Universidad Rafael Landívar

Facultad de Ingeniería

Campus Central

Estructura De Datos I

Ingeniera Alba Rosina Sagastume

**PROYECTO#1**

Guaflix

* Carlos Andrés Morales Lara

Carnet: 1171316

* Iván Andrés Arango Saucedo

Carnet: 1158116

Guatemala de la Asunción 18 de abril del 2018

**INTRODUCCIÓN**

Como proyecto#1 de estructura de datos I tenemos que realizar una integridad de una aplicación, donde hacemos una simulación de desarrollar una aplicación prototipo de servicio de transmisión de contenido (series, películas y documentales) vía internet, para nuestra empresa Albavisión. Por eso proveeremos un desarrollo de aplicación Web en Microsoft MVC para que los usuarios de Albavisión puedan tener su propio servicio de transmisión de contenido en línea (Guaflix) y que los usuarios puedan crear su usuario en esta aplicación para poder disfrutar del contenido que ofrecerá el administrador de Guaflix. Los objetivos de este proyecto principalmente se basan en que apliquemos nuestros conocimientos aprendidos durante todo el curso de Estructura de Datos I en una aplicación de software, además de brindar soluciones como un ingeniero en informática y sistemas para solucionar el problema que nos den de la manera más óptima y eficiente posible.

Como solución se tendrá un proceso de inicio de sesión (log in) distribuido en varias carpetas internas de la aplicación ya sea para el usuario de administrador o un usuario-cliente, se tendrá un control de inicio de sesión almacenados en disco en un archivo JSON para tener un control estricto, detallado y controlado de quien inicia sesión además de tener un control de los usuarios creados, dicho control solo es controlado por el usuario admin.

Si el administrador inicia sesión el será el único que puede agregar las películas a la base de datos de Guaflix (catálogo general) y también eliminarlas, además es el único que puede agregar los Json de Películas y los Json de Usuarios.

Si el usuario administrador ya termino de ingresar o eliminar elementos puede cerrar sesión y un usuario-cliente puede iniciar sesión para crear su watchlist o para buscar un elemento del catálogo general y si dicho usuario no fue agregado por el administrador por el Json de Usuarios entonces puede crear su usuario por medio de una interfaz donde sus datos se guardarán al Json de Usuarios serializando su contenido. Si el usuario termino de crear, agregar o eliminar elementos de su watchlist puede cerrar sesión para que otro usuario disfrute del contenido de Guaflix.

**ANÁLISIS**

*ENTRADAS:*

* Archivos Json de Películas y Usuarios.
* Ingreso de Películas y Usuarios por medio de una interfaz.

*SALIDAS:*

* name.showtree: year.showtree: gender.showtree: (Árbol de series por nombre, genero, año)
* name.movietree: year.movietree: gender.movietree: (Árbol de películas por nombre, genero, año)
* name.documentarytree: year.documentarytree: gender.documentarytree: (Árbol de documentales por nombre, genero, año)
* users.tree: Árbol de usuarios
* <username>.watchlist: (Listado del usuario de sus shows, películas o documentales)

*PROCESOS:*

* Inicio de Sesión por administrador.
* Ingreso de archivos Json de Películas y Usuarios.
* Ingreso manualmente de películas o eliminación de las mismas.
* Creación de usuario-cliente o inicio de sesión de usuario-cliente previamente creado por el Json Usuarios agregado por el usuario administrador.
* El usuario agrega películas, shows, documentales (que estén en el catálogo general de Guaflix) que el usuario le gusten a su watchlist.
* Luego de que el usuario tenga ya elementos en su watchlist el usuario puede ver su watchlist y si ya no desea tener un elemento lo puede eliminar de su watchlist.
* Si el catálogo general tiene muchos elementos y el usuario quiere ver si una película está o no en el catálogo entonces el usuario puede buscar ese elemento solamente al ingresar el nombre del elemento y el sistema le dirá si está o no dicho elemento buscado en el catálogo general.
* Si el administrador o usuario-cliente ya no desea hacer otro proceso, esté puede cerrar sesión.

*RESTRICCIONES:*

* El usuario y contraseña debe ser válido para el inicio de sesión para entrar a la aplicación de manejo de películas, series, shows, esta restricción es para el administrador y usuario-cliente.
* No se puede crear un usuario si tienen el mismo username.
* Si se quiere agregar un elemento del catálogo general, pero si dicho elemento ya se encuentra en el catálogo general ya no se agregará ese elemento.
* Si se quiere eliminar un elemento del catálogo general, pero si dicho elemento no se escribe tal y como esta mostrado en el catálogo general el sistema no eliminará el elemento.
* Si el usuario/administrador en uno de los campos de texto para ingresar elementos deja uno blanco o en ninguno escribe nada el sistema le dirá que debe llenar todos los campos de texto para poder continuar.

