**Documentación**

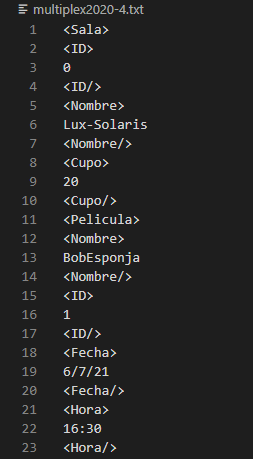
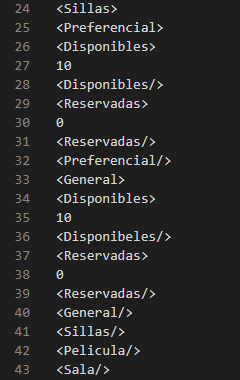
Juan Esteban Reyes

Alejandro Sacristan Leal

Lukas Fernando Rodríguez Pelaés

**Documento texto inicial**

Para recibir los datos y crear las salas se crea un documento de texto con el lenguaje XML, con las diferentes funciones vamos a leer cada dato de este para obtener las salas y de igual manera crear funciones para crear salas, un ejemplo de sala creada en el documento es esta:

Como se puede observar, cada dato del archivo esta entre “<>” y es cerrado con “</>” y en medio de estos se encuentra el dato a extraer y a llenar

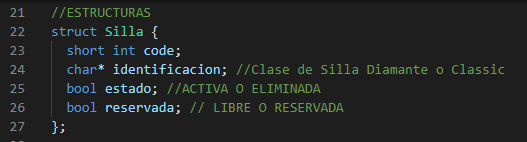
**Estructuras:**

Para este proyecto decidimos utilizar 4 estructuras

1. Estructura silla

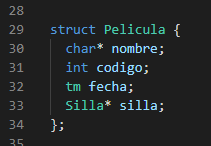
Esta estructura llamada “Silla” contiene los datos de esta, como lo son un short int para su código, un char\* para llenar la identificación de la silla (General o Preferencial),

Un bool llamado “estado” para definir si una silla esta disponible en una sala o no y por último un bool “Reservada” para saber si ya está ocupada.



1. Estructura Película

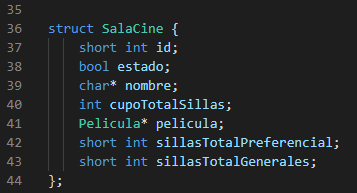
Esta estructura es utilizada para definir los datos de una película, como lo son su nombre (guardado en un char\*), su código, la fecha (guardada en tm que nos permite manipular fácilmente los datos de tiempo) y una silla de tipo Silla\* que es la estructura mencionada anteriormente



1. Estructura SalaCine

Esta estructura es utilizada para declarar los datos que conforman una sala.

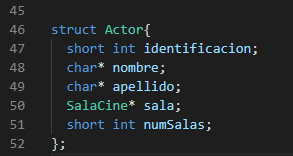
Un entero “CupoTotalSillas” para guardar la totalidad de las sillas en la sala, tres short int: uno para el id de la sala y dos para las sillas totales de tipo preferencial y general, un bool “estado”, en la que se define si esta activa o no, un char\* “nombre” que contiene el nombre de la sala y una “película” de tipo Película\* que guardara todos los datos de la película.



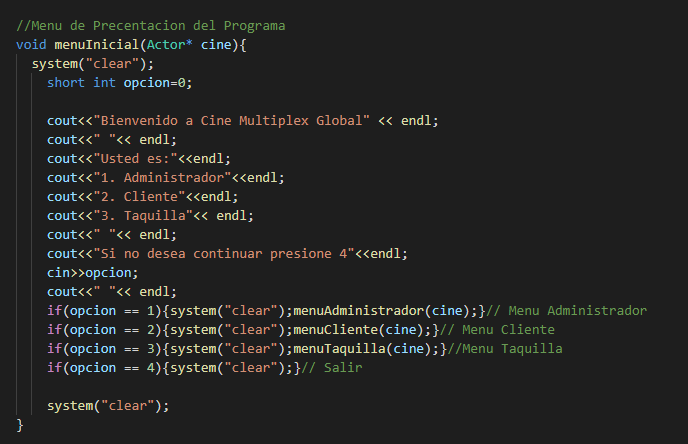
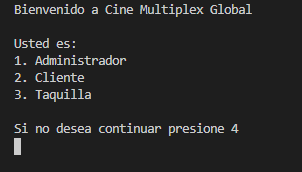
1. Estructura Actor

En esta estructura se guardan los datos del cliente.

Se utilizan, dos short int para guardar la identificación de la persona y el numero de la sala que le corresponde, dos char\* para guardar el nombre y apellido y una estructura sala de tipo SalaCine para acceder a los datos de esta.



