PONTIFICIA UNIVERSIDAD JAVERIANA
FACULTAD DE INGENERÍA
PROGRAMA INGENERÍA DE SISTEMAS
CURSO DE PROGRAMCION AVANZADA

ENTREGA CINEAPP

PROFESOR EDGAR ENRIQUE RUÍZ GARCÍA

ELABORADO POR JAIDER DAZA, KEVIN LEONARDO LÓPEZ GARCIA, DAVID ALEJANDRO SEQUERA LIÉVANO & JHOSEPH SAMIRT LIZARAZO MURCIA

BOGOTÁ D.C., 12 DE ABRIL DEL 2021

Herramientas:

- Visual Studio Code
- Git
- GitHub
- CodeBlocks

Lenguajes:

- HTML
- CSS
- JSON
- JavaScript
- C++

Requerimientos:

✓ Configurar inicialmente el multiplex (salas, sillas, películas y horarios) desde un archivo texto

✓ Consulta de salas: Se debe mostrar las sillas e indicar el tipo y si están libres u ocupadas

✓ Adicionar y Eliminar una sala de cine al múltiplex.

✓ Adicionar y Eliminar sillas a una sala de cine.

✓ Adición de una película y su asociación con la sala(s) y hora(s) de presentación.

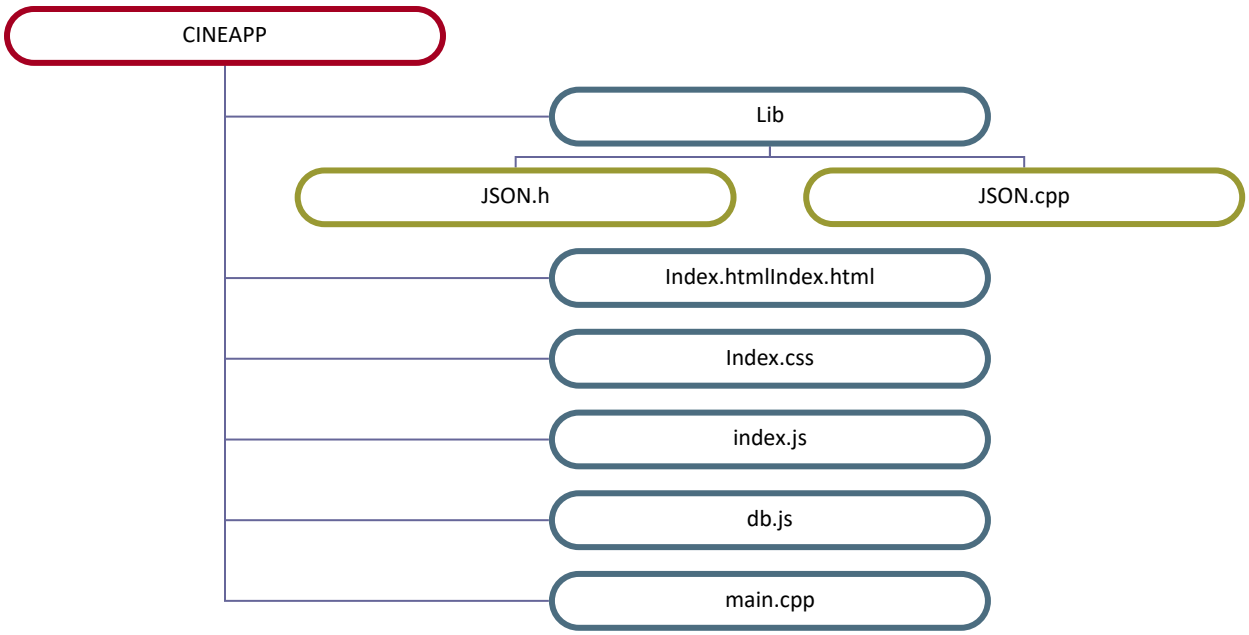
✓ Venta de tiquetes para ingreso a una sala de cine (Se debe generar el tiquete en un archivo texto).

✓ Consulta de Cartelera, el sistema debe generar un archivo HTML básico que haga uso de tablas (tema a investigar), al abrir el archivo en un navegador se mostrará la cartelera del multiplex. La cartelera incluye todas las películas que se están presentando, así como la sala y horario, para cada película se debe incluir una imagen que será mostrada en el navegador. El diseño de la página es libertad del grupo, pero el despliegue de la cartelera debe hacer uso de tablas.

Documentación:

Directorio:

El directorio se compone de una carpeta principal en la cual se encuentra un archivo HTML Donde se encuentra una estructura base para la presentación del proyecto, un archivo CSS para los estilos, y dos archivos JavaScript, uno se encarga de la lógica y funcionamiento de la pagina y el otro del almacenamiento de datos. En cuanto a el programa que opera el proyecto se encuentra un archivo C++ que maneja la interfaz en consola para el cambio de dato, también tiene una librería propia que se encarga del manejo del archivo JavaScript, actúa como una librería para un JSON, pero por temas de simplificación del trabajo, edita el archivo JavaScript.



