
**Título:**

INFORME TRABAJO PRÁCTICO ESPECIAL 1

**Nivel y Área:**

72.34 - Estructuras de Datos y Algoritmos

2º Cuatrimestre 2017

**Comisión:** S - **Carrera:** Informática - **Grupo:** Nº1

**Fecha:** 13 de octubre de 2017

**Alumnos Expositores:**


- AQUILI, Alejo Ezequiel
- BASSANI, Santiago
- LO COCO, Juan Pablo
- PINILLA, Lautaro Joaquín
- SANGUINETI ARENA, Francisco Javier

# ***Trabajo Práctico Especial 1***

## ***Algoritmos y estructuras principales utilizadas:***

El desarrollo e implementación del TPE Nº1 conlleva la siguiente jerarquía e disposición de estructuras de datos y algoritmos empleados:

Por un lado, se desarrolló una Blockchain compuesta por Block y una HashFunction, esta estructura es completamente genérica y puede ser utilizado bajo cualquier circunstancia que requiera una Blockchain simple y no distribuida montada sobre un ArrayList y no sobre una lista simplemente encadenada por el hecho de que fue requerido poder realizar un get sobre los bloques de la Blockchain, de esta forma se puede acceder a los mismos con una complejidad temporal de O(1). En cuanto al mining, cada Block posee un método mine y para validar la Blockchain con su correspondiente método se itera sobre la misma recorriendola de principio a fin resultando en una complejidad temporal de la validación O(n) donde n es el número de bloques de la Blockchain.

Por otra parte, se implementó un AVLTree con sus operaciones básicas de inserción remoción y sus pertinentes rotaciones. Esta estructura convive con la Blockchain guardando sus operaciones (AVLOperationData) en la Blockchain y mediante una estructura compuesta por un AVLTree y una Blockchain de AVLOperationData llamada AVLBlockchain. Este, permite que ambas estructuras de datos convivan (Lo cual conforma el principal objetivo del trabajo) y cumpla con los algoritmos y los requisitos solicitados.

## ***Decisiones importantes:***

Durante la implementación del TPE Nº1, como primera decisión importante se optó por trabajar con String en Java y no con Byte para el hash de los Blocks que conforman la Blockchain. Dado que natívamente los métodos de HashFunction trabajan con String y con arrays de byte (byte[]), que involucra iterar los caracteres de los Strings para realizar dicha conversión y trabajar con bytes o bytes[] qhe en Java resulta poco amigable. Junto con el hecho que no se encontró mejora alguna en el trabajo de la Blockchain con bytes[] se decidió trabajar con Strings que resultó en términos generales igual o mejor aún en lo que refiere a complejidades temporales y resultó mucho más cómodo para la realización de operación sobre el String como por ejemplo la comparación.
Por otra parte, se decidió utilizar la función de hashing MD5 por resultar la más rápida para la realización del mining de los bloques. Para ello, se comparó y probaron varias funciones de hash (MD2, MD5, SHA-1, SHA-256, SHA-318, SHA-512) y los resultados de estos estudios se pueden ver en el siguiente gráfico generado de la utilización de un framework graficador llamado JavaPlot:

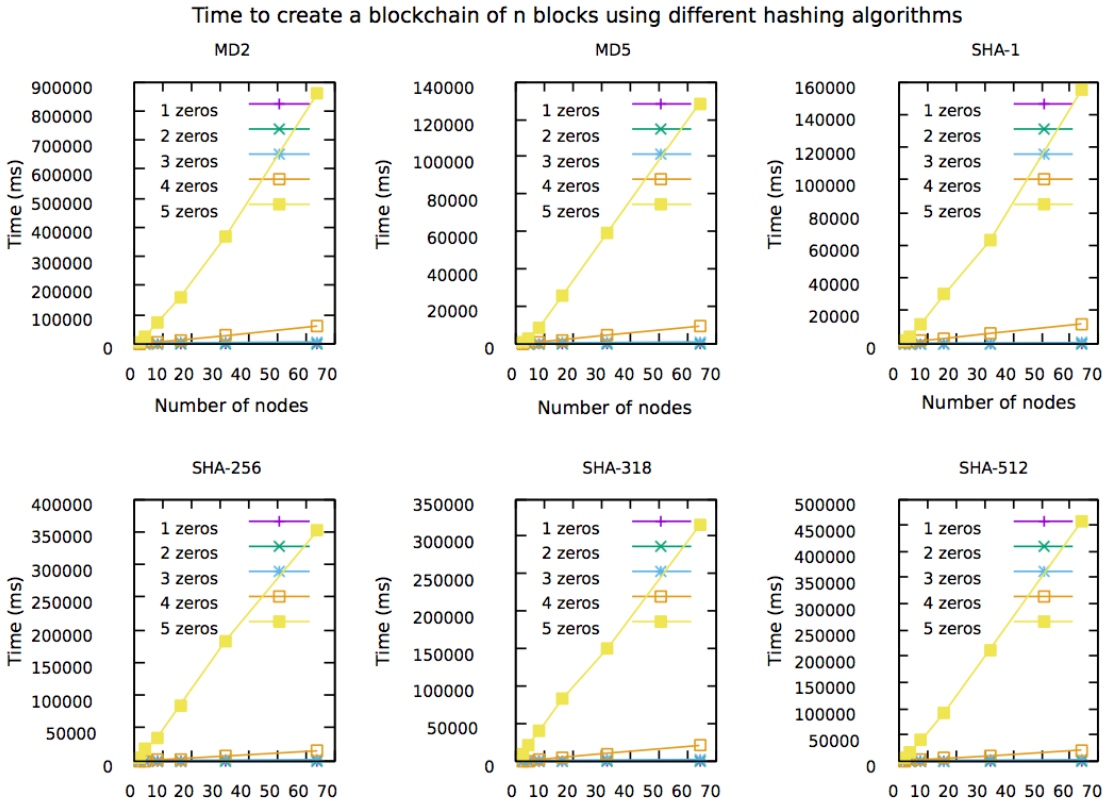


*Figura 1 – Gráficos del tiempo requerido para crear una Blockchain de n bloques en función del tiempo para los distintos algoritmos de hashing estudiados.*

