

---

## PROYECTO 1 – FASE 3 – UDrawing Paper

---

202000119 – Sergie Daniel Arizandieta Yol

### Resumen

El proyecto se enfoca en la implementación de tablas hash, criptografía y blockchain para la permanencia de datos dentro del sistema.

Donde son utilizados para el almacenamiento de transacciones dentro del sistema de los clientes con diversas operaciones que poseen, así mismo a pesar del cierre de la aplicación se guardan el blockchain en archivos json para su posterior lectura.

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 3 requerido en el 2022 (Se recalca que el proyecto no es una fase final).

### Palabras clave

BLOCKCHAIN, JAVA, JSON, GRAPHVIZ, HASH

### Abstract

*The project focuses on the implementation of hash tables, cryptography and blockchain for data permanence within the system.*

*Where they are used for the storage of transactions within the client system with various operations that they have, likewise, despite the closure of the application, the blockchain is saved in json files for later reading.*

*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 3 required in 2022 (It is emphasized that the project is not a final phase).*

### Keywords

BLOCKCHAIN, JAVA, JSON, GRAPHVIZ, HASH

## Introducción

En la actualidad muchas tecnologías han sido novedosas, pero al mismo tiempo peligrosas, ya que se trata de tener una red conjunta con permanencia de datos, pero permanencia dentro de un ámbito que cualquiera puede editar la información, blockchain es una tecnología que resuelve dicho problema implementando diferente método para el almacenamiento de datos y que no cualquiera o que sea tan fácil de corromper los datos.

Donde dicha tecnologías se tratan de integrar de una menor escala pero con introducción a ello.

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:

- ❖ Hash
- ❖ Blockchain
- ❖ Tablas Hash
- ❖ Arboles merkle
- ❖ Archivos Json
  - Definición de los archivos Json
- ❖ Generación de grafos mediante Graphviz

## Conceptos generales

- Hash

Una función criptográfica hash- usualmente conocida como “hash”- es un algoritmo matemático que transforma cualquier bloque arbitrario de datos en una nueva serie de caracteres con una longitud fija. Independientemente de la longitud de los datos de entrada, el valor hash de salida tendrá siempre la misma longitud.
- Blockchain

Se traduce como cadena de bloques. Básicamente, blockchain es de un conjunto de tecnologías que permiten llevar un registro seguro, descentralizado, sincronizado y distribuido de las operaciones digitales, sin necesidad de la intermediación de terceros.
- Tablas Hash:

Una tabla hash, matriz asociativa, hashing, mapa hash, tabla de dispersión o tabla fragmentada es una estructura de datos que asocia llaves o claves con valores. La operación principal que soporta de manera eficiente es la búsqueda: permite el acceso a los elementos almacenados a partir de una clave generada.
- ❖ Arboles Merkle

Un árbol hash de Merkle o árbol de Merkle o árbol hash es una estructura de datos en árbol, binario o no, en el que cada nodo que no es una hoja está etiquetado con el hash de la

concatenación de las etiquetas o valores de sus nodos hijo. Son una generalización de las listas hash y las cadenas hash

## Representación de los módulos del programa

Tabla I.

*Módulos del proyecto.*

- 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.

<i>Clases</i>
Main
Clients
Nodes_Colors
Árbol_AVL
Árbol_Binario
ArbolB
MatrizDispersa
NodoB
RamaB
Simple_Album
Simple_Sublist_Album
Storage
Árbol_Merkle
Block_Chain
Tabla_Hash
<i>Ventanas</i>
Login
Check_in
Admin_Module
Client_Module
Circular_Doble_espera
Delivery
Transaccion
Simple_Recorrido

### 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 actual 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 el cual tienen la instancia de los usuarios y la permanencia del cliente en cuestión en sesión, la cual se encarga del almacenamiento.

- **Arboles Merkle**

Es una clase para la generación de dicho árbol a partir de las transacciones del sistema

- **Blockchain**

Es una clase para el almacenamiento de datos de las transacciones en un dado tiempo.

- **Tablas Hash**

Es una clase para el almacenamiento de los mensajeros así mismo como para aplicar los métodos de colisión y hash de estos

- **Transacción**

Es una clase objeto para el almacenamiento de las operaciones de las transacciones realizadas

- **Simple\_Recorrido**

Es una lista TDA simple que permite el resguardo del bloackchain

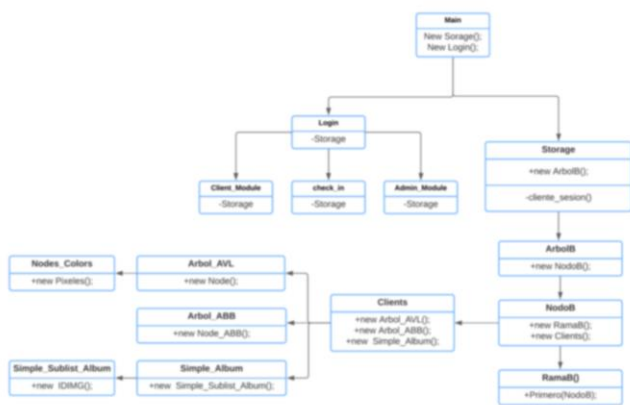


Figura 1. Diagrama de Clases.

