UNIVERSIDAD DE INGENIERIA Y TECNOLOGIA

ALGORITMOS Y ESTRUCTURA DE DATOS

PROYECTO

1. Consideraciones generales

- Debe usar sus propias estructuras de datos implementadas en el curso: circular array, linked lists, bst, heap, hash, etc.
- Deben construir dichas estructuras de datos de la manera más desacoplada posible. Hacer uso de plantillas (templates), de tal forma que pueda utilizarlos dentro de su aplicación sin necesidad de modificar el código interno de las estructuras, sólo instanciar objetos de estas para utilizarlos o extender la funcionalidad de una estructura mediante herencia.
- Las estructuras pueden utilizar lambdas para implementar los criterios de comparación.
- La interfaz gráfica puede ser por consola o GUI (Qt Project, Visual Studio, WxWidgets).
- En el caso de evidenciar plagio con otros grupos, se asigna una calificación de cero (0) a todos los integrantes del grupo.
- La calificación en la exposición es individual por cada integrante de grupo, por lo que es mandatorio que todos conozcan el trabajo.
- La aplicación debe funcionar correctamente para que se considere el 100% de la calificación.

---

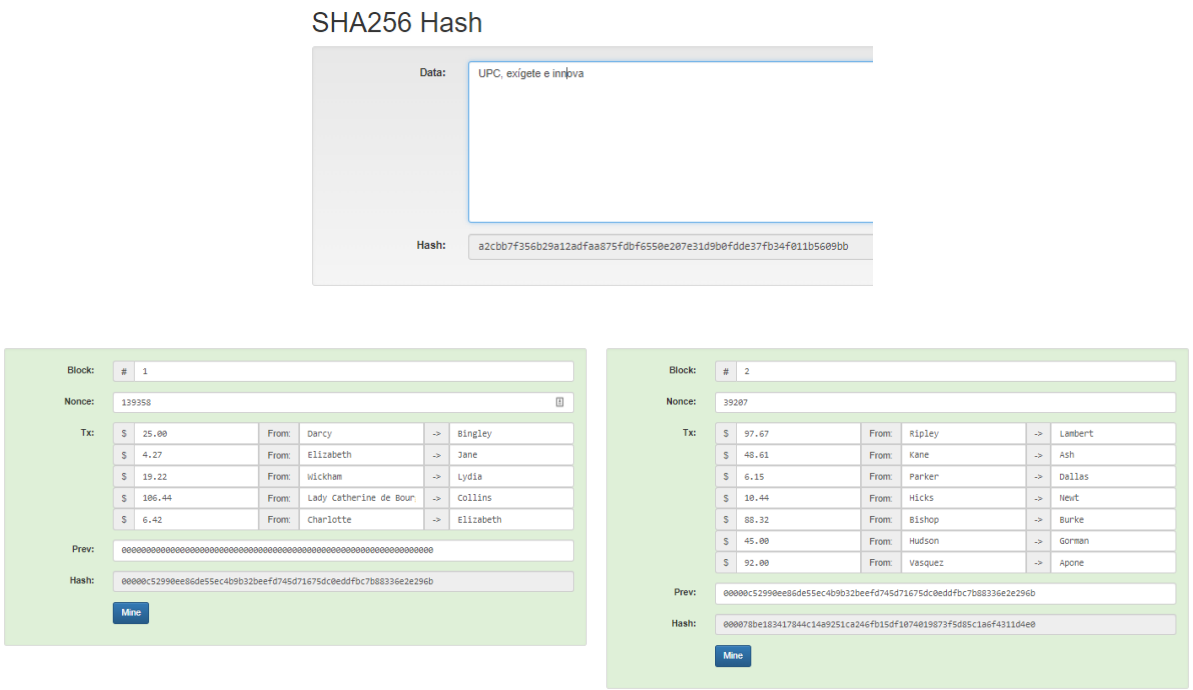
2. Enunciado del Proyecto

El proyecto consiste en desarrollar una aplicación transaccional de interacción de datos teniendo como base de seguridad una estructura de datos de cadena de bloques (Blockchains en un solo host).

https://www.youtube.com/watch?v=C5NZnD12yjg

2.1 Hito 1: Implementación del Blockchain

- La estructura de datos del Blockchains es una lista de bloques de registros ordenados cronológicamente.
- Cada bloque es identificable por un código hash, generado mediante un algoritmo de hash criptográfico a partir de toda la información que contiene el bloque. De esta manera, el código hash también servirá para verificar la integridad del bloque.
- Para mantener las propiedades del hash (indexación eficiente en el rendimiento de las búsquedas, libre de colisiones y estable) se recomienda utilizar una librería estándar de criptografía como OpenSSL que ya implementa los mejores algoritmos de hashing.