**DISEÑO**

*PSEUDOCÓDIGO*

Función: ***LogIn***

Entorno: usuario, contraseña

Algoritmo:

INICIO

Escribir usuario y contraseña

Leer usuario y contraseña

SI usuario no existe en @"C:\Proyecto1\Users.tree" ir a Función CrearUsuario

SINO

Verificar SI usuario existe en @"C:\Proyecto1\Users.tree"

SI usuario existe y la contraseña / usuario es incorrecto

Escribir (“Usuario invalida”)

SINO

return RedirectToAction ("VisualizarCatalogo", "Peliculas");

FIN SI

FIN

Función: ***CrearUsuarioManual***

Entorno: NuevoUsuario.Nombre, NuevoUsuario.Apellido; NuevoUsuario.Edad, NuevoUsuario.UserName, NuevoUsuario.Password, NuevoUsuario.ConfirmPassword

Algoritmo:

INICIO

Escribir NuevoUsuario.Nombre, NuevoUsuario.Apellido; NuevoUsuario.Edad, NuevoUsuario.UserName, NuevoUsuario.Password, NuevoUsuario.ConfirmPassword

Leer NuevoUsuario.Nombre, NuevoUsuario.Apellido; NuevoUsuario.Edad, NuevoUsuario.UserName, NuevoUsuario.Password, NuevoUsuario.ConfirmPassword

SI NuevoUsuario.UserName == listaNombreUsuario.ElementAt(i)

Escribir (“Usuario ya existente, pruebe uno diferente.”)

SINO

SI NuevoUsuario.Contraseña == NuevoUsuario.ConfirmPassword

Escribir (“Usuario creado exitosamente”)

JsonConvert.SerializeObject(NuevoUsuario)

Ingresar NuevoUsuario al ArbolB

FIN

Función: ***AgregarPeliculasManual***

Entorno: agregar.Tipo, agregar.Nombre, agregar.AñoLanzamiento, agregar.Genero

Algoritmo:

INICIO

Escribir agregar.Tipo, agregar.Nombre, agregar.AñoLanzamiento, agregar.Genero

Leer agregar.Tipo, agregar.Nombre, agregar.AñoLanzamiento, agregar.Genero

SI agregar.Tipo != “documental” || “show” || “película”

Escribir (“Tipo agregado no valido, ingrese de nuevo”)

SINO

Verificar si agregar existe en @"C:\Proyecto1\Peliculas.tree"

SI agregar.Nombre == listaNombre.ElementAt(i)

Escribir (“Pelicula ya existente, ingrese una diferente”)

SINO

JsonConvert.SerializeObject(agregar)

Ingresar agregar al ArbolBPlus

FIN

Función: ***AgregarPeliculasJson***

Entorno: Archivo Json de Peliculas

Algoritmo:

INICIO

Seleccionar un archivo Json de Peliculas

Leer archivo Json de Peliculas

Peliculas peliculas = JsonConvert.DeserializeObject<Peliculas>(Json.ElementAt(i));

Agregar valores del Json Peliculas al ArbolBPlus

FIN

Función: ***AgregarUsuariosJson***

Entorno: Archivo Json de Usuarios

Algoritmo:

INICIO

Seleccionar un archivo Json de Usuarios

Leer archivo Json de Usuarios

CrearUsuario usuarios = JsonConvert.DeserializeObject<CrearUsuario>(JsonUsers.ElementAt(i));

Agregar valores del Json Usuarios al ArbolBPlus

FIN

Función: ***CreaciónDeArchivos***

Entorno: Name<Tipo>tree, Year<Tipo>tree, Gender<Tipo>tree

Algoritmo:

INICIO

Si el administrador ingreso una nueva película con Tipo correcto se creará Name<Tipo>tree, Year<Tipo>tree, Gender<Tipo>tree en @" C:\Proyecto1\Name.<Tipo>tree";

En este se almacenarán los elementos que tengan el mismo tipo según su tree, si es Name.tree se agregarán todos los nombres de todos los elementos de ese tipo seleccionado.

Si es Gender.tree se agregarán todos los géneros de todos los elementos de ese tipo seleccionado.

Si es Year.tree se agregarán todos los años de todos los elementos de ese tipo seleccionado.

FIN

Función: ***BuscarPelicula***

Entorno: ArbolBPlus.Buscar(), BuscarPelicula

Algoritmo:

INICIO

En el textBox del MVC se pondrá la película a buscar

SI ArbolBPlus.Buscar(textBox) == true

Escribir (“La película <textBox> SI se ha encontrado en la librería de Guaflix”)

SINO

Escribir (“La película <textBox> NO se ha encontrado en la librería de Guaflix”)

FIN

Función: ***EliminarPelicula***

Entorno: EliminarPelicula(), ArbolBPlus.eliminar()

Algoritmo:

INICIO

En el textBox del MVC se pondrá la película a eliminar

SI eliminar == listaNombre.ElementAt(i))

JsonPeliculas.Remove(JsonPeliculas.ElementAt(posicion));

arbolBPlusPeliculas.Eliminar(eliminar);

Escribir (“Pelicula eliminada correctamente”)

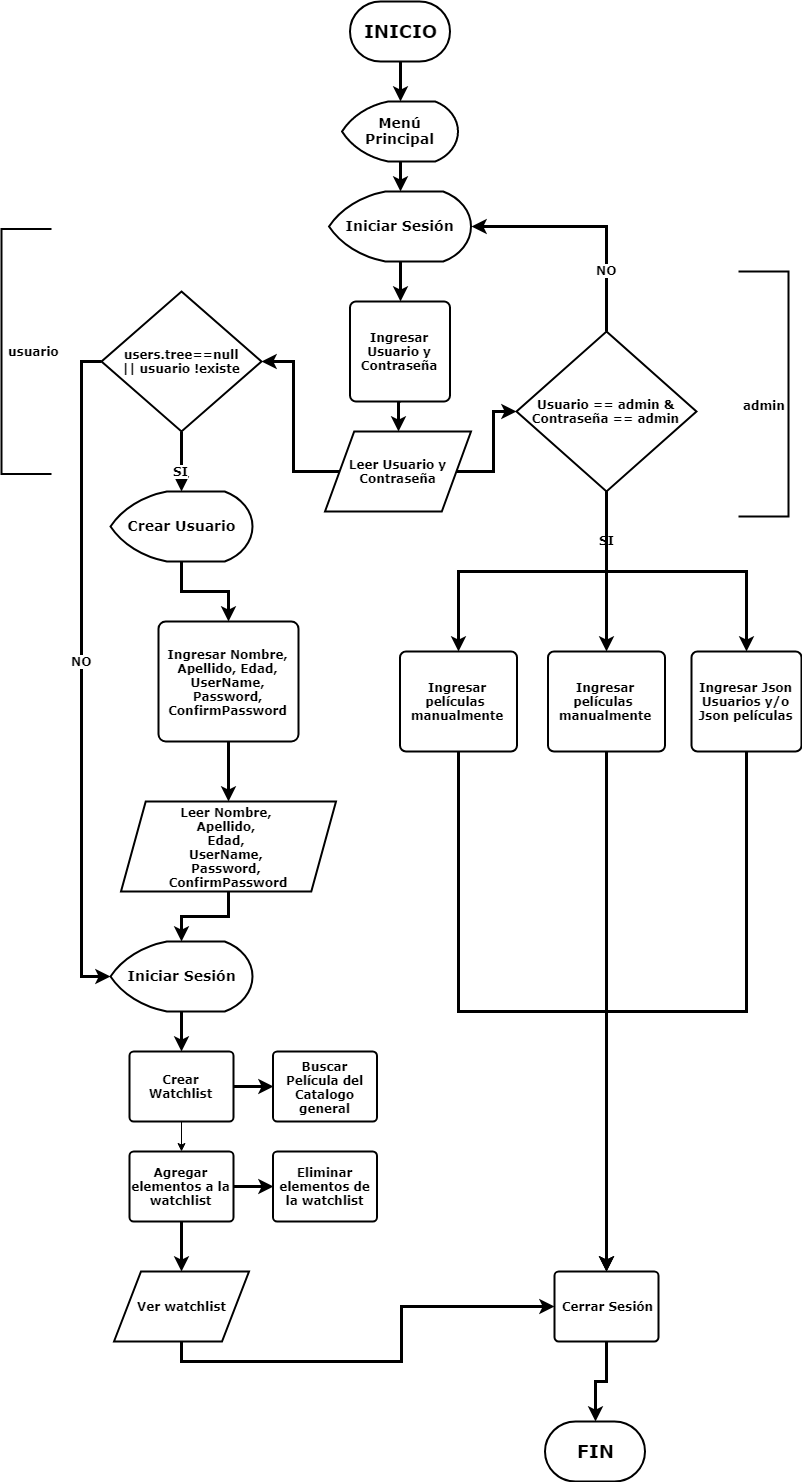
SINO

Escribir (“No se ha podido borrar la película debido a que no existe en el sistema”)

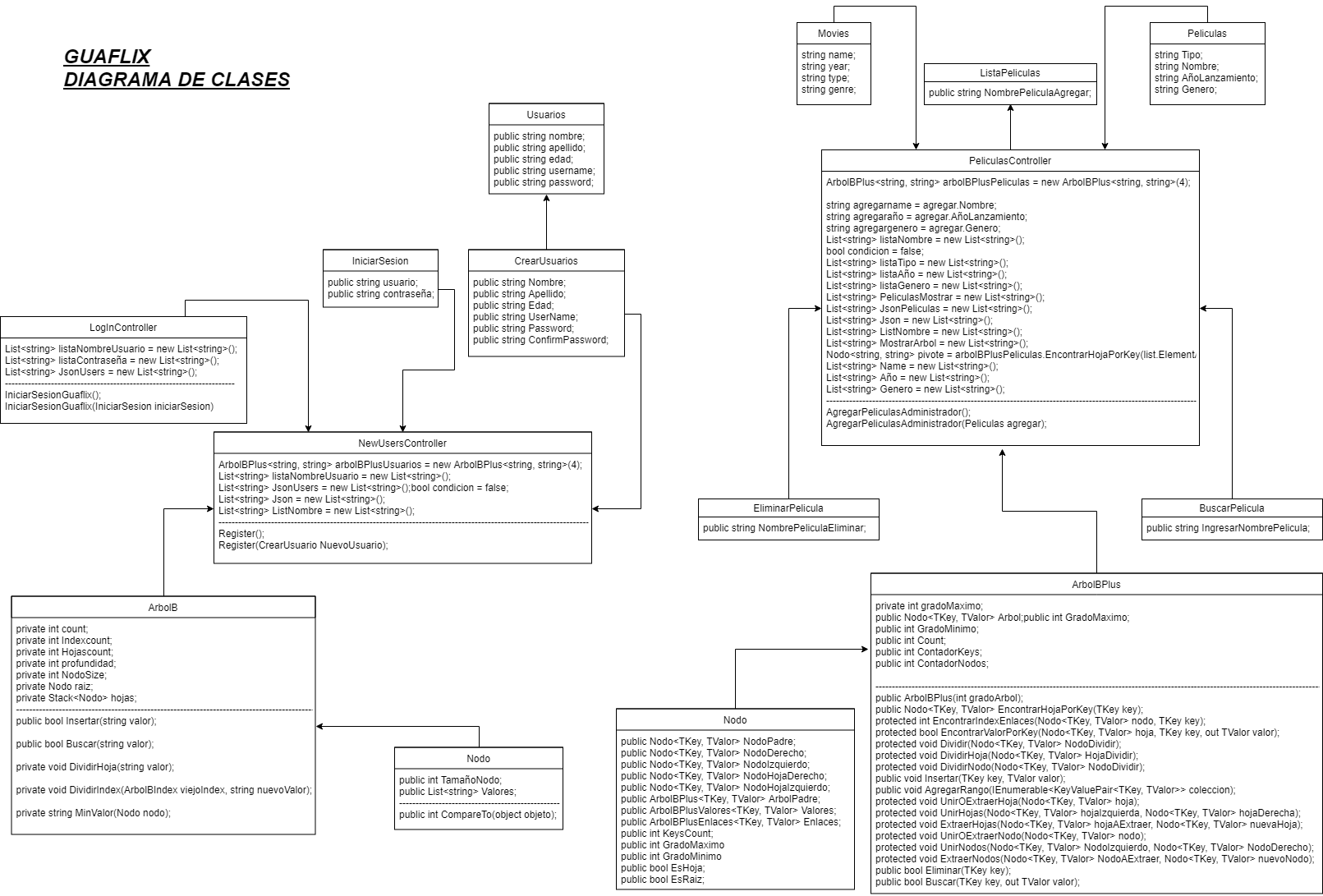
FIN

*DIAGRAMA DE FLUJO*

Diagrama de flujo adjuntado como archivo PNG.



*IDENTIFICACIÓN DE CLASES*

Diagrama de clases adjuntado como imagen PNG.

*DISEÑO DE PANTALLA- MANUAL DE USUARIO*

|  |  |
| --- | --- |
| Menú Principal   * En la primera de la página web contendrá los nombres de los creadores de la página, y abajo un botón el cual lo redirigirá a una vista en la cual podrá iniciar sesión como administrador o usuario.   **Iniciar Sesión**   * Esta vista la permitirá poder iniciar sesión como administrador o usuario y poder crear usuarios.   **Crear usuario**   * Esta vista la permitirá al usuario poder crear su usuario siempre y cuando cumplas con las siguientes condiciones:  1. Usuario no existe en la base de datos de usuarios. 2. Al ingresar sus contraseñas coincidan. 3. Que llene todos los campos que se le solicitan.  * Al cumplir con las condiciones solicitas el usuario podrá tener iniciar sesión a Guaflix.   **Perfil de Administrador**   * En esta vista el administrador tendrá el acceso de poder realizar cambios en la página de Guaflix, teniendo el fácil acceso de las siguientes vistas por medio de botones que lo redireccionaran a las vistas para agregar películas o usuarios y eliminar películas.   **Agregar películas de manera manual**   * Esta vista permite al usuario agregar películas a la base de datos de Guaflix, cumpliendo con llenar la siguiente información:  1. Tipo los cuales pueden ser documental, película o show. Cabe aclarar que si desea poner otro tipo la página no le permitirá agregar películas si no son los 3 mencionados antes. 2. Nombre de la película, la página web validará antes de agregar si existe una película con el mismo nombre, si existe una igual deberá agregar una con distinto nombre. 3. Año de lanzamiento, deberá definir el año en que fue estrenada la película, show o documental. 4. Género, se debe aclarar si es comedia, acción, ciencia ficción o al género al que pertenezca.   **Agregar películas por medio de archivo Json**   * Esta vista le permitirá al administrador agregar películas de manera más rápida por medio de un archivo Json que contenga los siguientes campos: name, year, type y genre.   **Eliminar películas**   * El administrador podrá eliminar películas agregadas previamente a la base de datos de Guaflix, cabe aclarar que si no existe la película que el administrador desea eliminar la página le notificará que no existe.   **Agregar usuarios por medio de archivo Json**   * Tanto como crear de manera manual usuarios en la página principal, el administrador tiene el acceso de agregar usuarios por medio de un archivo Json.     **Acceso de información de base de datos de Guaflix**   * Al agregar usuarios y películas o eliminar películas, quedará un registro en su disco local C, el cual usted como administrador podrá ver la información de manera ordena por un árbol B+, el cual contiene toda la información recolecta en formato Json, el cual es leída constantemente por la página web para poder llevar un registro actualizado de cada acción que se realiza en la página, tanto como el administrador o usuario. Dichas carpetas contienen:  1. Usuarios 2. Películas generales 3. Películas clasificadas por su tipo, año, género y nombre. 4. WatchList de los usuarios   **Menú principal para usuarios de la plataforma de Guaflix**   * El usuario contara con diversas opciones la cuales son: crear su lista y agregar películas a dicha lista, eliminar películas de su lista, ver su lista de películas, buscar películas en el catálogo de Guaflix y cerrar sesión.   **Crear lista y agregar películas a dicha lista**   * En esta vista el usuario podrá crear su lista y agregar películas a su lista siempre y cuando se encuentre disponible en el catálogo de Guaflix, este no aceptará elementos duplicados lo cual los descartará y solo se quedará con una copia de la película.   **Buscar películas**   * Esta vista le permitirá al usuario poder consultar si existe alguna película que desee agregar a su lista, y le notificará por medio de un mensaje si esta existe o no.   **Eliminar películas de la lista del usuario**   * Esta vista le permitirá al usuario poder eliminar películas de su watchlist.   **Visualizar lista de usuario**   * Esta vista le permitirá al usuario poder ver su lista actualizada con sus cambios realizados al eliminar o agregar películas.   **Cerrar sesión**   * Tanto el administrador como el usuario tendrán una opción en sus plataformas de cerrar sesión la cual los redireccionara a la página inicial para iniciar sesión. |  |

*CONCLUSIONES*

* Que los estudiantes de estructura de Datos I puedan ser capaces de interactuar con archivos Json para poder administrar su información de manera más organizada y fácil de utilizar.
* Ser capaces de crear estructuras de datos no lineales (ArbolB y ArbolB+).
* Ser capaces de poder utilizar páginas web y dominar el uso de controladores, vista y modelos (MVC).

*BIBLIOGRAFÍA*

* https://www.c-sharpcorner.com/article/json-serialization-and-deserialization-in-c-sharp/
* https://stackoverflow.com/questions/7524903/should-i-call-close-or-dispose-for-stream-objects
* https://stackoverflow.com/questions/227624/asp-net-mvc-controller-actions-that-return-json-or-partial-html
* https://stackoverflow.com/questions/28471103/how-to-call-controller-from-the-button-click-in-asp-net-mvc-4/28471202
* https://stackoverflow.com/questions/18355653/how-can-i-open-the-text-file-in-my-mvc-application