Manual de Técnico

Kelly Mischel Herrera Espino
UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Introducción:

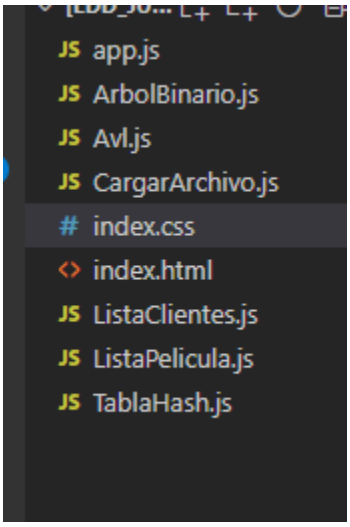
En el siguiente manual se explican a detalle las clases, métodos, funciones y estructuras de datos, utililizadas para el desarrallo optimo del programa. Se utilizaron estructuras lineales y no lineales. Esto con el fin de poder almacenar distintos datos que posteriormente serian utilizados en las distintas funcionalidades del programa.

Estructuras de Datos

- **Árbol binario de búsqueda:** para guardar los datos de los actores.
- **Árbol avl:** se almacenaron las películas.
- **Tabla Hash:** Se guardaron las categorías.
- **Lista simple:** Se almacenaron los datos de los clientes.

Clases

- **App.js**
- **Actor.js**
- **NodoArbolBinario.js**
- **ArbolBinario.js**
- **Película.js**
- **NodoAvl.js**
- **ArbolAvl.js**
- **CargarArchivo.js**
- **Cliente.js**
- **NodoCliente.js**
- **ListaCliente.js**
- **Categoria.js**
- **NodoLista.js**
- **ListaSimple.js**



Clase app

Esta clase contiene variois métodos, los cuales sirven para poder ocultar y mostrar etiquetas en el html. También se hace la comprobación de credenciales de los usuarios.

```
class App {
    static mostrarPelis;
    static mostrarPelis2;

    comprobarUsuario() {
        let nombre = document.getElementById("user").value;
        let password = document.getElementById("password").value;
        let checbox=document.getElementById("administrador").checked

        console.log("esto tiene "+(checbox))
        if (checbox) {

            if (nombre  == "EDD" && password == "123") {
                alert("Bienvenido admin")
                this.mostraPaginaAdmin()


            } else {

                if (CargarArchivo.listaCliente.buscarUsuario(nombre,password)) {
                    alert("Bienvenido "+nombre)
                    this.mostrarUsuarios()

                }
                else{
                    alert("Usuario o contraseña incorrectos")

                }

            }
```

Métodos

ComprarobarUsuario:

Con varios if anidados se comprueban los datos ingresados en el login, se manda a llamar al método buscarUsuario de la clase ListaCliente, en el cual se ingresan como parámetro la contraseña y el usuario. Si las credenciales coinciden, se retorna true y se procede a mostrar la pagina del admin o del usuario. De no ser así se muestra un mensaje de credenciales incorrectas.

```
comprobarUsuario() {
    let nombre = document.getElementById("user").value;
    let password = document.getElementById("password").value;
    let checbox=document.getElementById("administrador").checked

    console.log("esto tiene "+(checbox))
    if (checbox) {

        if (nombre  == "EDD" && password == "123") {
            alert("Bienvenido admin")
            this.mostraPaginaAdmin()


        } else {

            if (CargarArchivo.listaCliente.buscarUsuario(nombre,password)) {
                alert("Bienvenido "+nombre)
                this.mostrarUsuarios()

            }
            else{
                alert("Usuario o contraseña incorrectos")

            }
```

mostrarUsuario:

Muestra en el html las opciones para el usuario

```
mostrarUsuarios(){
    let opcionesUsuario =document.getElementById("opciones
    let cerrar = document.getElementById("cerrar");
    let admin = document.getElementById("login");
    let info = document.getElementById("info")
    let linea = document.getElementById("linea")



    admin.style.display="none"
    info.style.display="block"
    opcionesUsuario.style.display="block"
    linea.style.display="block"

    cerrar.style.display="inline"
}
```

mostrarLogin:

Se muestra al iniciar el programa el Login en donde el usuario debe de escribir sus credenciales.

```
mostrarLogin() {


    let admin = document.getElementById("login");
    let principalAdmin = document.getElementById("opcionesAdmin")
    let cargarCategorias=document.getElementById("cargar_Categorias")
    let botonesP=document.getElementById("botonesPelicula")

    let principalG = document.getElementById("opcionesGenerales");
    let cerrar = document.getElementById("cerrar");
    let vista_Imagen = document.getElementById("vista_Imagen")
    let opcionesUsuario =document.getElementById("opcionesUsuario")
    let botones=document.getElementById("botonesActores")
    if(CargarArchivo.div_peliculas!=null){

        CargarArchivo.div_peliculas.style.display="none"
    }
    if (CargarArchivo.div_categoria!=null) {
        CargarArchivo.div_categoria.style.display="none"
    }
    if (CargarArchivo.div_actores!=null) {
        CargarArchivo.div_actores.style.display="none"
    }
    if ( App.mostrarPelis!=null) {
        App.mostrarPelis.style.display="none"
    }
```

mostrarPaginaAdmin:

Se muestran las opciones para el administrador y se ocultan las etiquetas que no responden a esta vista.

```
mostraPaginaAdmin() {
    let principalAdmin = document.getElementById("opcionesAdmin")
    let admin = document.getElementById("login");
    let principalG = document.getElementById("opcionesGenerales");
    let cerrar = document.getElementById("cerrar");
    let iniciar = document.getElementById("iniciar")
    let linea = document.getElementById("linea")
    let info = document.getElementById("info")
    let cargar_peliculas = document.getElementById("cargar_peliculas")
    let vista_Imagen = document.getElementById("vista_Imagen")

    //_________
    admin.style.display = "none"
    principalG.style.display = "none"
    cargar_peliculas.style.display = "none"
    // iniciar.style.display="none"

    vista_Imagen.style.display = "flex"
    vista_Imagen.style.justifyContent="space-between"
    info.style.display = "block"
    linea.style.display = "block"
    cerrar.style.display = "inline"
    principalAdmin.style.display = "block"
}
```

CargarPeliculas:

Se muestran la etiqueta input file que corresponde a la carga del archivo de películas.

```
CargarPeliculas() {
    let linea = document.getElementById("linea")

    let cargar_peliculas = document.getElementById("cargar_peliculas")
    let cargarClientes=document.getElementById("cargar_clientes")
    let cargarCategorias=document.getElementById("cargar_Categorias")


    let cerrar = document.getElementById("cerrar");

    cargar_peliculas.style.display = "block"
    cargarClientes.style.display="none"
    cargarCategorias.style.display="none"
    linea.style.display="block"
    cerrar.style.display="inline"
}


CargarClientes() {
```

CargarClientes:

Se muestran la etiqueta input file que corresponde a la carga del archivo de Clientes.

```
CargarClientes() {
    let cargar_peliculas = document.getElementById("cargar_peliculas")
    let cargar_cliente = document.getElementById("cargar_clientes")
    let cargarActores = document.getElementById("cargar_Actores")
    let cargarCategorias=document.getElementById("cargar_Categorias")


//_______________
    cargar_peliculas.style.display = "none"
    cargarActores.style.display="none"
    cargarCategorias.style.display="none"
    cargar_cliente.style.display = "block"
}
```

CargarActores:

Se muestran la etiqueta input file que corresponde a la carga del archivo de Actores.

```
CargarAutores() {
    let cargarActores = document.getElementById("cargar_Actores")
    let cargar_peliculas = document.getElementById("cargar_peliculas")

    let cargarClientes=document.getElementById("cargar_clientes")
    let cargarCategorias=document.getElementById("cargar_Categorias")

    //_______
    cargar_peliculas.style.display="none"
    cargarCategorias.style.display="none"
    cargarClientes.style.display="none"
    cargarActores.style.display = "block"
}
```

CargarCategorias:

Se muestran la etiqueta input file que corresponde a la carga del archivo de Categorías.

```
CargarCategoria() {
    let cargarActores = document.getElementById("cargar_Actores")
    let cargar_cliente = document.getElementById("cargar_clientes")
    let cargarCategoria = document.getElementById("cargar_Categorias")
    let cargar_peliculas = document.getElementById("cargar_peliculas")

    //_____________________
    cargar_peliculas.style.display="none"
    cargar_cliente.style.display="none"
    cargarActores.style.display="none"
    cargarCategoria.style.display = "block"
}
```

MostrarPelicula:

Se muestra la etiqueta div para posteriormente poder mostrar el grafo de el árbol avl.

```
mostrarPelicula(){
    let cargarActores = document.getElementById("cargar_Actores")
    let cargarClientes=document.getElementById("cargar_clientes")
    let cargarCategorias=document.getElementById("cargar_Categorias")
    let cargar_peliculas = document.getElementById("cargar_peliculas")


    CargarArchivo.arbol_avl.graficar()
    //_____________________________
    let lienzo=document.getElementById("lienzo")
    //_____________________
    cargar_peliculas.style.display="none"
    cargarCategorias.style.display="none"
    cargarClientes.style.display="none"
    cargarActores.style.display="none"
    lienzo.style.display="block"

}
```

MostrarCategoría:

Se muestra la etiqueta div para posteriormente poder mostrar el grafo de la Tabla Hash.

```
mostrarCategoria(){
    CargarArchivo.tablaHash.grafica1()
    let cargarActores = document.getElementById("cargar_Actores")
    let lienzo=document.getElementById("lienzo")
    let cargarClientes=document.getElementById("cargar_clientes")
    let cargarCategorias=document.getElementById("cargar_Categorias")
    let cargar_peliculas = document.getElementById("cargar_peliculas")


    //__________
    cargar_peliculas.style.display="none"
    cargarCategorias.style.display="none"
    cargarClientes.style.display="none"
    cargarActores.style.display="none"
    lienzo.style.display="block"
}
```

MostrarCliente:

Se muestra la etiqueta div para posteriormente poder mostrar el grafo de la lista simple de clientes.

```
mostrarCliente(){
    CargarArchivo.listaCliente.graficar()
    let cargar_peliculas = document.getElementById("cargar_peliculas")

    let cargarActores = document.getElementById("cargar_Actores")
    let lienzo=document.getElementById("lienzo")
    let cargarClientes=document.getElementById("cargar_clientes")
    let cargarCategorias=document.getElementById("cargar_Categorias")


    //____________
    cargar_peliculas.style.display="none"
    cargarCategorias.style.display="none"
    cargarClientes.style.display="none"
    cargarActores.style.display="none"
    lienzo.style.display="block"
```

MostrarActores:

Se muestra la etiqueta div para posteriormente poder mostrar el grafo del árbol binario de actores.

```
mostrarActores(){
    let cargarActores = document.getElementById("cargar_Actores")
    let cargarClientes=document.getElementById("cargar_clientes")
    let cargarCategorias=document.getElementById("cargar_Categorias")
    let cargar_peliculas = document.getElementById("cargar_peliculas")
    let vista=document.getElementById("mostrarPeli")


    CargarArchivo.arbolBinario.graficar()
    let lienzo=document.getElementById("lienzo")
    //___________-
    vista.style.display="none"
    cargar_peliculas.style.display="none"
    cargarCategorias.style.display="none"
    cargarClientes.style.display="none"
    cargarActores.style.display="none"
    lienzo.style.display="block"
}
```

MostrarPeliculas:

Se muestran las opciones para que se puedan visualizar de forma ascendente y descendente las películas

```
MostrarPeliculas(){
    let botones=document.getElementById("botonesPelicula")
    App.mostrarPelis=document.getElementById("mostrarPeliculas")
    App.mostrarPelis.innerHTML=""
    let vista=document.getElementById("mostrarPeli")
    CargarArchivo.arbol_avl.prer_order(CargarArchivo.arbol_avl.raiz)

    if (CargarArchivo.div_categoria!=null) {
        CargarArchivo.div_categoria.style.display="none"
    }

    if (CargarArchivo.div_peliculas!=null) {
        CargarArchivo.div_peliculas.style.display="none"
    }
    //"text-align:center; display: block
    botones.style.display="block"
    vista.style.textAlign="center"
    vista.style.display="block"
}
```

MostrarBotonesActores:

Opciones para poder visualizar en pre orden, in orden y post orden los actores almacenados en el árbol binario.

```
//Mostrar botones del actores
MostrarBotonesActores(){
    let botones=document.getElementById("botonesActores")
    let botonesP=document.getElementById("botonesPelicula")

    if(CargarArchivo.div_peliculas!=null){

        CargarArchivo.div_peliculas.style.display="none"
    }
    if (CargarArchivo.div_categoria!=null) {
        CargarArchivo.div_categoria.style.display="none"
    }

    if ( App.mostrarPelis!=null) {
        App.mostrarPelis.style.display="none"
    }
    botonesP.style.display="none"
    botones.style.display="block"

}
```