Archivos:

En el archivos index.html existen 4 campos dentro de los cuales el archivo index.js inserta los datos, el archivo index.js tiene unas plantillas las cuales son llenadas por datos dentro del archivo db.js este tiene la estructura de tablas SQL básica a fin de que los datos se entiendan y tengan coherencia lógica, su forma y estructura son las de un objeto JavaScript, para el manejo interno de los datos esta una aplicación en consola la cual se vale de una librería para el manejo de estos objetos JavaScript dentro del archivo db.js.

Manejo de datos:

Librería

```
struct Object {
     int n;
     char** field;
     char** value;
     int start;
     int end;
 };
struct ObjectArray {
     int n;
     Object* object;
 };
```

Estructura:

Las estructuras presentadas a continuación representan la forma como se manejan los datos dentro del archivo, aunque en realidad estos se vean reflejados en la estructura de un objeto JSON dentro de la librería ya que de esta manera se logran dos objetivos, unificar el número de funciones que realizan el manejo de registros y poder manejar las estructuras de manera dinámica sin tener que estar limitados a una estructura fija.

Al momento de plantear la estructura se pensó en un modelo estandarizado que ayude a agilizar y organizar de manera adecuada la organización de los datos de este modo se planteó un modelo simple SQL el cual no maneja tipos datos entre si sino “llaves”, las cuales permiten reutilizar los modelos de peticiones ya conocidos sin necesidad de inventar nuevos.

```
struct Movie{
    char* id;
    char* name;
    char* img;
};
struct Room{
    char* id;
    char* name;
    char* seats;//int
};
struct Showtime{
    char* id;
    char* time;
    char* room_id;
    char* movie_id;
};
struct Seats{
    char* id;
    char type;
    char state;
    char* position;
    char* showtime_id;
};
struct Ticket{
    char* id;
    char name;
    char seat_id;
};
```

Funciones:

Liberia:

Manejo interno del archivo, como creación de registros, modificación de campos, extracción de registros, conversión de tipos a lenguaje C++ y generación de identificadores únicos para los registros. Dentro de la librería existen funciones externas para su el uso de programas externos, pero también que ayudan a simplificar el trabajo de funciones externas.

Internas:

```
string generarFecha()
void changePrice(string &general,string &preferencial)
char* stoc(string a)
char **createMatrix(int f, int c)
void fillMatrix(char** m, int f, int c)
void resizing(char*& m, int p)
string genID()
```

Externas:

```
Object StoO(char*);
char* OtoS(Object);
Object getObject(std::string path, std::string id);
ObjectArray getObjectsByTable(std::string path, std::string tableName);
ObjectArray getObjectsByKey(std::string path, std::string field, std::string value);
void printObject(Object);
void printObjectArray(ObjectArray );
Object templateObject(std::string fields);
bool insertNewObject(std::string path, std::string tableName, Object object);
bool createObject(std::string path, std::string tableName, std::string fields);
bool insertObject(std::string path, Object object);
bool changeField(Object &object, std::string field, std::string value);
 bool updateObject(string path, string id, string field, string value)
bool insertField(Object &object,int p, std::string field, std::string value);
bool deleteObject(std::string path, std::string id);
bool fileEdit(std::string path,int position = 0, int delay = 0, char* token = "");
```

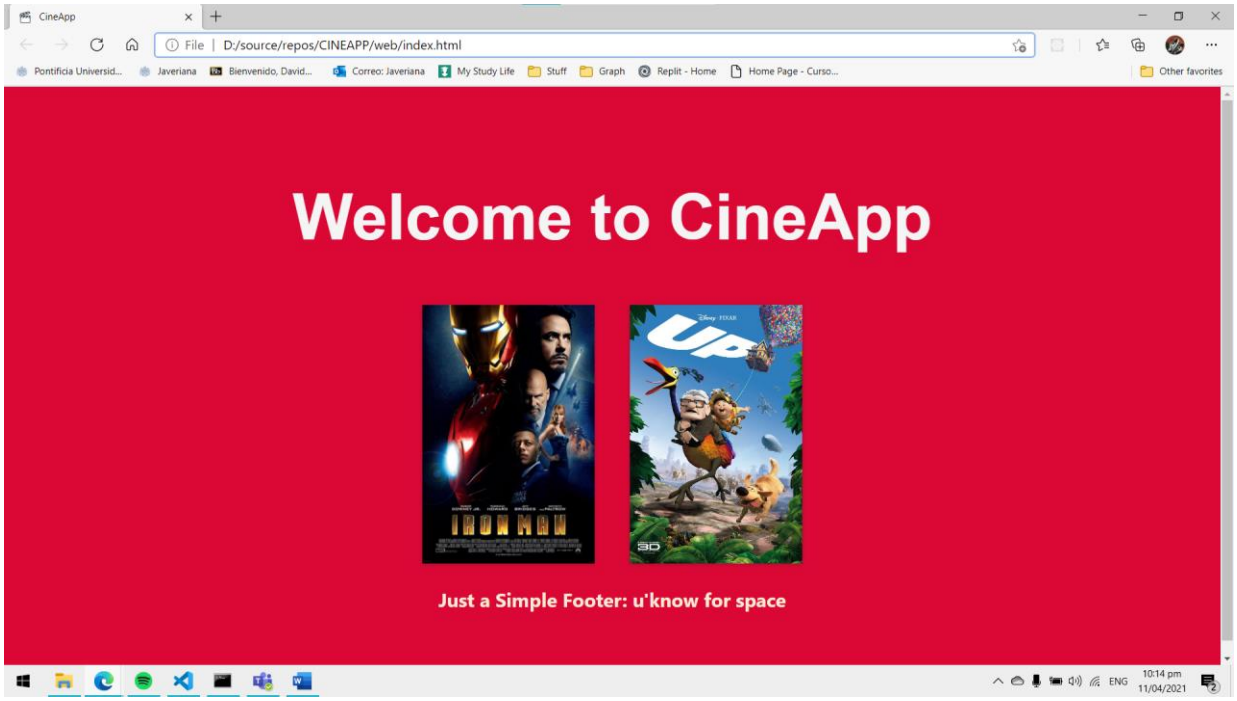
Programa:

Contiene la presentación al administrador del sistema, el manejo de peticiones y cambios al archivo, lógica del proyecto de cine, las funciones se hicieron autodescriptivas con mucha dedicación.

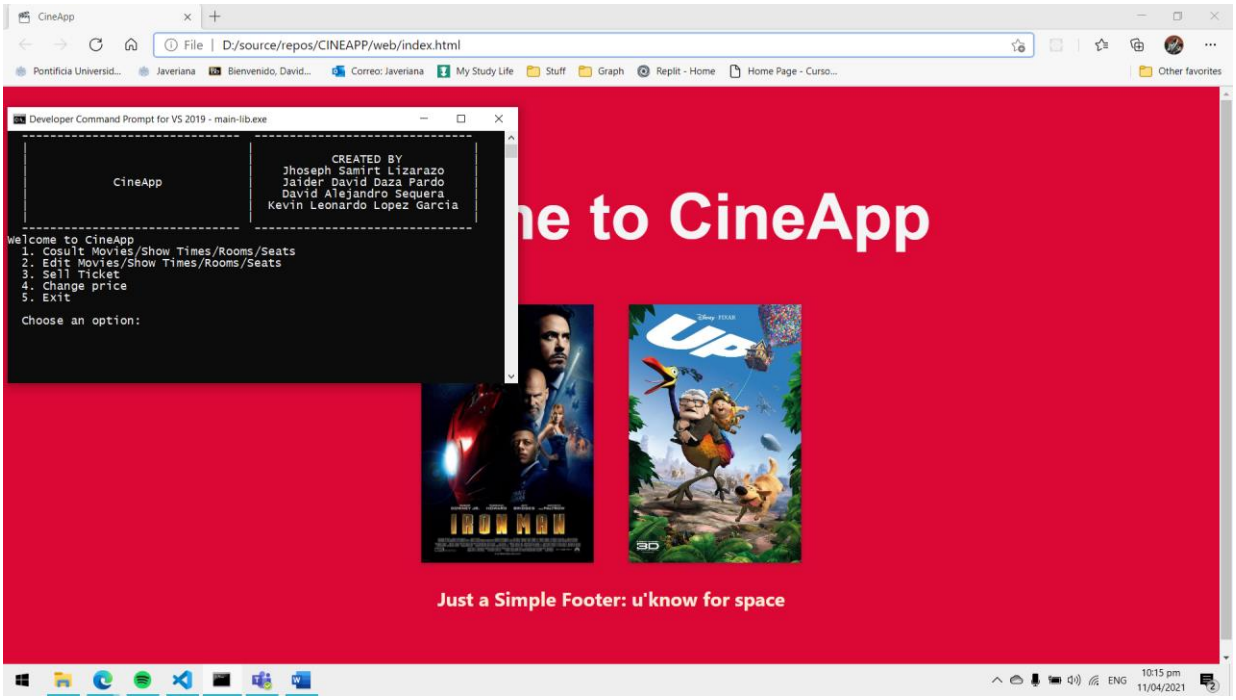
```
void header();
void cleanMemoryObject(JSON::Object);
void cleanMemoryArray(JSON::ObjectArray);
void showMovies();
void addMovie();
void addRoom();
void addShowtime();
void addSeat();
void addSeatsToShowroom(string);
void deleteMovie(string);
void deleteShowtime(string);
void deleteRoom(string);
void deleteSeat(string );
void deleteSeatByMovie();
void deleteShowtimeByMovie(string);
bool sellSeats(string,string);
void generateTicket(string,string,string,string,string);
```