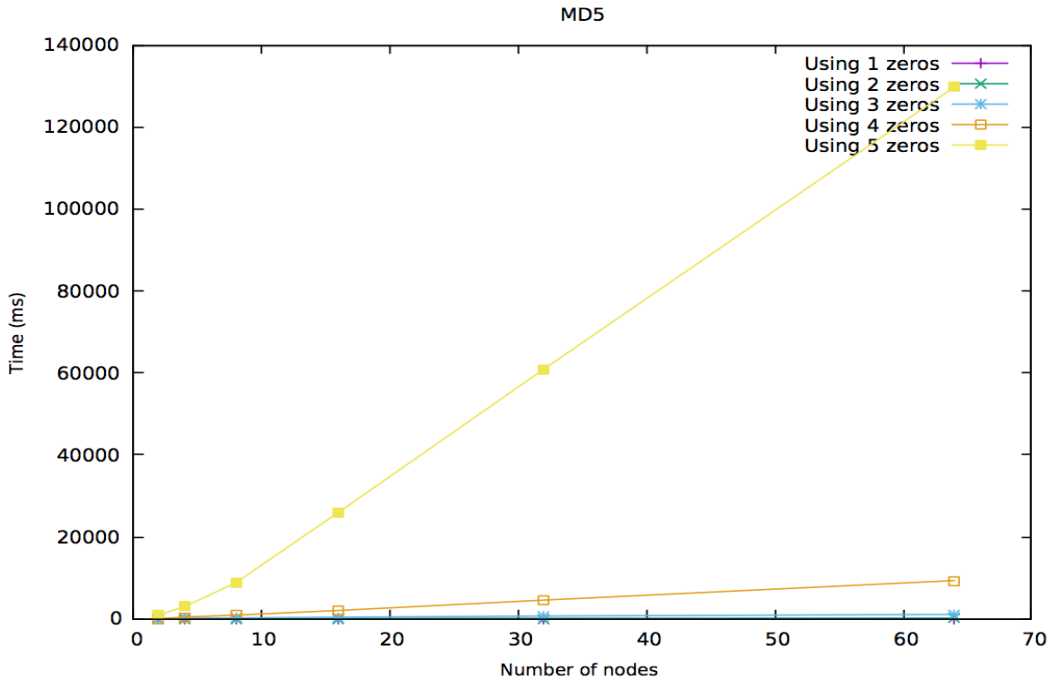


*Figura 2 – Gráfico del tiempo requerido para la creación de una Blockchain de n bloques en función del tiempo para el algoritmo de hashing elegido (MD5).*

Otra decisión de gran importancia dentro de la implementación del TPE fue la crear una estructura de datos que brinde soporte al almacenamiento de operaciones realizadas sobre un árbol AVL, dicha estructura

(AVLOperationData) permite llevar el registro de operaciones realizadas bloque a bloque en la Blockchain, donde cada operación del AVLTree genera como resultado un AVLOperationData y esto es lo que se almacena en el campo de data de la Blockchain.

Sumada a esta decisión de forma paralela se decidió implementar una Blockchain y todos sus componentes del tipo Serializable, de esta forma se puede guardar un stream del objeto o instancia de la Blockchain que se posee en Runtime en un archivo. Luego se puede leer un archivo que contenga guardado una Blockchain y reconstruir el árbol subyacente a dicha cadena.

## ***Dificultades:***

Las principales dificultades circularon en torno a decidir cuáles eran las estructuras de datos a implementar para poder cumplir con el objetivo, luego de que forma estas iban a convivir y finalmente determinar la implementación de una estructura que haga convivir una Blockchain y un AVLTree lo cual se escapó de la planificación inicial, resultando el primer diseño descartado y se procedió con este último nuevo enfoque, por otro lado, se presentaron durante la implementación dificultades con la serialización de la Blockchain y sus componentes.

Un gran número de dificultades se presentaron con la implementación de los test, para el testing se utilizó el framework de Java, JUnit 5 y resultó problemático realizar testeos (como, por ejemplo, las clases Singleton) sobre métodos donde su scope se alejaba sustancialmente de los test a realizar lo cual resultó complicado el testing de las estructuras subyacentes a AVLBlockchain.

## ***Conclusiones:***

Se concluye que es posible la implementación de un AVLTree montado conjuntamente con una Blockchain que registre las operaciones realizadas sobre el árbol con funcionamiento dinámico y permitiendo operar con el árbol utilizando también esta estructura de datos tan particular como es la Blockchain, lo cual nos acerca más a este impulso o piedra fundamental de una concepción nueva y revolucionaria de internet, como lo es el internet del valor.

Más allá de las criptomonedas, el Internet del valor también se ha creado sobre estándares abiertos, pero encuentra su base en la tecnología Blockchain. De la misma forma que podemos acceder a páginas webs en todo el mundo en el Internet de la Información, en el Internet del valor tenemos una herramienta nueva para compartir y gestionar valor de activos o bienes digitales sin la necesidad de depender de una entidad central de confianza que centralice el proceso.