

PROYECTO 2 - MOVIECATS

MANUAL TÉCNICO

FACULTAD DE INGENIERÍA, USAC

JULIO 2022

YENIFER ESTER YOC LARIOS, 201952336

OBJETIVOS

Objetivo General:

Brindar a los clientes de Moviecats, la oportunidad de alquilar vía online las diferentes películas que dicha empresa ofrece, así como mostrar tanto al usuario como al administrador, una vista amigable y fácil de manejar.

Objetivos Específicos:

Presentar una interfaz interactiva, que permita una interacción entre el usuario y el programa.

Implementar las estructuras lineales para la conveniencia del usuario

Generar gráficos que ayuden a apreciar las entradas correspondientes.

Alcances:

Que pueda implementarse el uso y manipulación de las estructuras de datos, así como el muestreo de los datos de manera amigable y entendible.

Especificación Técnica: Lenguaje Python

Requisitos del hardware:

- Un ordenador de escritorio o portátil.

Requisitos del Software:

- Navegador web tales como: Mozilla Firefox, Safari (macOS), Microsoft Edge, Avast Secure Browser, Opera.

LÓGICA DEL PROGRAMA

El programa se realizó por medio del lenguaje JavaScript para implementar estructuras de datos lineales y visualizarlas en la página web.

Las clases creadas fueron las siguientes:

Clase Cliente:

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

Clase Película:

```
class Pelicula{
    constructor(id, nom_pelicula, descripcion, puntuacion, precio, comentarios,
        this.id = id;
        this.nom_pelicula = nom_pelicula;
        this.descripcion = descripcion;
        this.puntuacion = puntuacion;
        this.precio = precio;
        this.comprador = comprador;
        this.comentarios = comentarios;
        this.disponible = disponible
    }
}
```

Clase Actor:

```
class Actor{
    constructor(dni, nombre, correo, descripcion){
        this.dni = dni;
        this.nombre = nombre;
        this.correo = correo;
        this.descripcion = descripcion;
    }
}
```

Clase Categoría:

```
class Categoria{
    constructor(id, company){
        this.id = id;
        this.company = company;
    }
}
```

Clase Comentario:

```
class Comentario{
    constructor(comentario, autor){
        this.comentario = comentario;
        this.autor = autor;
    }
}
```

ESTRUCTURAS:

Lista Clientes:

```
class Nodo{
    constructor(objeto_cliente){
        this.objeto_cliente = objeto_cliente;
        this.siguiente = null;
    }
}

class Lista_clientes{
    constructor(){
        this.head = null;
    }
}
```

Tabla Hash:

```

class Nodo{
    constructor(objeto_valor){
        this.objeto_valor = objeto_valor;
        this.siguiente = null;
        this.anterior = null;
    }
}

class Tabla_Hash{
    constructor(){
        this.head = null;
        this.end = null;
    }
}

```

Lista Categorías:

```

class Nodo{
    constructor(objeto_categoria){
        this.objeto_categoria = objeto_categoria;
        this.siguiente = null;
        this.anterior = null;
    }
}

class Lista_categoria{
    constructor(){
        this.head = null;
        this.end = null;
    }
}

```

Árbol AVL:

```
var contador = 1
class Nodo{
    constructor(objeto_pelicula){
        this.objeto_pelicula = objeto_pelicula;
        this.izquierda = null;
        this.derecha = null;
        this.altura = 0;
        this.id = contador++;
    }
    graficar(){ ...
    }
    getCodigoInterno(){ ...
    }
}

class Arbol_AVL{
    constructor(){
        this.raiz = null;
    }
    MAXIMO(valor1,valor2){
        if(valor1>valor2) return valor1;
        return valor2;
    }
}
```

Árbol Binario:

```
var contador = 1
class Nodo{

    constructor(ob_actor){
        this.ob_actor= ob_actor;
        this.izquierda = null;
        this.derecha = null;
        this.id = contador++;
    }
    graficar(){ ...
    }
    getCodigoInterno(){ ...
    }
}
class Arbol_binario{
    constructor(){
        this.raiz = null;
    }
    insertar(objeto_actor){
        this.raiz = this.add(objeto_actor, this.raiz);
    }
}
```

MAIN:

```
import { Lista_clientes } from "../Estructura_cliente.js";
import { Cliente } from "../Cliente.js";
import { Arbol_binario } from "../Estructuras_actor.js";
import { Arbol_AVL } from "../Estructuras_peli.js";
import { Pelicula } from "../Pelicula.js";
import { Lista_comentarios } from "../Estructura_comentario_p";
import { Actor } from "../Actor.js";
import { Categoria } from "../Categoria.js";
import { Tabla_Hash } from "../Estructura_tabla.js";
import { Lista_categoria } from "../Lista_categorias.js";
import { Valor_hash } from "../Valor_hash.js";
import { Lista_pelis } from "../lista_libros.js";
import { Comentario } from "../Comentario.js";

let user_archive;
let cate_archive;
let usuario_ingresado = 1;
let nombre_us_ing = "";
let contra_ingre;
let peli_archive;
let actor_archive;
let contador_ac = 0;

let lista_us = new Lista_clientes();
let arbol_pelis = new Arbol_AVL();
let arbol_actores = new Arbol_binario();
let tabla_hash = new Tabla_Hash();
```

ANEXOS

DIAGRAMA DE CLASES