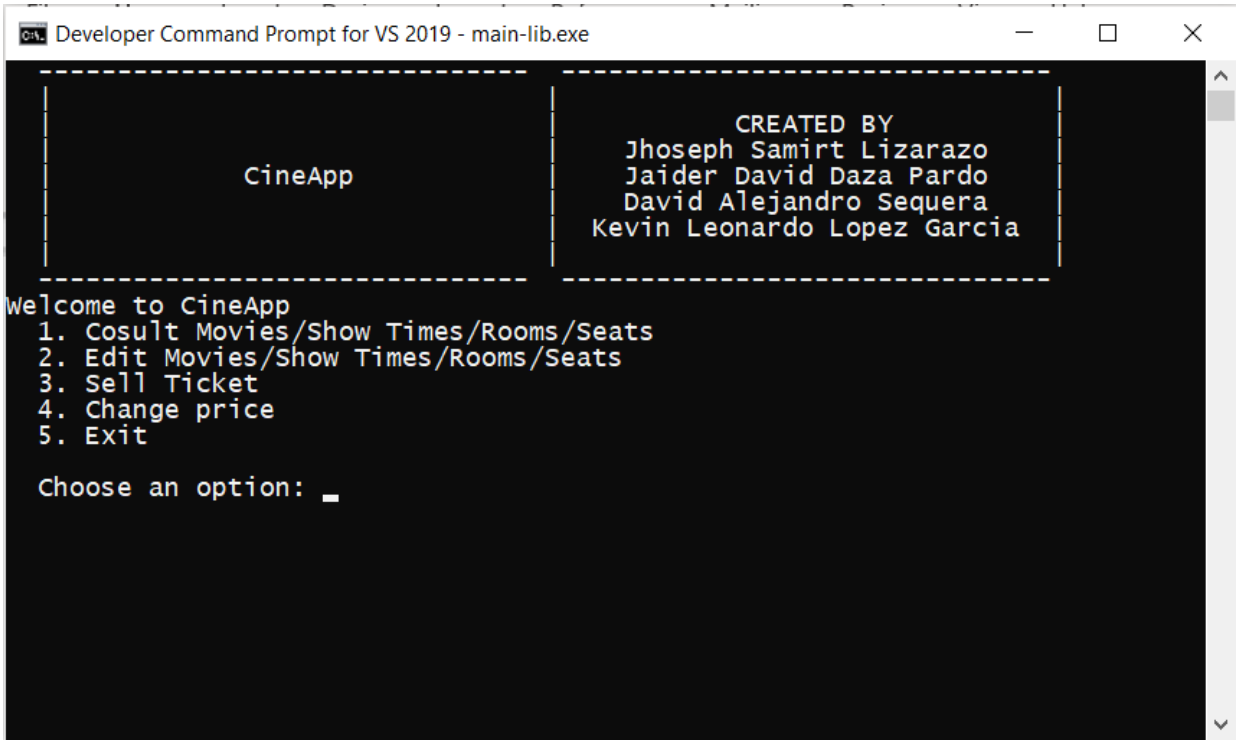
Pruebas:

1. HTML Básico

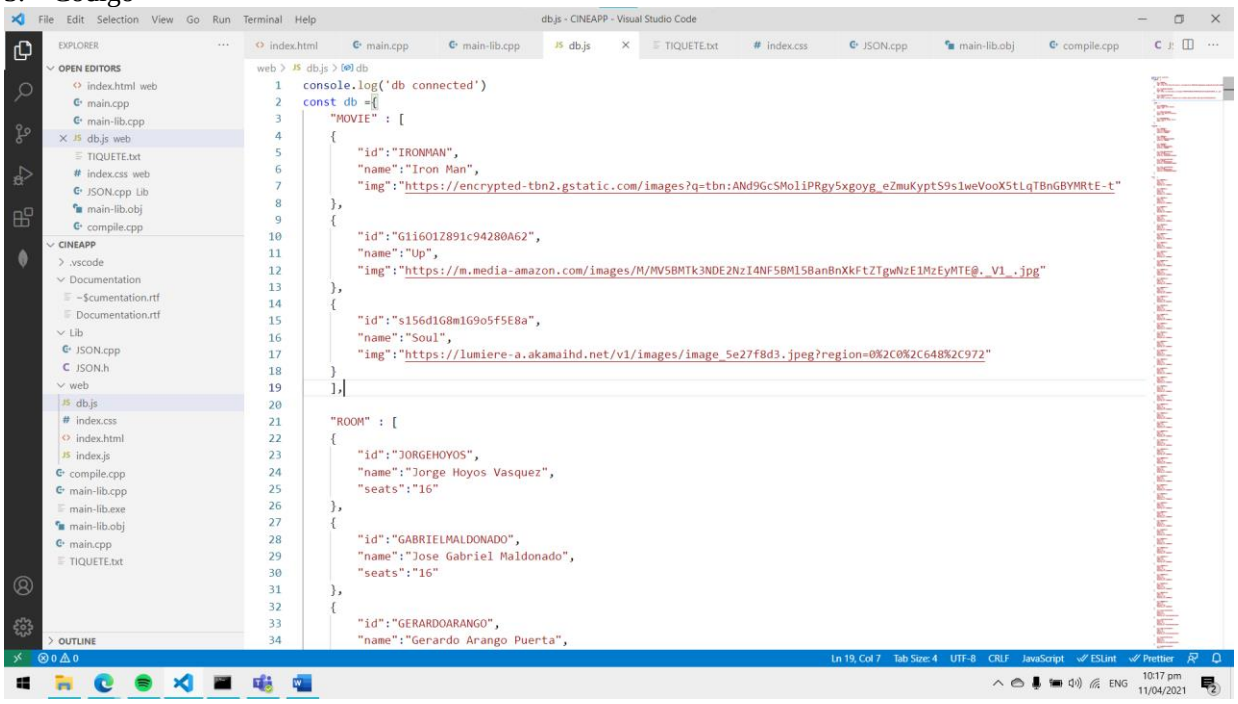


2. Consola

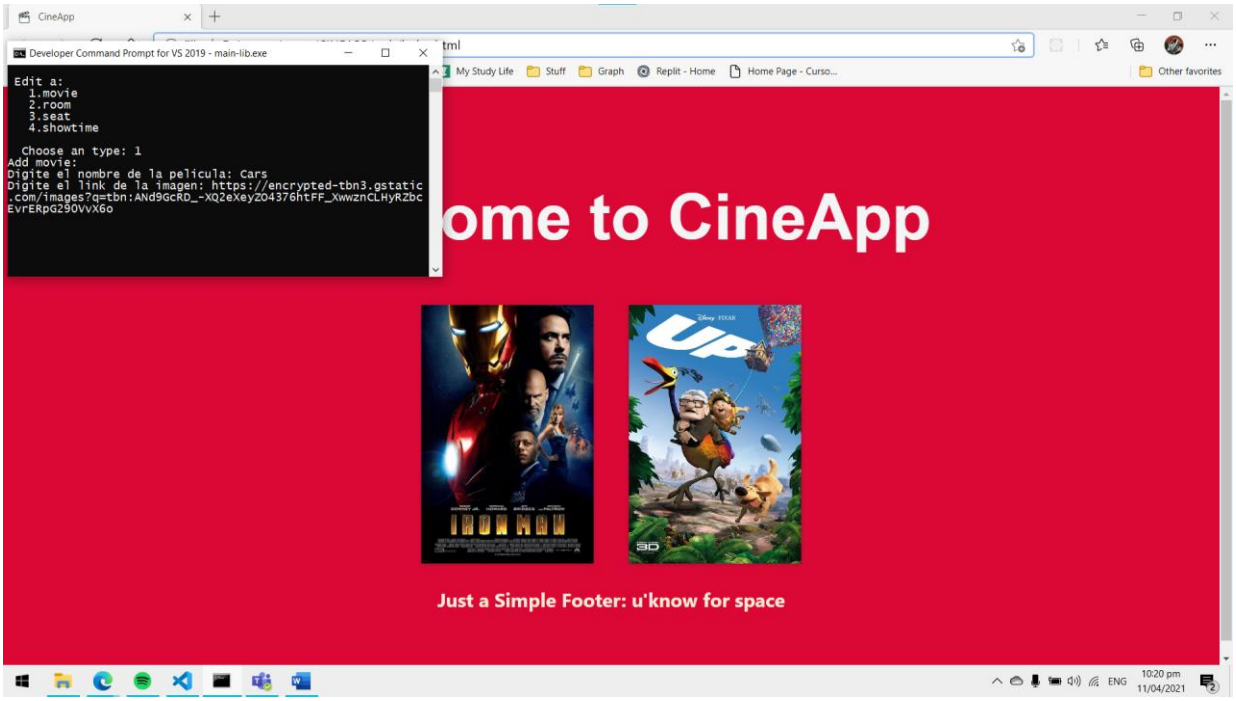


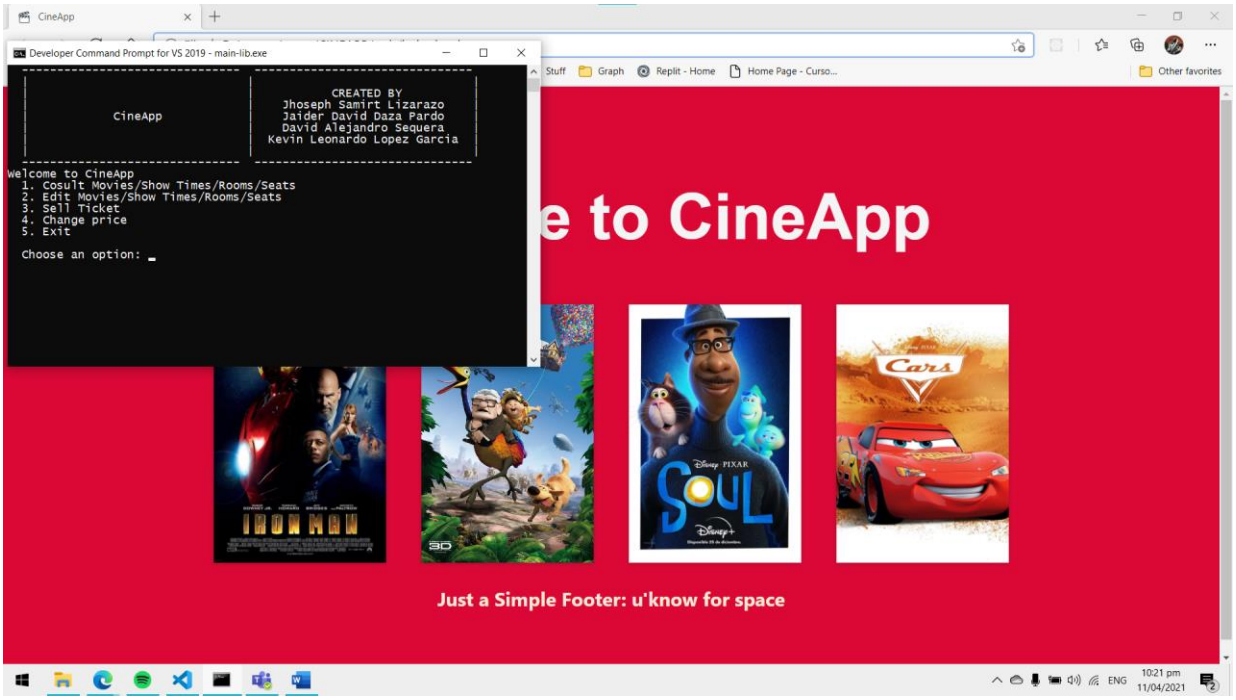


3. Código

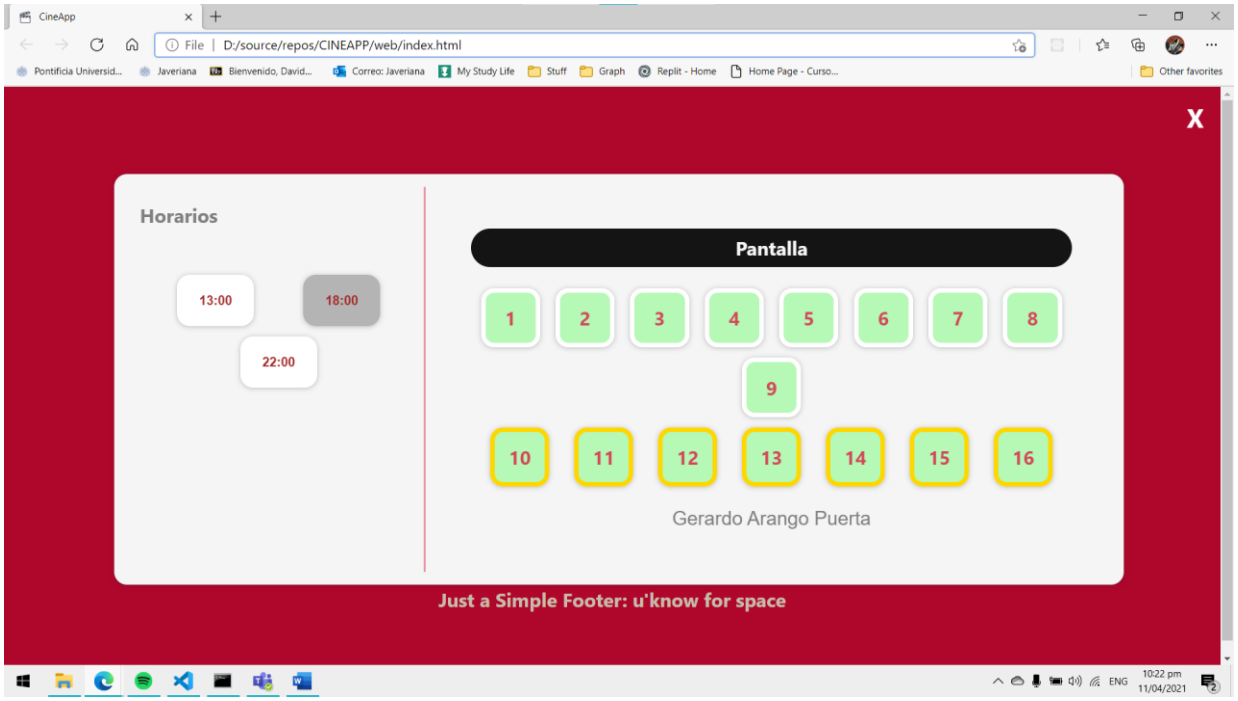


4. Añadir una película

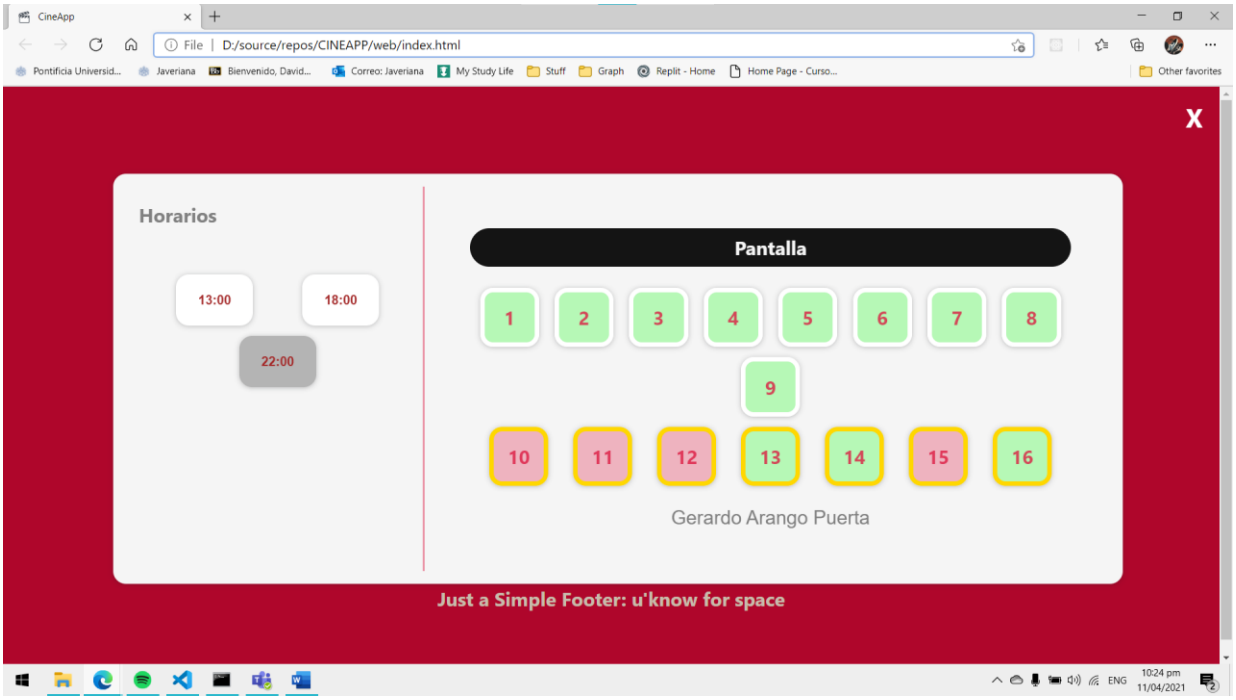

CineApp
Developer Command Prompt for VS 2019 - main-lib.exe
Edit a:
1.movie
2.room
3.seat
4.showtime
Choose an type: 1
Add movie:
Digite el nombre de la pelicula: Cars
Digite el link de la imagen: https://encrypted-tbn3.gstatic
.com/images?q=tbn:ANd9GcRD_-XQ2eXeyZO4376htFF_XwwznCLHyRZbc
EvrERpG29OvvX6o
My Study Life
Stuff
Graph
Replit - Home
Home Page - Curso...
Other favorites
ome to CineApp
Just a Simple Footer: u'know for space
ENG
10:20 pm
11/04/2021

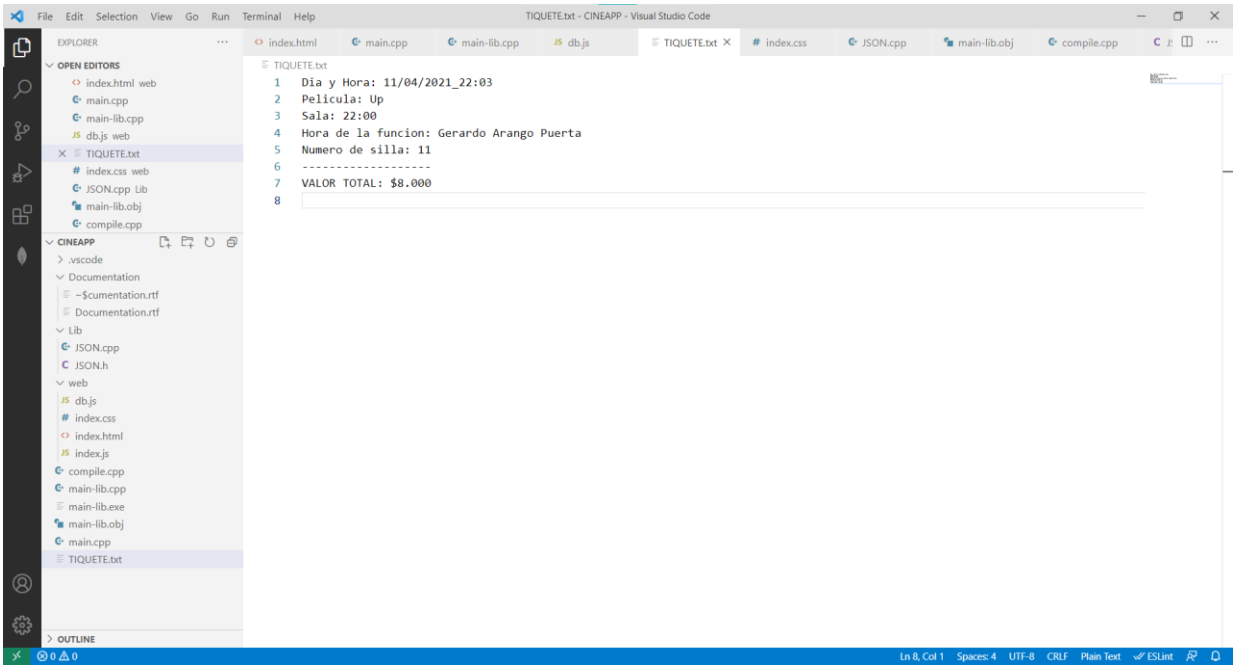

CineApp
Developer Command Prompt for VS 2019 - main-lib.exe
CineApp
CREATED BY
Jhoseph Samirt Lizarazo
Jaider David Daza Pardo
David Alejandro Sequera
Kevin Leonardo Lopez Garcia
Welcome to CineApp
1. Cosult Movies/Show Times/Rooms/Seats
2. Edit Movies/Show Times/Rooms/Seats
3. Sell Ticket
4. Change price
5. Exit
Choose an option:
Stuff
Graph
Replit - Home
Home Page - Curso...
Other favorites
e to CineApp
Just a Simple Footer: u'know for space
ENG
10:21 pm
11/04/2021

5. Horarios

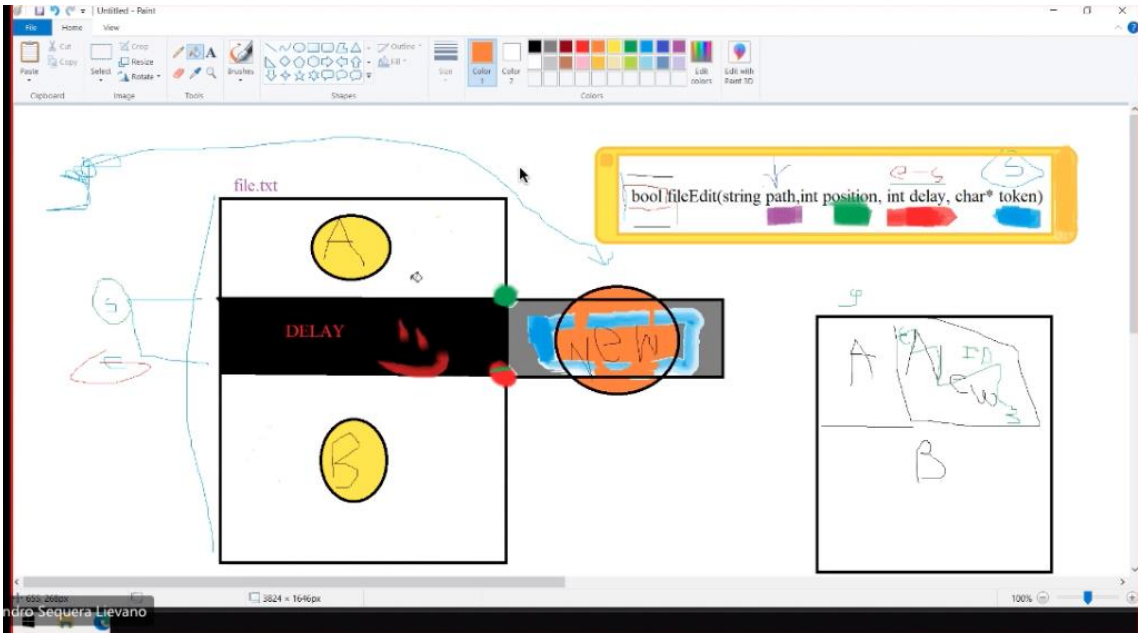


6. Venta de tickets





Extra: Ilustración didáctica de la función “fileEdit” en paint



Cronograma:

1. Aprender lo básico de HTML (Insertar una Imagen)
2. Aprender lo básico de CSS (clases, id)
3. Comparación sintáctica de C++ con JS
4. Introducción a Objetos de JavaScript (JSON)
5. Modelo de la estructura del Cine (requerimientos)
6. Modelo de la librería. (tipo objeto, conexión con archivo)

7. Funciones del programa. (interfaces y peticiones)
8. Funciones de la librería. (manejo de archivo)

Bibliografía:

C++:

- auto (C++) | Microsoft Docs
- Walkthrough: Create and use a static library (C++) | Microsoft Docs
- std::optional - cppreference.com
- c++ - Function with a custom return type and the "false" return conditions? - Stack Overflow
- <regex> | Microsoft Docs
- <regex> - C++ Reference (cplusplus.com)
- Range-based for Statement (C++) | Microsoft Docs

HTML, CSS, JavaScript:

- Conceptos básicos de HTML - Aprende sobre desarrollo web | MDN (mozilla.org)
- CSS básico - Aprende sobre desarrollo web | MDN (mozilla.org)
- Fundamentos de JavaScript - Aprende sobre desarrollo web | MDN (mozilla.org)