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, casi como las implementaciones seguras de almacenamiento que talvez no sean lo más fácil de comprender pero son las cuales permiten la protección de los datos.

El presente software solo es una de miles de tipos de implementación que uno requiera, siendo así, ¿cómo sete ocurriría a ti implementar con estas nuevas tecnologías 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>

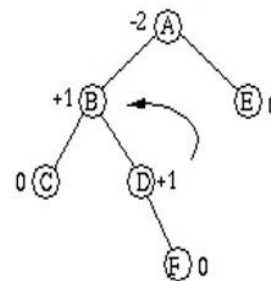
Ellson, J. (2005) Graphviz. Graphviz org. <https://graphviz.org/doc/info/lang.html>

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

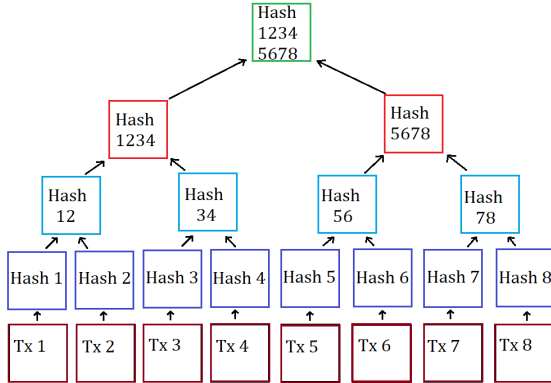
Modelo “Árbol AVL”



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

[https://docs.google.com/presentation/d/1rC4-cHSJSI2fSI5i2Qs3BqpA5a7cJM\\_r/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true](https://docs.google.com/presentation/d/1rC4-cHSJSI2fSI5i2Qs3BqpA5a7cJM_r/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true) , recopilado 5/5/2022

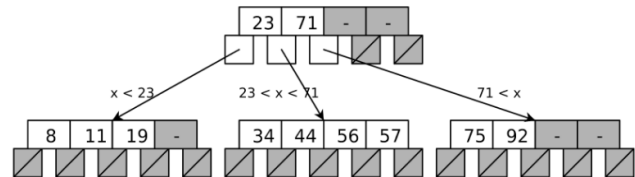
### Modelo “Árbol Merkle”



Fuente: Presentación 9, Clase: Estructuras de Datos sección C  
Link: [https://docs.google.com/presentation/d/1rC4-cHSJSI2fSI5i2Qs3BqpA5a7cJM\\_r/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true](https://docs.google.com/presentation/d/1rC4-cHSJSI2fSI5i2Qs3BqpA5a7cJM_r/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true) , recopilado 5/5/2022

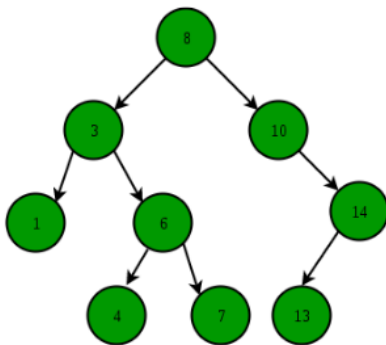
<https://docs.google.com/presentation/d/1Af9xe9z6rVX1UccnZQbOMWLLdo0LAUdR/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true>, recopilado 5/5/2022

### Modelo “Árbol B”



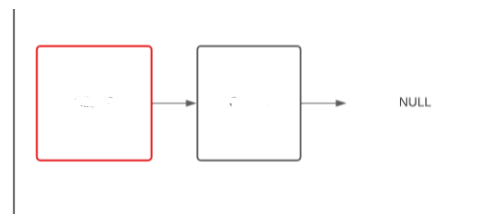
Fuente: Presentación 6 Link:, Clase: Estructuras de Datos sección C Link: [https://docs.google.com/presentation/d/1AFMi2DPGOPBDFus52wOI\\_GPU0ZuXRYb/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true](https://docs.google.com/presentation/d/1AFMi2DPGOPBDFus52wOI_GPU0ZuXRYb/edit?usp=sharing&ouid=108306629557817213351&rtpof=true&sd=true) , recopilado 30/3/2022

### Modelo “Árbol binario de búsqueda”



Fuente: Presentación 6 Link:, Clase: Estructuras de Datos sección C Link:

### Modelo “Simple\_Recorrido”



Fuente: Elaboración Propia