Búsqueda de claves distribuidas con algoritmo SHA-1 en java

Alumnos:

Ayrton Coronado - acoronadoh@uni.pe

Flores Huamaní - lfloresh@uni.pe
Barrientos Porras - herlees.barrientos.p@uni.pe
Universidad Nacional de Ingeniería, Facultad de Ciencias,

Curso:

CC462 Sistemas concurrentes y distribuidos

Laboratorio 4

Resumen

El presente trabajo es la implementación y deploy del proceso de blockchain de búsqueda de claves en clientes(mineros) distribuidos en java, empleando comunicación cliente-servidor mediante sockets y threads en java.

Palabras clave: Blockchain, sockets, cliente,minero, servidor, threads, java..

Contenidos

01. Introducción

Este es el proceso de blockchain de búsqueda de keys mediante el algoritmo SHA-1 distribuyendo palabras a todos los mineros, estos detienen su búsqueda al momento que algún minero encontró la primera clave y para validar el resultado todos los demás mineros deben comprobar la veracidad de la key.

02. Marco Teórico

a. Creación de Threads en java.

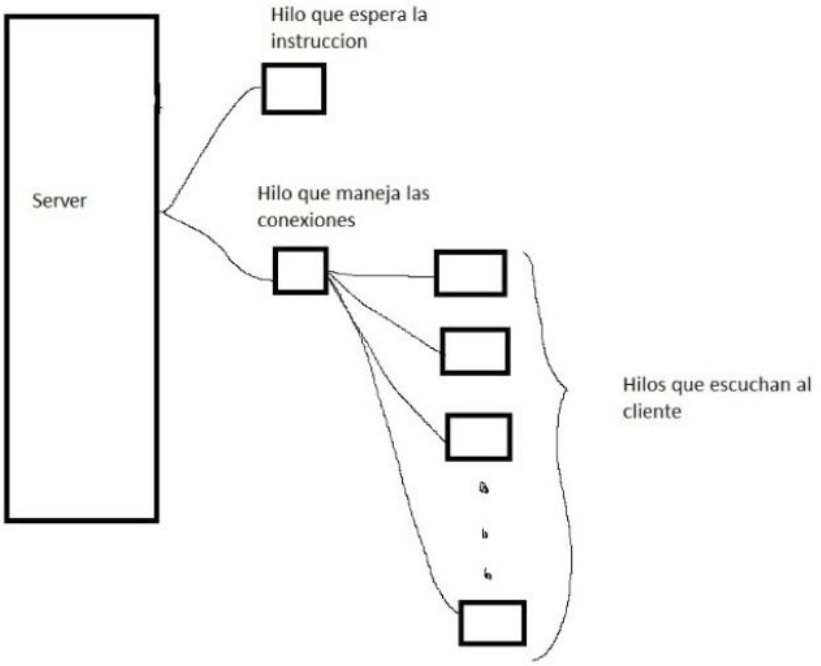
Hay dos formas de crear un nuevo hilo de ejecución. Una es declarar una clase como una subclase de Thread. Esta subclase debería anular el método run de la clase Thread. Entonces se puede asignar e iniciar una instancia de la subclase. Por ejemplo, un hilo que calcula números primos mayores que un valor establecido podría escribirse de la siguiente manera:

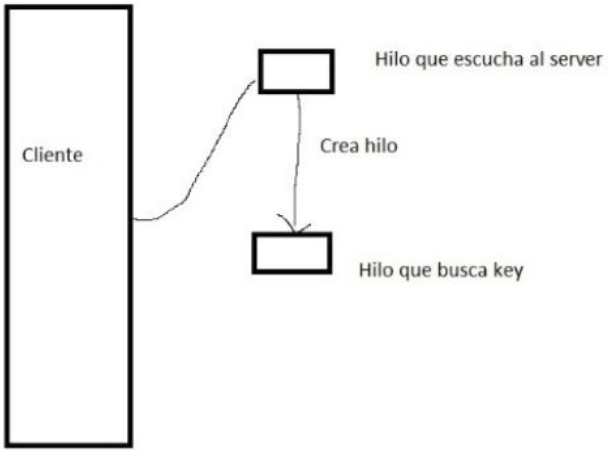
La otra forma de crear un hilo es declarar una clase que implemente la interfaz Runnable. Esa clase luego implementa el método run. Luego se puede asignar una instancia de la clase, pasarla como argumento cuando creamos Thread e iniciarlo. El mismo ejemplo en este otro estilo tiene el siguiente aspecto:

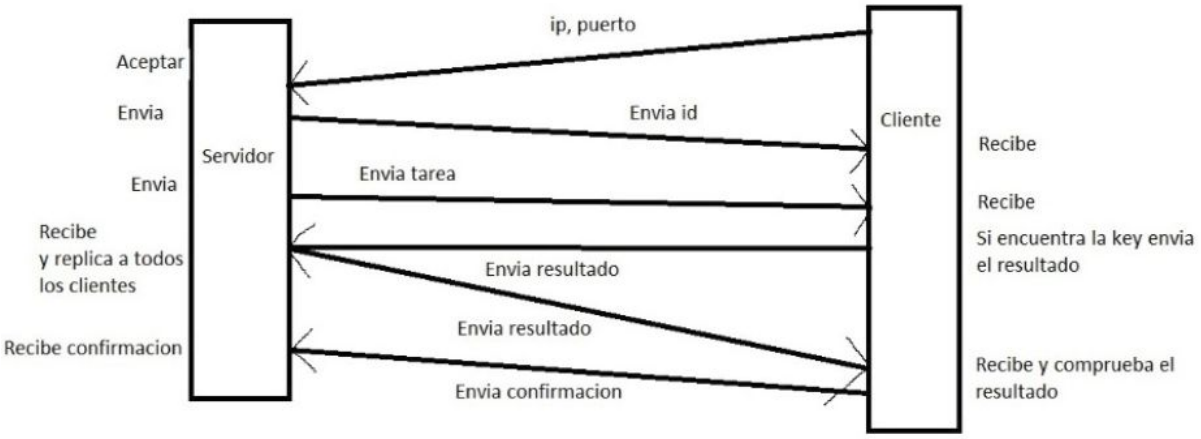
b. Cliente-Servidor mediante sockets

Se establece la comunicación cliente-servidor creando una conexión por sockets usando threads.

c. Diagrama del proceso







03. Metodología

3.1 Manejo de threads

Para la implementación del diagrama anterior se requiere trabajar estrictamente con threads ya que se generan procesos independientes para el correcto funcionamiento.

Por ejemplo:

. Proceso principal para iniciar el servidor y que reciba y envíe información.

. Procesos principales para iniciar cada cliente, que reciban y envíen información.

. Proceso para aceptar nuevas conexiones por socket en el servidor.

. Proceso para cada conexión por socket creada por cada cliente.

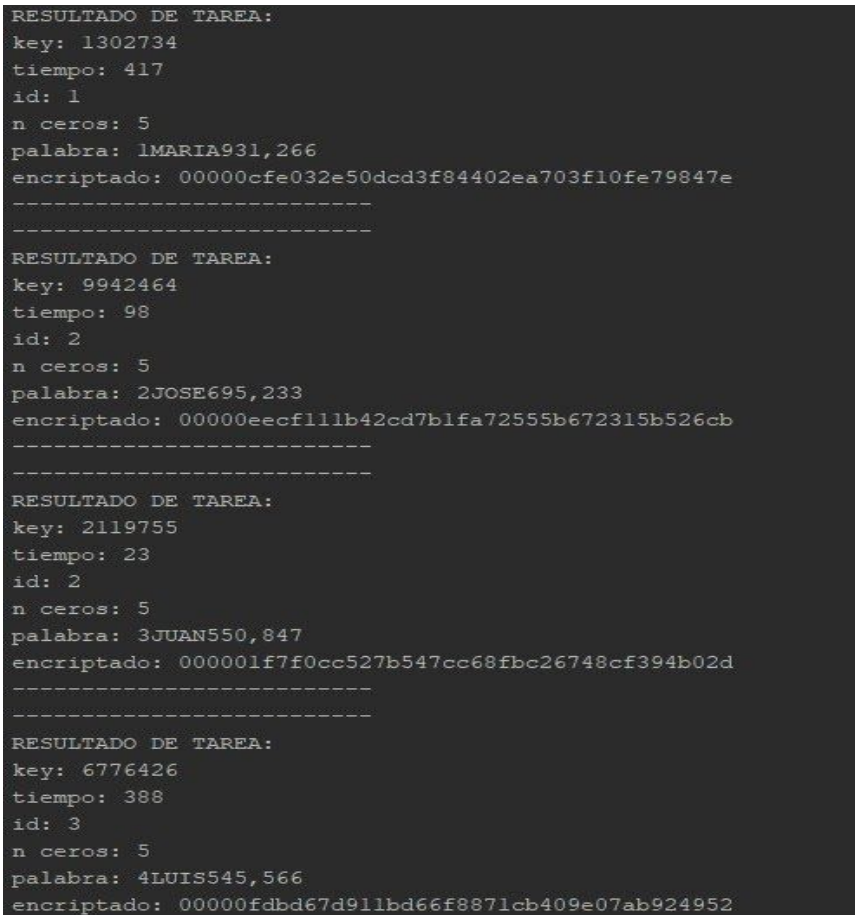
. Proceso para la búsqueda de claves de cada cliente(minero).

3.2 Comunicación entre clientes

Para lograr que diferentes clientes(mineros) se comuniquen de forma síncrona en la búsqueda de claves, el servidor actúa de intermediario y comunica el resultado de cada primer minero que encontró la clave a los mineros restantes para el proceso de validación.

04. Results and Discussion

Al inicio el servidor cuenta con una lista de 4 palabras que envía mediante broadcast a los mineros, se pide buscar una clave para cada palabra cuyo resultado de aplicar el algoritmo SHA-1 presente 5 ceros.



Para cada palabra se obtiene un id del primer minero que encontró la clave, la clave, el tiempo y el resultado del algoritmo SHA-1 de 5 ceros.

05. Conclusiones

- El uso de hilos permite simular el proceso de blockchain de búsqueda con varios mineros.
- El uso de sockets es una forma fácil de comunicación remota.
- El servidor permite la comunicación síncrona de todos los clientes en la búsqueda de claves.

06. Anexo Código

- https://github.com/lfloresh/CC462Concurrentes

07. Anexo Documentación

- https://docs.oracle.com/en/java/javase/14/
- https://brilliant.org/wiki/secure-hashing-algorithms/