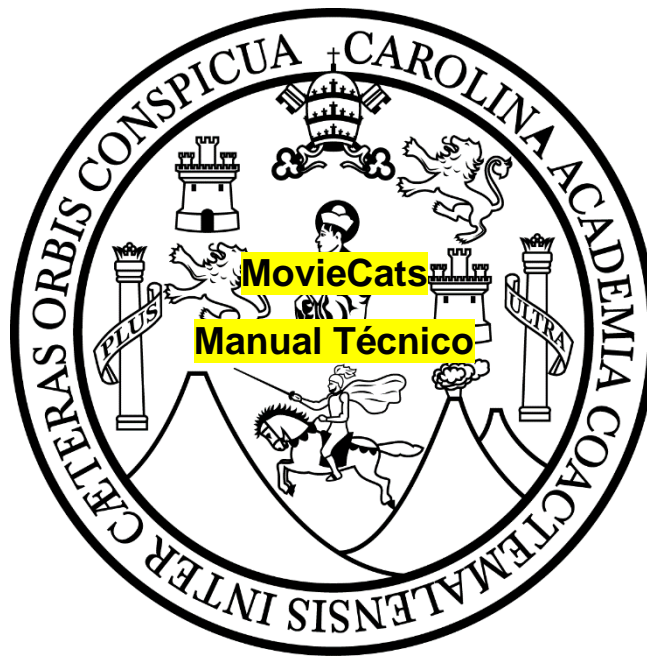


Universidad de San Carlos de Guatemala

Escuela de Ciencias y Sistemas

Estructuras de datos



Luis Mariano Moreira García

202010770

ESPECIFICACIONES

CPU: Intel Pentium

Ram: 2GB

Disco Duro: 200 mb libres

Contar con **javascript** activado en el navegador

OBJETIVOS ESPECÍFICOS

Demostrar el uso correcto de las estructuras de datos de: Lista Simple Enlaza, Árbol Binario de Búsqueda, Árbol AVL, Tabla Hash, Árbol de Merkle.

Manejar el lenguaje de programación de JavaScript, para mostrar la información se utilizó HTML y se mejoró la presentación con CSS.

Aplicar los conocimientos de las estructuras para lograr una máxima eficiencia.

OBJETIVOS GENERALES

Emplear la codificación sha-256.

Entender la importancia de mantener la información segura.

Conocer las posibilidades y límites que tienen las estructuras de datos.

ACERCA DEL PROGRAMA

Para el funcionamiento del programa se debe tomar en cuenta la estructura de los archivos json:

Películas -> AVL

```
[
  {
    "id_pelicula": 6482706815287,
    "nombre_pelicula": "Araceli Evans",
    "descripcion": "Do in pariatur ad proident. Pariatur cupidatat sit cillum cupidatat cupidatat  
minim deserunt sunt. Dolor veniam esse adipisicing ex. Eu deserunt elit sunt irure magna  
voluptate dolor id. Aliquip fugiat laborum ex veniam anim Duis amet ut adipisicing exercitation  
in cupidatat.\r\n",
    "puntuacion_star": 3,
    "precion_Q": 75,
    "paginas": 178,
    "categoria": "Thriller"
  },
  {
    "id_pelicula": 9443300323028,
    "nombre_pelicula": "Howard Hogan",
    "descripcion": "Exercitation non occaecat tempor occaecat culpa ad proident dolor. Ipsum amet  
est consequat nostrud sit qui voluptate culpa consectetur nostrud pariatur eiusmod. Enim deserunt  
ex excepteur ipsum tempor veniam dolore. Labore esse mollit tempor exercitation incididunt  
laborum est nostrud id.\r\n",
    "puntuacion_star": 3,
    "precion_Q": 210,
    "paginas": 169,
    "categoria": "Thriller"
  },
  {
    "id_pelicula": 8225194163684,
    "nombre_pelicula": "Deloris Shelton",
    "descripcion": "Deserunt veniam ut et incididunt exercitation dolor deserunt pariatur dolor  
proident cupidatat. Sunt enim labore commodo ad ex mollit mollit ut nisi aliquip sint. Eu id  
Lorem mollit ea incididunt cillum. Veniam aliquip excepteur eiusmod et qui ex dolor fugiat  
consectetur laborum. Et pariatur tempor cillum ex aute aliquip veniam. Commodo excepteur sit  
occaecat anim fugiat laboris. Dolore laborum ea dolor mollit fugiat pariatur.\r\n",
    "puntuacion_star": 5,
    "precion_Q": 232,
    "paginas": 249,
    "categoria": "Fantasia"
  }
]
```

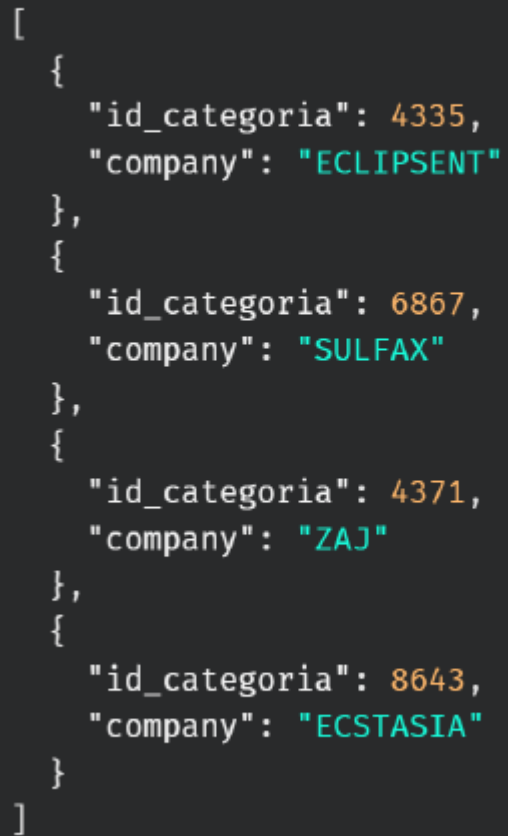
Clientes -> Linked list:

```
[
  {
    "dpi": 3206292060642,
    "nombre_completo": "Riley Shaw",
    "nombre_usuario": "Phillips",
    "correo": "phillipsshaw@knowlysis.com",
    "contrasenia": "anim",
    "telefono": "+502 (943) 527-2850"
  },
  {
    "dpi": 2648015604652,
    "nombre_completo": "Estes Nixon",
    "nombre_usuario": "Alma",
    "correo": "almanixon@knowlysis.com",
    "contrasenia": "pariatur",
    "telefono": "+502 (808) 447-2688"
  },
  {
    "dpi": 1566211594505,
    "nombre_completo": "Kate Meadows",
    "nombre_usuario": "Herring",
    "correo": "herringmeadows@knowlysis.com",
    "contrasenia": "quis",
    "telefono": "+502 (894) 557-2178"
  }
]
```

Actores -> Árbol binario de búsqueda:

```
[
  {
    "dni": 878,
    "nombre_actor": "Poole Gaines",
    "correo": "poolegaines@knowlysis.com",
    "descripcion": "amet"
  },
  {
    "dni": 229,
    "nombre_actor": "Riggs Rosario",
    "correo": "riggsrosario@knowlysis.com",
    "descripcion": "labore"
  },
  {
    "dni": 592,
    "nombre_actor": "Alana Barrett",
    "correo": "alanabarrett@knowlysis.com",
    "descripcion": "commodo"
  },
  {
    "dni": 479,
    "nombre_actor": "Oneil Oneal",
    "correo": "oneiloneal@knowlysis.com",
    "descripcion": "tempor"
  }
]
```

Categorías -> Hash table:



```
[
  {
    "id_categoria": 4335,
    "company": "ECLIPSENT"
  },
  {
    "id_categoria": 6867,
    "company": "SULFAX"
  },
  {
    "id_categoria": 4371,
    "company": "ZAJ"
  },
  {
    "id_categoria": 8643,
    "company": "ECSTASIA"
  }
]
```

EXPLICACIÓN DE LAS CLASES UTILIZADAS:

User_node: Este se encarga del nodo que será almacenado en la lista de clientes (en el código los usuarios son equivalentes a los clientes)

User_list: Se encarga de almacenar los nodos clientes, cuenta con los métodos capaces de iniciar sesión, obtener el cliente que inicio sesión, graficar y mostrar.

Actor_node: Nodo que almacena a los actores del nodo árbol binario.

Actor_binary_search_tree: Arbol binario, capaz de mostrar la información en pre, post e in order.

Commentary_node y commentary_list: No fueron utilizados, su propósito iba a ser el de almacenar todos los comentarios.

Movie_node: Se almacenará las películas, se usa para el avl.

Movie_avl_tree: Árbol binario equilibrado que almacena a las películas, necesita del otro árbol avl (**Ordered_movie_avl_tree**) para desplegar la información en orden alfabético y orden alfabético inverso.

Nodo: Nodo de la lista que ira en la tabla hash.

Lista: Lista enlazada para tratar las colisiones.

Hash_table: Funciona para almacenar las categorías, su única función es completativa.

Data_node: El dato que almacenará el merkle tree.

Hash_node: Para realizar el hash en el merkle tree.

Merkle_tree: Funciona para las transacciones, no se terminó la implementación con blockchain.