Manual Técnico

Christian Alessander Blanco González
202000173
Facultad de Ingeniería de la Universidad de San Carlos de Guatemala

INTRODUCCIÓN

El presente manual técnico tiene como finalidad describir el diseño de dicho proyecto. Esta pagina web fue desarrollada con uno de los lenguajes de programación llamado JavaScript (JS) y con el lenguaje de marcado (HTML). Realizado en el entorno de Visual Studio Code. Dicha pagina web cuenta con una interfaz gráfica que fue realizada a partir de CSS.

También cuenta con un diagrama de clases diseñado por el estudiante donde se podrá visualizar de una mejor forma en la que se desglosó la resolución de dicho programa, con el objetivo de facilitar dicho entendimiento del desarrollo. El motivo de dicha pagina web es poder leer archivos con la extensión json y la implementación de las estructuras de datos lineales vistas en clase.

OBJETIVOS

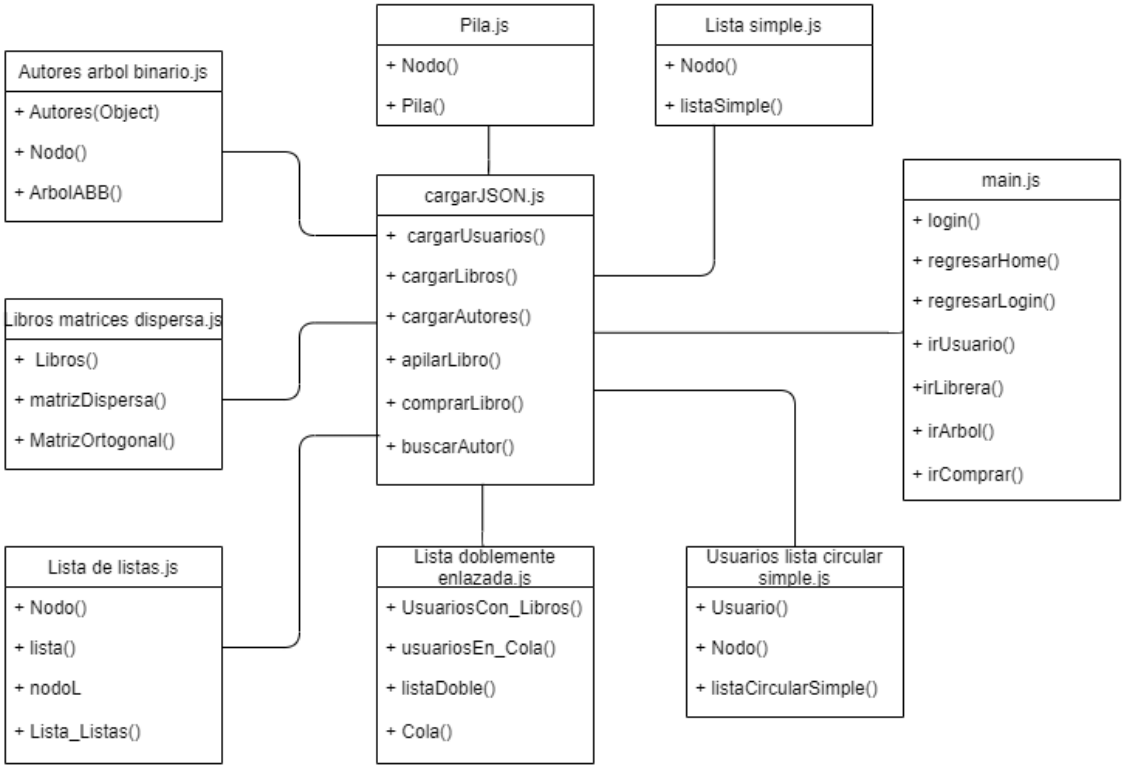
- Demostrar los conocimientos adquiridos sobre estructuras de datos lineales poniéndolos en práctica en una aplicación de simulación.
- Utilizar el lenguaje de JavaScript para implementar estructuras de datos lineales y visualizarlas en una página WEB.
- Definir e implementar algoritmos de búsqueda, recorrido y eliminación.

DESARROLLO DEL PROGRAMA

La aplicación web esta conformada por una interfaz grafica realizada con un archivo de extensión css para poder obtener una mayor amigabilidad con el usuario que utilizará este programa. Este programa consiste en la venta de libros, y así mismo ayudar a la población con obtener los libros digitales.

