MÉTODO DE LA INGENIERIA

Fase 2. Recopilación de la información necesaria

Tipos de redes de ordenadores

https://www.javatpoint.com/types-of-computer-network

Networks and Graphs

<https://kc.columbiasc.edu/ICS/icsfs/Networks.pdf?target=a96539a9-4cfd-44a5-baf2-fd798e1772ea>

HashMap

<https://www.geeksforgeeks.org/implementing-generic-graph-in-java/>

lista de adyacencia

<https://runestone.academy/runestone/static/pythoned/Graphs/UnaListaDeAdyacencia.html>

matriz de adyacencia

<https://www.jc-mouse.net/java/matriz-de-adyacencia-representacion-de-grafos-en-java>

Optimization algorithms for networks and graphs

<https://books.google.com.co/books?hl=es&lr=&id=pCcHSjzQ7VEC&oi=fnd&pg=PA1&dq=Networks+and+Graphs&ots=sdBjPL1xQe&sig=oa5MLEAqvLf1fXCT44vQqOhK_ZI>

Fase 3. Búsqueda de soluciones creativas

Con una lista de adyacencia donde los nodos son arreglos los cuales tienen un arraylist donde cada posición contiene un computador adyacente al nodo del arreglo de los contiene

con una matriz de adyacencia donde la matriz son los nodos que contiene el grafo y tienen un valor dependiendo de si los nodos están o no conectados

con una lista de adyacencia formado de arraylist, donde cada arraylist es un nodo y contiene otro arraylist donde cada posición contiene un computador adyacente al nodo del arreglo de los contiene de esta forma se podrán agregar fácilmente nuevos nodos al grafo.

Con una matriz de adyacencia de arraylist, donde cada arraylist es un nodo del grafo y en otro arraylist son los mismo nodos, donde dependiendo de si tiene o no conexión tendrá un valor u otro, de esta forma se podrán agregar fácilmente nuevos nodos al grafo.

Como un hashmap donde se guarden los nodos con sus conexiones.

Fase 4. Transición de la formulación de ideas a los diseños preliminares

Descartamos la lista y matriz de adyacencia ya que resulta complicado agregar nuevos nodos y eliminarlos al implementarlos con arreglos

Lista de adyacencia por medio de arraylist de modo que se pueda eliminar o agregar directamente un nuevo nodo simplemente eliminar o agregar un elemento a la lista de nodos y agregar o eliminar sus conexiones con nodos adyacentes

Una matriz de adyacencia implementada con arraylist de modo que se puede eliminar o agregar un nodo simplemente eliminando o agregando una fila y columna a la matriz

Fase 5. Evaluación y selección de la mejor solución

Eficacia: la idea cumple con las funciones requeridas

Eficiencia: la idea realiza las operaciones en poco tiempo

Facilidad del algoritmo: el algoritmo es sencillo de implementar

Realizar los grafos con arrayList es la mejor opción.