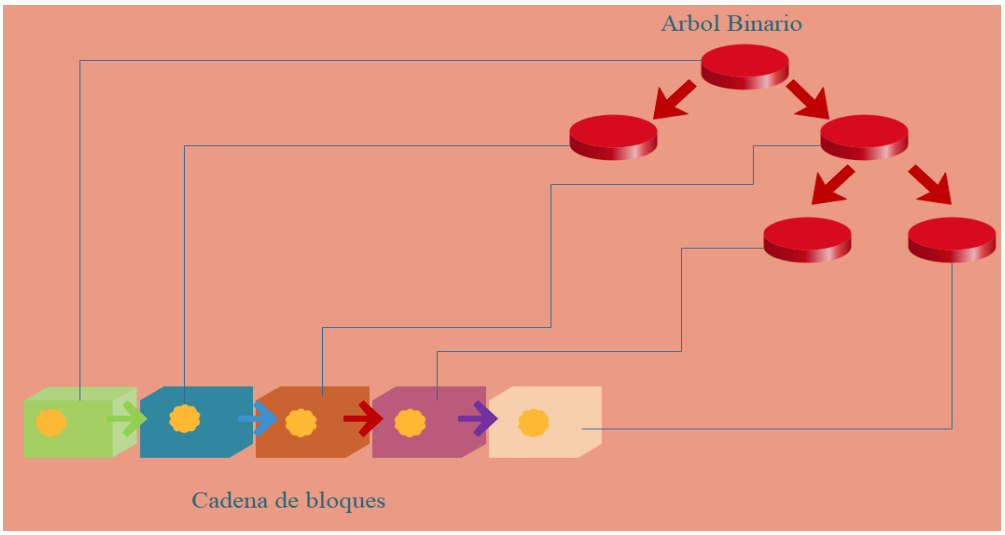


Fuente: https://andersbrownworth.com/blockchain/

- Cada bloque hace referencia a un bloque anterior en el campo "Prev", también conocido como bloque padre. Usted debe proponer la estructura de datos más apropiada para gestionar los bloques de manera eficiente de acuerdo a las operaciones que se requieren.
- La identidad propia del bloque hijo cambia si la identidad del padre cambia.
    - Si por alguna razón el contenido del bloque padre se modifica, el código hash del padre también cambia.
    - El cambio de hash del padre requiere una alteración en el puntero "Prev" del hijo.
    - Esto, a su vez, hace que el hash del hijo mute, lo que requiere un cambio en el puntero del nieto, que a su vez altera al nieto y así sucesivamente.
- Este efecto en cascada garantiza que, una vez que un bloque tiene muchas generaciones que le suceden, no pueda cambiarse sin **forzar consecuentemente un recálculo** de todos los bloques posteriores (**para asegurar la integridad de los bloques**).
    - Por ende, la existencia de una larga cadena de bloques fortalece la seguridad que sea inmutable.
- Se debe investigar e implementar alguna técnica de "**proof of work".**

2.1 Hito 2: Aplicación

Como caso aplicativo, se requiere implementar un sistema informático transaccional que permita a un usuario del sistema registrar operaciones de manera segura (blockchain) para luego realizar búsquedas de manera eficiente usando diversas estructuras de datos como mecanismos de indexación para ciertos criterios de búsqueda.



Requerimientos de la aplicación:

- La aplicación debe cargar los datos transaccionales desde archivos de texto. Puede trabajar con algún dataset de Kaggle o generar datos ficticios.
- Ejemplos de dominios transaccionales:
  - Retiros de dinero: cliente, lugar, monto, fecha
  - Transferencias bancarias: emisor, receptor, monto, fecha
  - Registro de ventas: cliente, vendedor, monto, fecha
- La aplicación deberá permitir el ingreso de nuevas transacciones.
- Dado los siguientes criterios de búsqueda, usted debe indexar los bloques con la estructura de datos mas apropiada de acuerdo al tipo de filtrado requerido:
  - Igual a X
  - Entre X y Y
  - Inicia con
  - Está contenido en
  - Máximo valor de
  - Mínimo valor de
- Usted debe utilizar al menos tres estructuras de datos para implementar los criterios de búsquedas: BST, Hash, Heap, BTree, B+Tree, Trie.
  - El uso de B+Tree tendrá puntos extras en evaluación continua.
- Realizar cálculos en las transacciones por usuario. Por ejemplo: cálculo del monto total recaudado por el usuario X, la cantidad de clientes registrados por el usuario X.
- La aplicación debe soportar la eliminación de registros, pero solo se aplicará en el índice.
- El Blockchain no permite alteraciones en su contenido, pero para este caso aplicativo vamos a forzar las actualizaciones de registros con el fin de realizar el recalculo en cascada.

3. Entrega

- El trabajo se realizará de manera grupal. Se debe asignar las tareas y hacer seguimiento a su cumplimiento.
    - Utilizar Project boards o tables de GitHub
    - Opcionalmente también pueden usar Jira o Clickup.
- Elaborar un pequeño informe en el Readme o Wiki de Github con la siguiente estructura:
    - Introducción
    - Descripción del caso de estudio planteado por el grupo
    - Importancia del Blockchain en el dominio de datos elegido.
    - Explicación de la estructura de datos del Blockchain y la estrategia para asegurar la integridad de su contenido. Además, indicar como se implementó el **proof of work.**
    - Explicación de cada una de las estructuras de datos utilizada en su aplicación de acuerdo a los criterios de búsqueda.
        - Use diagramas para una mejor claridad
    - Análisis de la complejidad en notación Big O de los métodos del Blockchain.
        - Principalmente de los métodos de insertar y buscar
        - Realizar una tabla comparativa de Blockchain con índices vs sin índices.
    - Conclusiones
    - Referencias bibliográficas
- Anexar al informe la lista de actividades concluidas por cada integrante del grupo.
- El informe se subirá al aula virtual, así como el enlace del repositorio Github asignado para grupo.
    - Fecha de Hito 1: 10 de Octubre
    - Fecha de Hito 2: 01 de Diciembre
- PD: Tendrán la asesoría constante de los ACLs, quienes les proveerán un dataset y casos de prueba para uniformizar los resultados obtenidos.

4. Presentación

Para obtener la calificación completa indicada en la rúbrica, cada estudiante debe participar de la presentación del proyecto y responder las preguntas planteadas, además que la aplicación debe funcionar a un 100%. La exposición tendrá una duración máxima de 10 minutos en donde deben participar todos los integrantes, luego tendrán una ronda de preguntas.

Presentación en PowerPoint conteniendo no más de 10 slides:

- Descripción del problema y objetivos
- Elaboración de la solución (gráficos)
- Conclusiones

En lugar de PowerPoint puede presentar también su informe siempre y cuando no sea muy extenso e incluya gráficos.