ESPECIFICACIONES TECNICAS

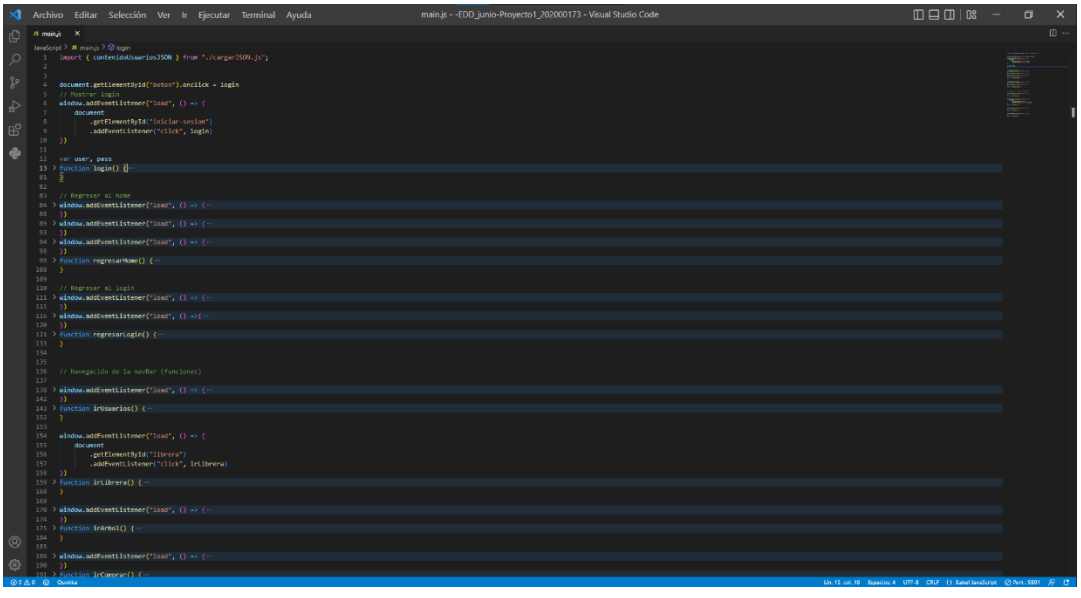
Lenguajes implementados	JavaScript, CSS, HTML
IDE usado	**Ninguno**
Editor de código	**Visual Studio Code**
Sistema Operativo	**Windows 10 (64 bits)**
Diseño web	**CSS**
Librerias implementadas	**Graphviz, d3**



En el diagrama de clases podemos visualizar cada uno de los archivos de JavaScript que fueron trabajados, en la cual podemos ver también cada una de las clases que conforman en cada archivo.

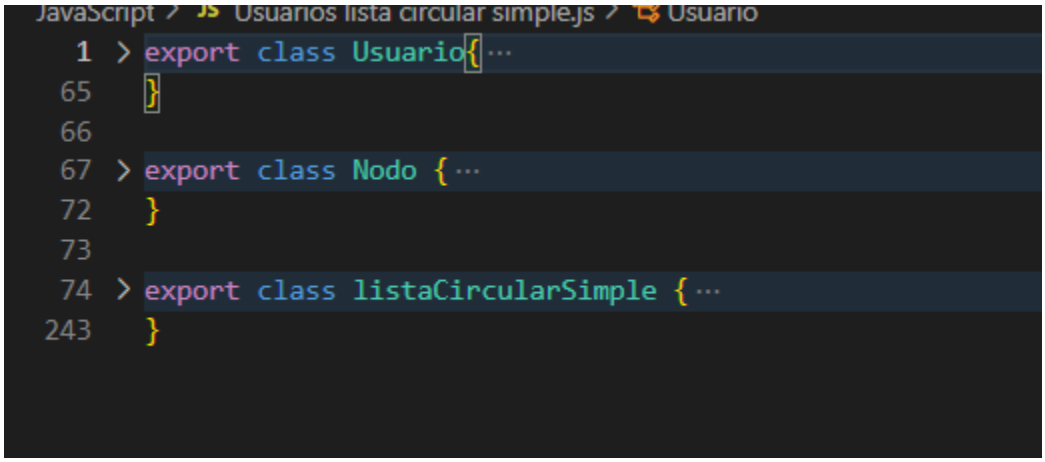
cargarJSON:

es creado para la realización de las cargas masivas, de apilar los libros, comprar libros y de la realización de la búsqueda de los autores. También dentro de las funciones de carga se realizo el envio de datos a los parámetros de los objetos Usuario, Libro y Autores.