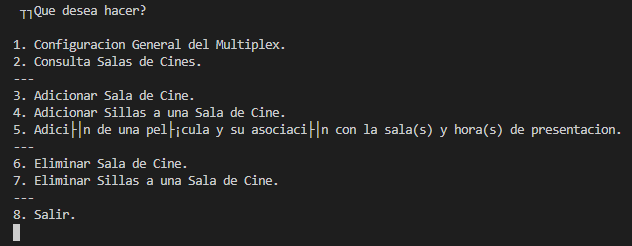
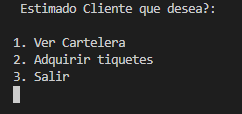
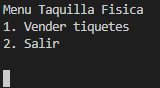
**Menú Principal:**

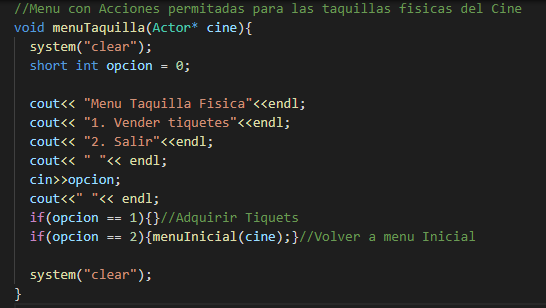
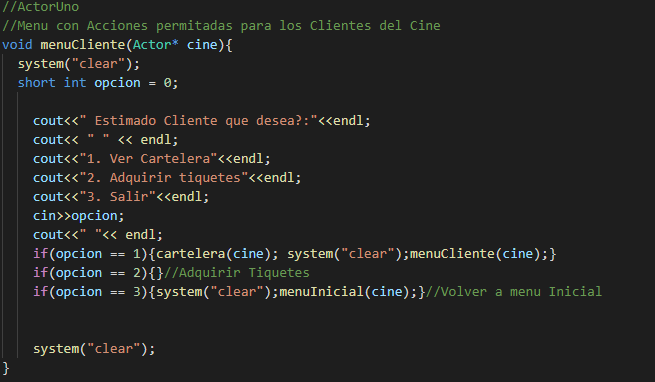
Empezamos con la función para crear un menú, esta función es de tipo void y recibe como parámetros una variable tipo “Actor\*”

En este menú que accederemos a las 3 categorías, Administrador, Cliente y taquilla, cada una tendrá un menú nuevo, se usan “couts” y condiciones dependiendo la opción del usuario, estas opciones están vinculadas a las funciones de los demás menús

Administrador Cliente Taquilla

Cada menú tiene la misma estructura del principal, siendo de tipo void, con la diferencia que según la opción digita se vincularan funciones que le den el debido funcionamiento.

Falta incluir en el menú algunas funciones que se encuentran en progreso.

**HTML – Cartelera:**

Al momento de hacer el HTML tuvimos que investigar el lenguaje, aprendiendo su funcionamiento y haciendo pruebas en un índex aparte

Cuando creamos la página nos pareció que se veía demasiado simple, tras investigar encontramos css con el cual podíamos cambiar el diseño con etiquetas e ID que declarábamos en HTML

