde que debe tener dos perfiles, uno para el **administrador** del punto de venta y otro para los **usuarios** de los asientos con las mismas condiciones de seguridad y acceso como se plantearon en la práctica 3 de laboratorio<sup>1</sup>.

**Nota 2:** Debe permitir registrar todas las transacciones y guardarlas en archivos, ya que la aplicación se puede abrir/cerrar y la información debe estar disponible.

### 2.1.1. Funcionalidades interfaz Administrador punto de venta

- [0.75] Tener la opción de ingresar las películas que se presentarán en cartelera y próximos estrenos, esta información debe ser administrada por medio de una base de datos<sup>2</sup> como se muestra a continuación:

Cuadro 1: Películas ofertadas por la sala de cine.

ID	Nombre	Género	Duración	Sala/Hora	Asientos disponibles	Clas.
1	Misión de rescate	Acción	120 min.	4/7pm	57/140	16+
2	Triple Frontera	Acción	125 min.	2/7pm	28/140	16+
3	Bird Box: A ciegas	Drama	124 min.	3/10pm	74/140	16+
4	A mi altura	Infantil	102 min.	1/10am	14/50	7+
5	Ricardo Quevedo: Hay gente así	Comedia	58 min.	7/7pm	49/140	16+

- [0.75] Permitir ofertar asientos para vender a los usuarios basandose en la disponibilidad <sup>3,4</sup>.
- [0.5] Generar un reporte con las ventas realizadas del día.

### 2.1.2. Funcionalidades interfaz usuario para venta de asientos

- [0.5] Una vez que el usuario ingrese se debe desplegar las funciones disponibles, tome como ejemplo la información del Cuadro 1.
- [0.5] Posteriormente a la elección de la película, debe permitir al usuario seleccionar uno de los asientos disponibles <sup>5</sup> y realizar el pago de la misma manera como en el problema 1 de la práctica 2 de laboratorio.

<sup>1</sup>Puede usar la práctica 3 de laboratorio como referencia para gestionar los usuarios.

<sup>2</sup>La base de datos se debe administrar como en la práctica 3 de laboratorio usando archivos.

<sup>3</sup>Puede basarse en alguna sala de cine popular para la clasificación de los asientos, mínimo 3 tipos de asiento se deben ofrecer a los usuarios incluyendo precio diferencial.

<sup>4</sup>La actualización de asientos en las salas se realiza cuando el usuario compre uno de los asientos en las salas disponibles.

<sup>5</sup>Tome como referencia la Figura 1 y el Cuadro 2 para diseñar la interfaz (consola) y mostrar los asientos disponibles y el precio

Cuadro 2: Tipo de asiento.

Tipo	Clasificación	Costo
1	General 2D	\$7900
2	General 3D	\$10800
3	VibroSound 2D	\$9900
4	VibroSound 3D	\$11900

- [0.5] Luego preguntar al usuario la ubicación del asiento y la sala, se finaliza la compra y la interfaz vuelve al estado inicial para permitir la compra a otro usuario.

### 2.1.3. Consideraciones de implementación.

- [0.5] Actualizar la disponibilidad de los asientos cada vez que se finalice un proceso de compra por parte de los usuarios.
- [0.5] Realiza adecuadamente la implementación del sistema usando contenedores, plantillas y archivos.