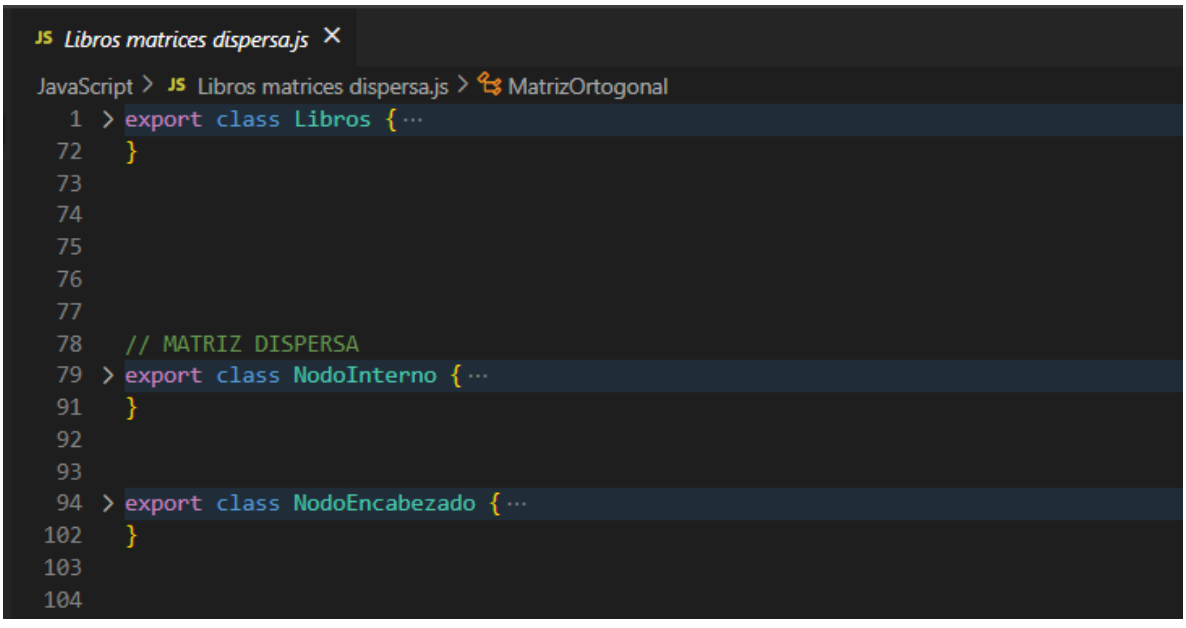
Usuarios lista cirular simple:

Esta clase es la encargada de guardar el objeto Usuarios y poder utilizarlo en alguna cualquier parte de todo el proyecto con solo importarla y tambien esta compuesta por una lista simple circular que es la encargada del recorrido de los objetos Usuario.



Libros matrices dispersas:

Es el encargado de guardar el objeto Libros y poder realizar la implementación de la matriz dispersa y ortogonal. En la cual cada matriz contiene su nodo interno, a diferencia de la matriz dispersa que contiene una lista encabezado para poder llevar un mejor control de lo que se grafica por medio de la librería Graphviz. En comparación de la matriz ortogonal y la dispersa la matriz dispersa en su método de insersion realiza el ordenamiento de una vez y en la matriz ortogonal no realizamos eso.



```
105
106 > export class ListaEncabezado {···
180   }
181
182
183 > export class MatrizDispersa {···
472   }
473
474
475
476
477   /*
478
479       MATRIZ ORTOGONAL
480
481   */
482
483 > class NodoOrtogonal {···
542   }
543
544 > export class MatrizOrtogonal {···
812   }
813
```

Autores árbol binario:

Es la encargada de guardar los datos extraídos del archivo json donde vienen los autores, la cual también se encuentra en ella la implementación del árbol binario de búsqueda, en donde esta compuesta por dos clases que son nodo, y ArbolBB donde nodo puede obtener cualquier tipo de dato y en la clase ÁrbolBB es la encargada de realizar toda la implementación de inserción y graficar el árbol.

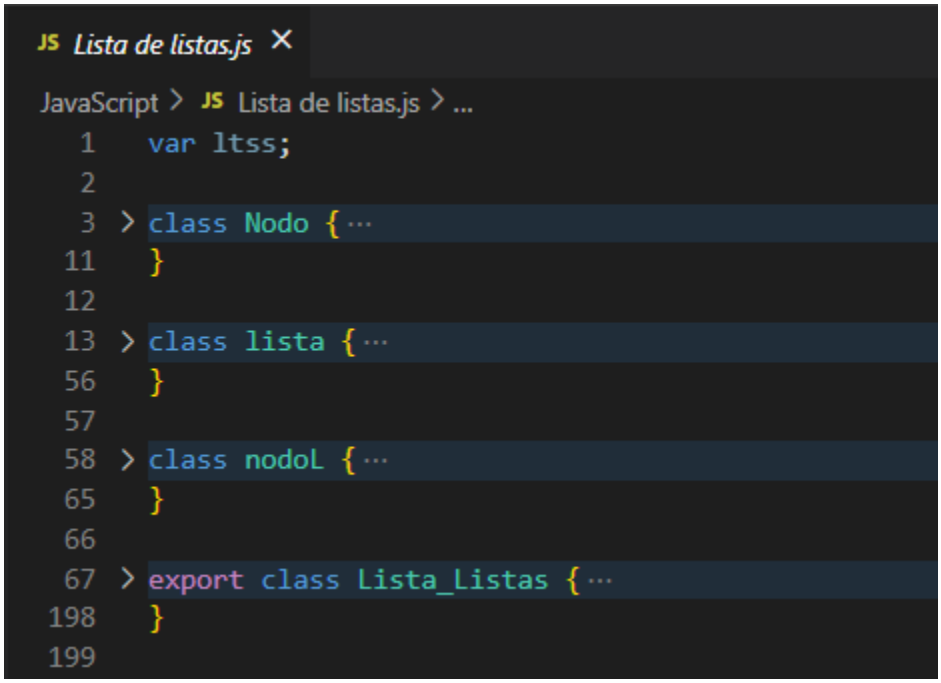
JS Autores arbol binario.js ×

JavaScript > JS Autores arbol binario.js > ArbolABB

```
1   > export class Autores{···
46    }
47
48  > export class Nodo{···
54    }
55
56  > export class ArbolABB{···
144   }
145
```

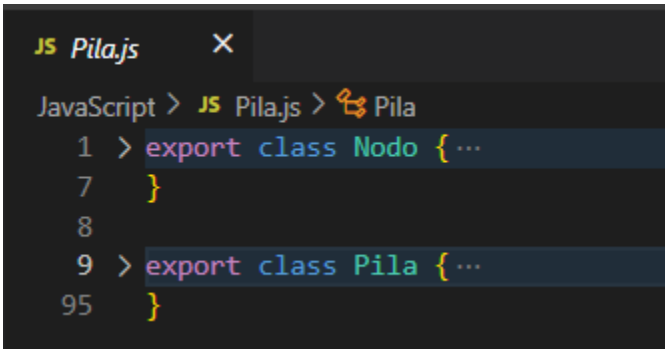
Lista de listas:

Es la encargada de mostrar a cada usuario con cada uno de libros que contiene por medio de listas de listas, donde la lista de los nombres de los usuarios es una lista simple, y la lista de los libros que apunta hacia debajo de la lista de los usuarios es una lista simple. Esta compuesta por dos clases, por dos nodos y dos clases listas, donde cada lista contiene cada uno de sus nodos y en la clase Lista_Listas es la encargada de toda la inserción y de realizar la grafica por graphviz.



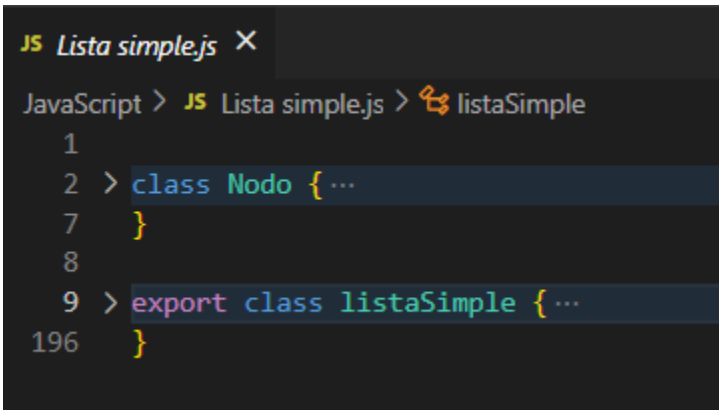
Pila:

Es la encargada de mostrar las existencias de los libros por medio de graphviz ingresando solo el nombre. Esta compuesta por dos clases una que contiene el nodo donde puede recibir cualquier carácter y la clase Pila que es la encargada de apilar y des apilar los libros al venderse.



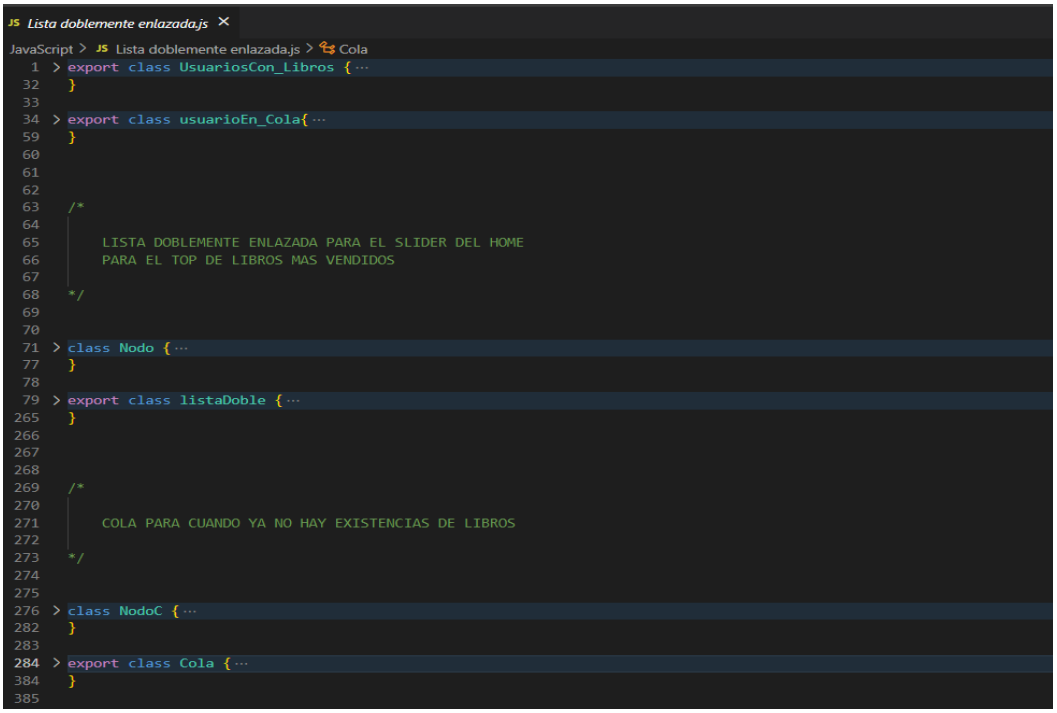
Lista simple:

Esta es la encargada de poder realizar el ordenamiento de los libros utilizando dos algoritmos de ordenamiento siendo estos el ordenamiento burbuja y el ordenamiento Quicksort. Esta compuesta por dos clases una que es el nodo y la clase listaSimple que es la encargada de la inserción, recorrido, ordenar y mostrar los libros ya clasificados y ordenados en la aplicación web.



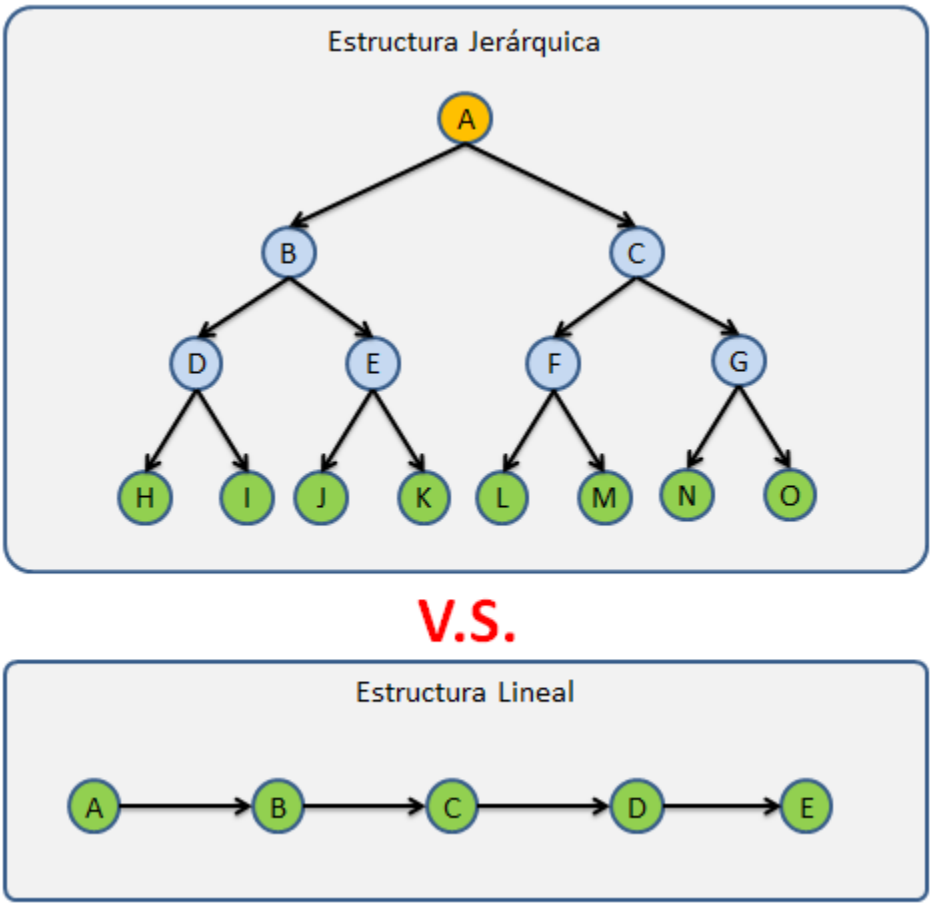
Lista doblemente enlazada:

Esta es la encargada de mostrar el top de usuarios con mas libros de toda la aplicación web y así mismo de la realización de la cola cuando no hay disponibilidad de libros. Esta compuesta por 6 clases, dos clases son objetos una para el top y otra para la cola, dos nodos las cuales un nodo es para la lista doblemente enlazada y el segundo nodo es para la realización de la cola.



ARBOLES

Los árboles son las estructuras de datos más utilizadas, pero también una de las más complejas. Los árboles se caracterizan por almacenar sus nodos en forma jerárquica y no en forma lineal como las Listas ligadas, Colas, pilas, etc, de las cuales ya hemos hablado en días pasados.



Datos importantes de los árboles:

Para comprender de mejor manera que es un árbol comenzaremos explicando cómo está estructurado.

- **Nodos:** Se le llama Nodo a cada elemento que contiene un árbol.
- **Nodo Raíz:** Se refiere al primer nodo de un árbol, solo un nodo del árbol puede ser la raíz.
- **Nodo Padre:** Se utiliza este término para llamar a todos aquellos nodos que tienen al menos un hijo.
- **Nodos hijo:** Los hijos son todos aquellos nodos que tiene un padre.

- **Nodo Hermano:** Los nodos hermanos son aquellos nodos que comparte a un mismo padre en común dentro de la estructura.
- **Nodo Hoja:** Son todos aquellos nodos que no tienen hijos, los cuales siempre se encuentran en los extremos de la estructura.
- **Nodo Rama:** Estos son todos aquellos nodos que no son la raíz y que además tiene al menos un hijo.

ESTRUCTURA DE DATOS

Una estructura de datos es un tipo de dato compuesto que permite almacenar un conjunto de datos de diferente tipo. Los datos que contiene una estructura pueden ser de tipo simple (caracteres, números enteros o de coma flotante, etc.)

A cada uno de los datos o elementos almacenados dentro de una estructura se les denomina miembros de esa estructura y estos pertenecerán a un tipo de dato determinado.

- **Estructuras de datos estáticas:** Son aquellas en las que se asigna una cantidad fija de memoria y no cambia durante la ejecución de un programa, es decir, las variables no pueden crearse ni destruirse durante la ejecución del programa.
- **Estructuras de datos dinámicas:** Son aquellas en las que su ocupación en memoria puede aumentar o disminuir durante el tiempo de ejecución de un programa.

  A su vez las estructuras de datos dinámicas se pueden clasificar en lineales y no lineales:

  - Estructuras lineales
  - Estructuras no lineales

Estructuras lineales:

Son aquellas en las que se definen secuencias como conjuntos de elementos entre los que se establece una relación de predecesor y sucesor. Las diferentes estructuras de datos basadas en este concepto se diferencian por las operaciones de acceso a los elementos y manipulación de las estructuras.

Existen tres estructuras lineales especialmente importantes: las pilas, las colas y las listas.

CONCLUSIONES

- Como hemos observado los árboles son estructuras bastante complejas, tiene una gran aplicación en la ciencia y en la programación convencional. En los últimos años este tipo de estructuras ha sido utilizadas con mucha frecuencia en la Inteligencia artificial.

- Mediante el desarrollo de este trabajo se ha logrado desarrollar y entender el curso de Estructura de Datos.