Texto

Descripción generada automáticamente Texto

Descripción generada automáticamente

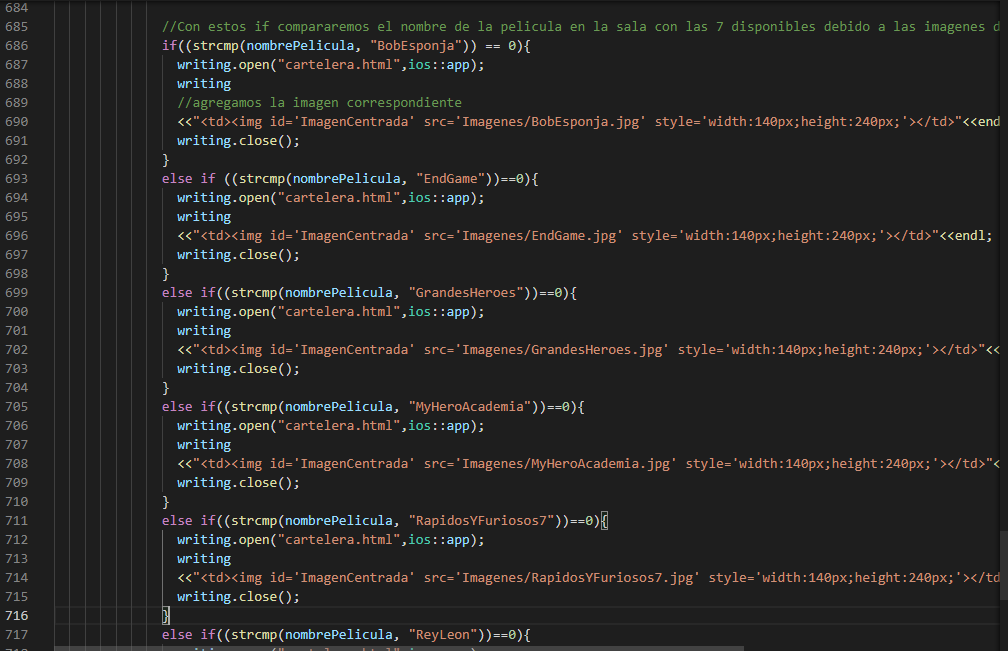
Aprendimos a incluir imágenes

Imagen 10

Por último en c++ añadimos todo el código y por medio de ciclos “for” y condiciones “if” creamos las tablas del tamaño que nos convenía.

La función aquí utilizada, es llamada “Cartelera” y es de tipo void, esta recibe como parámetro una variable tipo “Actor\*” para que disponga de datos como lo son, nombre de película, horario y sala.

Texto

Descripción generada automáticamente 

Este fue el progreso de la pagina

Imagen que contiene Texto

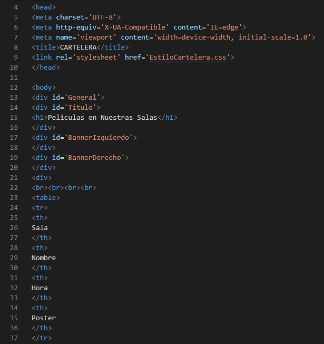
Descripción generada automáticamente Escala de tiempo

Descripción generada automáticamente

Escala de tiempo

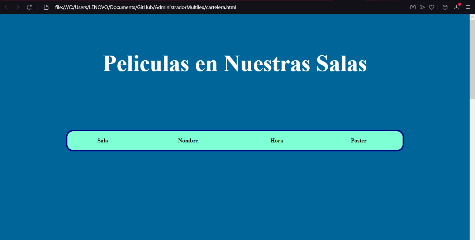
Descripción generada automáticamente con confianza baja Captura de pantalla de un celular

Descripción generada automáticamente

Falta lograr que los datos se puedan ingresar de forma adecuada y que coincidan con las imágenes y la cantidad de salas deseadas.

Por el momento este es el html que se esta imprimiendo:

Avance: El fallo no se encuentra en el for debido a que se realizo la prueba con un numero aleatorio y imprimió la tabla de ese tamaño:

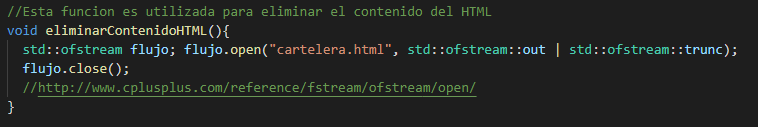


**Eliminar Contenido HTML:**

Para evitar errores en el HTML, por ejemplo que al momento de mostrar otra cartelera se duplique la tabla, se creo “eliminarContenidoHTML”

Con esta función de tipo void y que no recibe parámetros, se eliminara todo lo escrito en el archivo “cartelera.html” esta función es utilizada al comienzo de “cartelera” para asegurarnos de que el archivo esta vacío.

Esta es la función:



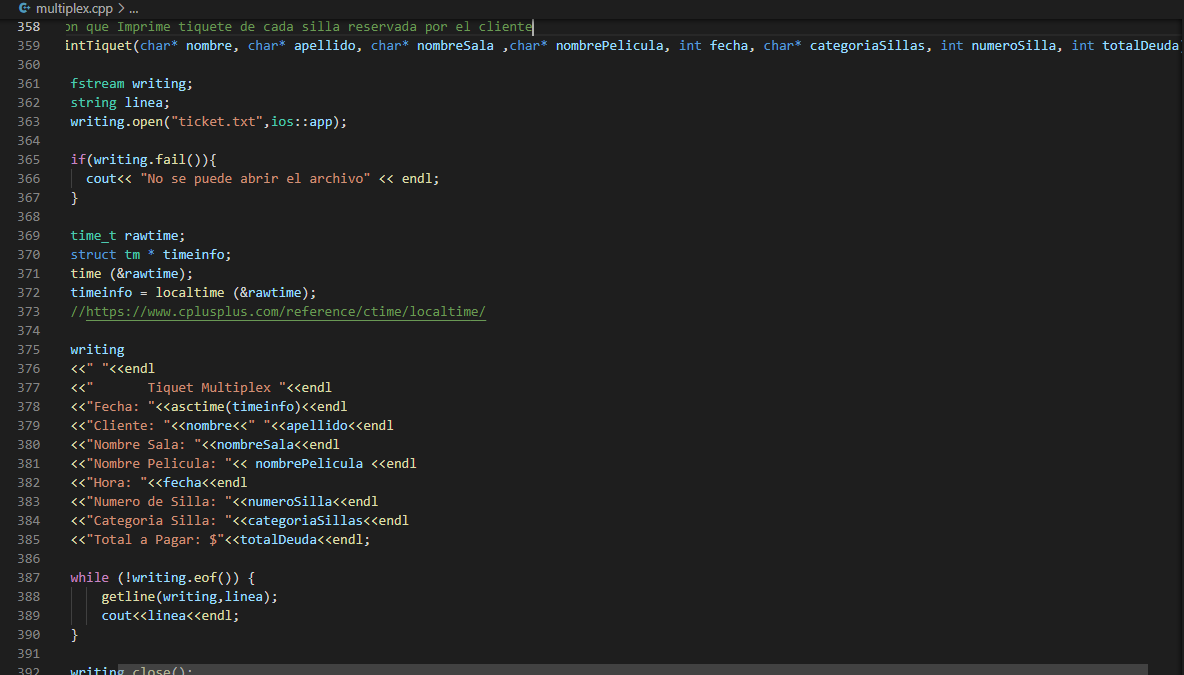
**Print Tiquet:**

Print Tiquet es una función de tipo void que nos ayudara a retornar en un documento de texto el tiquet de una persona, esta inicia su funcionamiento dentro de la función “VentaTiquete” en la que se llenan los datos y luego son ingresados a Print Tiquet, en “VentaTiquete” la encontramos dentro de un for dependiendo de la cantidad de sillas que pidió la persona.

La función cuenta con 8 parámetros que recibe para completar su funcionamiento:

5 parámetros tipo char\* que se usaran para imprimir el nombre y apellido de la persona, uno para el nombre de la sala, el nombre de la película y el último el tipo de categoría de la silla (General, Preferencial)

3 parámetros enteros, en donde se guarda la fecha, el numero de silla que pidió la persona y uno para la deuda total.

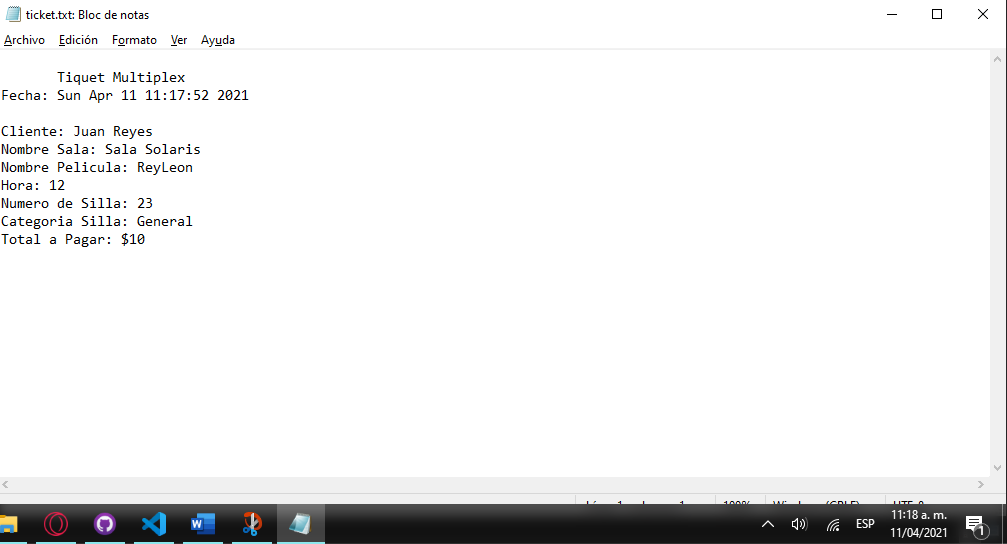


Luego con el uso de fstream se creo un archivo de texto llamado “ticket.txt” para editarlo.

En él se escribieron por medio de couts los datos recibidos y se finalizo cerrando el documento de texto.

También se utilizo una función de tipo, su funcionalidad es dar la hora del momento en la que se creo el tíquet.

Después de correr el programa nos encontramos con esto:



Un documento.txt con la hora del momento de su creación y los datos que fueron recibidor anteriormente.

**VentaTiquete**

**Cargar Configuración Multiplex**

**Crear Silla**

**Crear Sala**

**Eliminar Sala**

Eliminar\_sala es una función de tipo SalaCine\* que recibe como parámetros:

Un entero preguntando por la sala a eliminar llamado “numeroSalaEliminar” y una estructura SalaCine \*sala, recibiendo asi la estructura de una sala

Su funcionamiento consiste en cambiar el estado de la sala de true a false por medio de la variable “estado” que se encuentra en la estructura sala.

Por último retorna “sala” guardando asi el cambio Realizado

Su código es el siguiente:

