

---

## PROYECTO 1 – FASE 2 – UDrawing Paper

---

202000119 – Sergie Daniel Arizandieta Yol

### Resumen

El proyecto se enfoca en la gestión de Tipo de Dato Abstracto (TDA) para el uso de memoria dinámica mediante listas, árboles y sus derivados en Java, siendo así cada lista siendo aplicada a distintos propósitos.

Donde se aplican para la gestión de usuarios tanto de administradores y clientes. Donde cada uno tienen sus características para generación de reportes y datos, usando Json para la carga de los datos, al mismo tiempo mostrando el estado del sistema de manera gráfica mediante Graphviz en la interfaz.

Dicho proyecto hace utilización de estas tecnologías de una manera novedosa para la búsqueda, lecturas y escrituras optimas de los datos a gestionar para la UDrawing Paper Fase 2 requerido en el 2022 (Se recalca que el proyecto no es una fase final).

### Palabras clave

TDA, JAVA, JSON, GRAPHVIZ, ARBOLES

### Abstract

*The project focuses on the management of Abstract Data Type (ADT) for the use of dynamic memory through lists, trees and their derivatives in Java, thus each list being applied to different purposes.*

*Where it is applied for user management of both administrators and clients. Where each one has its characteristics to generate reports and data, using Json to load the data, while it shows the status of the system graphically through Graphviz in the interface.*

*This project makes use of these technologies in a novel way for the optimal search, reading and writing of the data to be managed for the UDrawing Paper Phase 2 required in 2022 (It is emphasized that the project is not a final phase).*

### Keywords

TDA, JAVA, JSON, GRAPHVIZ, TREES

## Introducción

En el ámbito cotidiano a nivel laboral y empresario es indispensable una buena gestión de memoria, por lo cual se recurre a la memoria dinámica que cambian constantemente el tamaño de esta, para ser menos cargados en los softwares utilizados por diferentes equipos al tener que procesar demasiados registros por lo cual se recurre a las TDA's, almacenado cada registro en sus tuplas así guardando toda la información en dicho proyecto enfocado en árboles.

Al mismo tiempo su uso e implementación suele ser confuso, por lo cual también el manejo de estas es de alta importancia, una de las implementaciones de TDA's.

Estas aparte de almacenar a la información es la que presenta de dichos datos para poder generar una salida mediante archivos de extensión JPG para presentación de la estructura y DOT para los grafos.

Donde los archivos de entrada son de extensión JSON

## Desarrollo del tema

Las tecnologías requeridas y fundamentales para el desarrollo del software son:

- ❖ Implementación de TDA
  - Árboles
  - Manipulación de las tuplas de las TDA
- ❖ Archivos Json
  - Definición de los archivos Json
- ❖ Generación de grafos mediante Graphviz

## Conceptos generales

- Tipo de dato abstracto (TDA)

Son el nivel más alto de abstracción y son independientes del lenguaje de programación, básicamente son una colección de valores y de operaciones que se definen mediante una especificación en el presente proyecto implementado en listas dinámicas.

- Árboles:

Se define como una tupla de n elementos los cuales son asignados dinámicamente donde hay diferentes tipos de árboles para el gestionamiento de datos para la optimización de búsquedas, entre las funciones que puede haber, recordando que todas se manejan con un id para el orden de estas:

- Obtener datos
- Mostrar
- Eliminar
- Agregar
- Otros



Para ver mas a fondo los tipos de árboles implementados, recorridos y características a fondo ver:

<https://drive.google.com/file/d/17kXBQxLIUO9O7O52EVuRqINOpRmi19Dy/view?usp=sharing> , Autor: Sergie

Arizandieta, fuente de información clase de Estructuras de datos, Facultad de ingeniería, Usac, 2022

- Manipulación de las tuplas de una TDA

Es muy similar a la POO (Programación Orientada Objetos), ya que cada tupla cuenta con atributos específicos, los cuales se pueden hacer referencia gracias a que son definidos en su constructor el cual se encarga de definir los atributos de las tuplas cuando llamadas para crear una instancia de estas.

- Archivos JSON

JSON sus siglas en ingles son “JavaScript Object Notation” el cual traducido seria “Notación de Objetos de JavaScript”. Se utiliza para estructurar datos en forma de texto y permite el intercambio de información entre aplicaciones de manera sencilla, liviana y rápida. Es una alternativa al XML.

- Grafos en Graphviz:

Graphviz es un software de visualizador de gráficos de código abierto, el cual implementa su extensión DOT para sus grafos.

#### ➤ DOT

Es un lenguaje descriptivo de texto plano el cual es una forma simple de describir los grafos.

## Representación de los módulos del programa

Tabla I.

*Módulos del proyecto.*

<i>Clases</i>
Main
Clients
Nodes_Colors
Árbol_AVL
Árbol_Binario
ArbolB
MatrizDispersa
NodoB
RamaB
Simple_Album
Simple_Sublist_Album
Storage
<i>Ventanas</i>
Login
Check_in
Admin_Module
Client_Module
Circular_Doble_espera

### Descripción de módulos:

- Main  
Se encarga solo de la instancia de la clase de storage para la permanencia de datos e iniciando la aplicación con el modulo de login.

- **Clients**

Es un objeto para los clientes que se registren en el sistema.

- **Nodes\_Colors**

Es un objeto para almacenar los pixeles de las imágenes.

- **Árbol\_AVL**

Es un árbol AVL implementado para el almacenamiento de las imágenes de los usuarios por ID y cada imagen tiene su propio árbol binario.

- **Árbol\_Binario**

Es un árbol binario implementado para el almacenamiento de las capas de cada imagen por ID.

- **ArbolB**

Es un árbol binario implementado para el almacenamiento de los clientes de cada uno por DPI.

- **MatrizDispersa**

Es una matriz común, donde las casillas que no están siendo ocupadas, optimizando la memoria exponencialmente.

- **NodoB**

Es un objeto que posee las características de un árbol b el cual se inserta en una RamaB.

- **RamaB**

Es un objeto que actúa como cabecera para los nodoB, básicamente una lista de nodosB, para entender el funcionamiento leer el documento adjunto en el apartado de árboles.

- **Simple\_Album**

Es lista simple donde se almacenan los álbumes de cada usuario, teniendo como sublista para las imágenes.

- **Simple\_Sublist\_Album**

Es lista simple donde se guardan las imágenes de los álbumes.

- **Storage**

Es una clase la cual tienen la instancia de los usuarios y la permanencia del cliente en cuestión en sesión, la cual se encarga del almacenamiento.

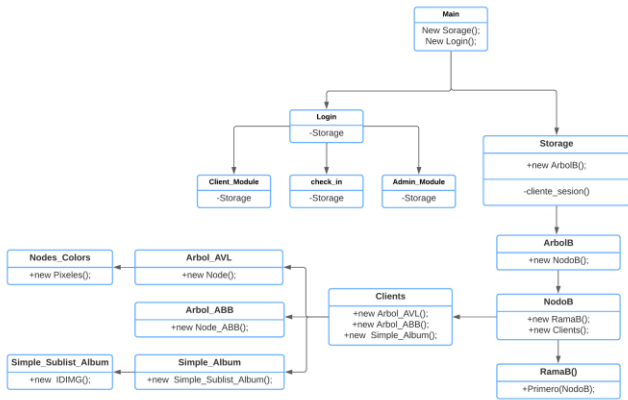


Figura 1. Diagrama de Clases.

Fuente: elaboración propia.

## Conclusiones

El uso de nuevas tecnologías, implementaciones inclusive librerías que puedan hacer el trabajo de desarrollo es vital para una empresa viéndolo a gran escala, como son las TDA con los arboles, ya que la mayoría de las veces el cliente siempre requerirá que se tenga el menor estrés en sus computadoras al ejecutar los softwares necesarios.

Además, no siempre podemos dar por hecho que se tendrá datos estáticos, por lo cual el uso de memoria dinámica a grandes rasgos es fundamental, la utilización y operaciones de las TDA son flexibles a como lo pueda lograr el desarrollador del software.

El presente software solo es una de miles de tipos de implementación que uno requiera, siendo así, ¿cómo se ocurriría a ti implementar memoria dinámica para la optimización de procesos muy rígidos que conozcas?

## Referencias bibliográficas

Arias Guerra, D. (2008) Estructura de datos Avanzadas (Revisado, ed., Vol. 9). Universidad de las Ciencias Informáticas. <https://cutt.ly/eWdkzvt>

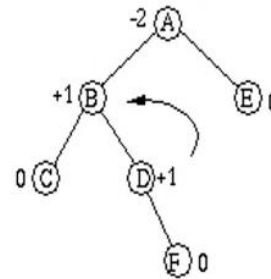
Ellson, J. (2005) Graphviz. Graphviz org. <https://graphviz.org/doc/info/lang.html>

Ferris Castell, R. (2004) Algoritmos y Estructura de Datos I (Revisado ed., Vol. 1) Universidad de Valencia

[http://informatica.uv.es/iiguia/AED/oldwww/2001\\_02/Teoria/Tema\\_10.pdf](http://informatica.uv.es/iiguia/AED/oldwww/2001_02/Teoria/Tema_10.pdf)

## Anexos

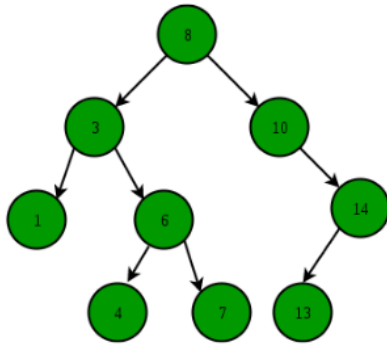
Modelo “Árbol AVL”



Fuente: Presentación 7, Clase: Estructuras de Datos sección C Link:

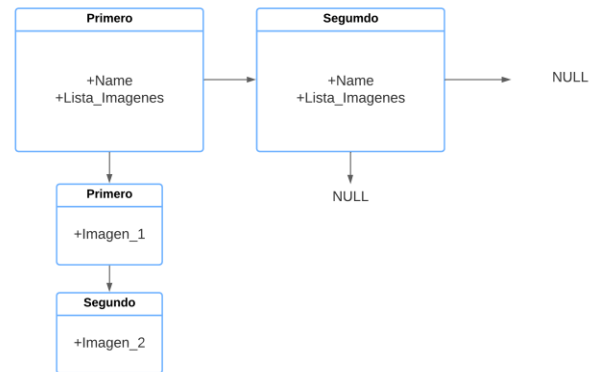
[https://docs.google.com/presentation/d/1rC4-cHSJSI2fSI5i2Qs3BqpA5a7cJM\\_r/edit?usp=sharing&ouid=108306629557817213351&rtfpof=true&sd=true](https://docs.google.com/presentation/d/1rC4-cHSJSI2fSI5i2Qs3BqpA5a7cJM_r/edit?usp=sharing&ouid=108306629557817213351&rtfpof=true&sd=true) , recopilado 30/3/2022

### Modelo “Árbol binario de búsqueda”



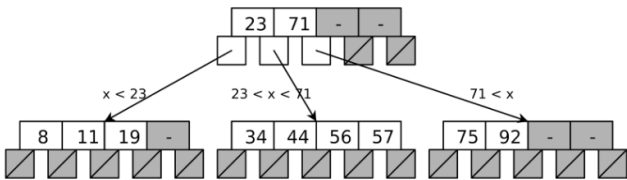
Fuente: Presentación 6 Link:, Clase: Estructuras de Datos sección C Link:  
<https://docs.google.com/presentation/d/1Af9xe9z6rVX1UccnZQbOMWLLdo0LAUdR/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true>, recopilado 30/3/2022

### Modelo “Simple\_Album”



Fuente: Elaboración Propia

### Modelo “Árbol B”



Fuente: Presentación 6 Link:, Clase: Estructuras de Datos sección C Link:  
[https://docs.google.com/presentation/d/1AFMi2DP\\_GOPBDFus52wOl\\_GPU0ZuXRYb/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true](https://docs.google.com/presentation/d/1AFMi2DP_GOPBDFus52wOl_GPU0ZuXRYb/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true), recopilado 30/3/2022