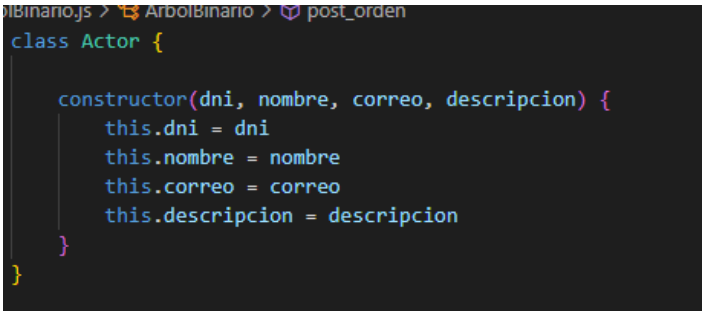
MostrarCategorías:

Muestra las etiquetas divs para poder visualizar las categorías.

```
Mostrar_categorias(){
    let botones=document.getElementById("botonesActores")
    CargarArchivo.div_actores.innerHTML=""
    CargarArchivo.tablaHash.MostrarCategorias()
    botones.style.display="none"
    CargarArchivo.div_categoria.style.display="block"


}
```

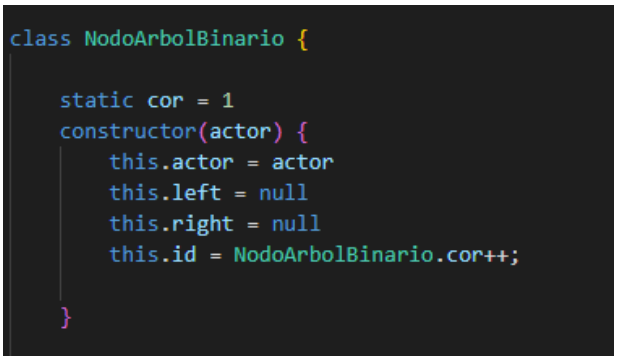
Clase Actor

Contiene un constructor con los datos de un Actor. Con esta clase se crean objetos de tipo Actor.

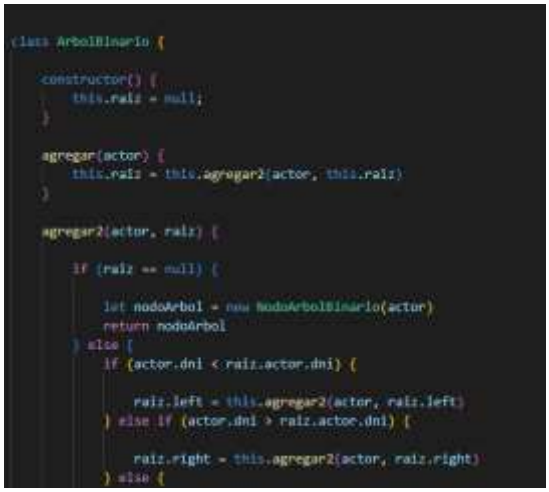


Clase NodoArbolBinario

Tiene un contructor que recibe como parametro un actor y tiene como atributos un id, hijo izquierdo y el hijo derecho, estos son de tipo NodoArbolBinariol.



Clase ArbolBinario



Clase Película

Esta clase tiene como atributos los datos necesarios para poder crear objetos de tipo película.

```
class Pelicula{

    constructor(id,nombre,descripcion,puntuacion,precio,comentario){
        this.id=id
        this.nombre=nombre
        this.descripcion=descripcion
        this.puntuacion=puntuacion
        this.precio=precio
        this.comentario=comentario
    }
}
```

Clase ArbolAvl

Métodos:

- **Altura**
- **Máximo**
- **rotacionDerecha**
- **rotacionIzquierda**
- **balance**
- **insertar**
- **pre_order**
- **comentarios**
- **Ascendente**
- **Descendente**
- **Graficar**

```
class ArbolAvl {

    constructor() {
        this.raiz = null;
    }


    altura(raiz) {
        if (raiz == null) {
            return 0;
        }
        return raiz.altura;
    }

    Maximo(dato1, dato2) {

        return dato1 > dato2 ? dato1 : dato2;
    }
```

Clase CargarArchivo

Métodos:

- **CargarPeliculas**
- **CargarActores**
- **CargarClientes**
- **CargarCategorías**

Clase Cliente

```
class Cliente {

    constructor(dpi, nombre, usuario, correo, contrasenia, telefono) {
        this.dpi = dpi;
        this.nombre = nombre
        this.usuario = usuario
        this.correo = correo
        this.contrasenia = contrasenia
        this.telefono = telefono
    }
}
```

Clase NodoCliente

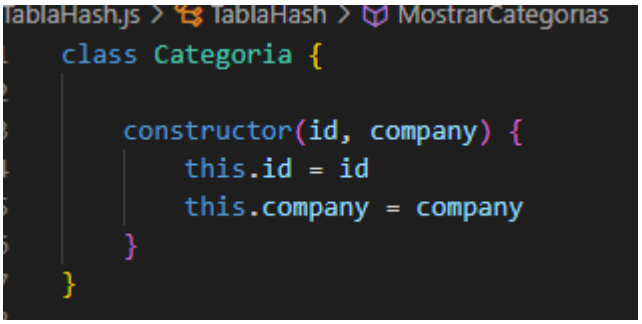
```
class NodoCliente {

    constructor(cliente) {
        this.cliente = cliente
        this.siguiente = null
    }
}
```

ListaCliente

```
class ListaCliente {

    constructor() {
        this.cabeza = null

    }
    insertar(cliente) {
        let nuevo = new NodoCliente(cliente)
        if (this.cabeza == null) {
            this.cabeza = nuevo
        } else {
            let aux = this.cabeza
            while (aux.siguiente != null) {

                aux = aux.siguiente
            }
            aux.siguiente = nuevo
        }
```

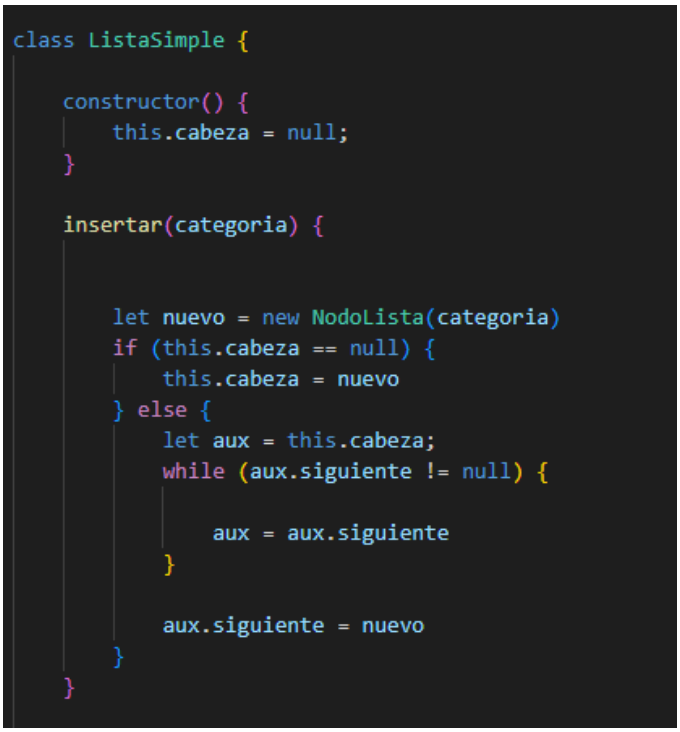
Clase Categoria

Sirve para crear objetos de tipo Categoría, tiene como atributos el id y la compañía.



ListaSimple

Esta lista sirve para poder almacenar los datos en caso de colisiones.



TablaHash

```
}
class TablaHash {

    constructor() {
        this.tamanio = 20;
        this.cabeza = null
        this.factor = 0
        this.id = 0
    }

    crearListas() {
        for (let index = 0; index < 20; index++) {

            let lista = new ListaSimple()

            this.agregarListas(index, lista)
        }

    }

    agregarListas(llave, lista) {
```

Conclusión:

El proyecto se realizó con las estructuras de datos lineales y no lineales. Se explico la funcionalidad y la lógica del programa. Al guardar los datos de forma dinámica con las estructuras se tiene un espacio que se adapta a la cantidad de datos que se vayan almacenando. Caso contrario con los arreglos estáticos