Trabajo Práctico Obligatorio

****

Materia: Programación III

Curso: Viernes - Turno mañana

Profesor: Wehbe Ricardo

Integrantes:

* Bandoni Ivo - Nro. De Legajo: 1155345
* Quiroga Valentín - Nro. De Legajo: 1139886
* Rosello Mateo - Nro. De Legajo: 1154422

Introducción

Problemática planteada

Una compañía de logística debe decidir si construir o no ocho centros de distribución para almacenar y transportar la materia prima agrícola suministrada por sus 50 clientes ubicados en diferentes provincias del país. Cada cliente produce una cantidad anual de materia prima, la cual es comprada por la compañía. Con el objetivo de optimizar los costos operativos, la empresa analiza la ubicación estratégica de los centros de distribución a lo largo de rutas o vías férreas para facilitar el transporte hacia los puertos de exportación. Cada centro de distribución tiene un costo anual fijo, independiente del volumen de materias primas que maneje.

Estrategia para resolver la problemática

Para abordar la problemática, inicialmente generamos una matriz que actúa como un mapa indicando las conexiones entre clientes y centros. El objetivo es determinar el costo final para cada cliente, considerando el transporte de su mercadería hasta el puerto. Para lograr esto, creamos un objeto de tipo “Grafo”, el cual lee los archivos con los datos del problema (archivos de texto) y almacena las conexiones entre los nodos. Luego implementa el algoritmo Dijkstra, que le permite calcular el costo mínimo de cada cliente a cada centro. Este algoritmo examina tanto rutas directas (si existen) como aquellas que requieren pasar por nodos vecinos para alcanzar el valor mínimo.

Una vez que hemos registrado los caminos mínimos de cada cliente a cada centro en la matriz, sumamos el costo de transporte de cada centro hasta el puerto. Para obtener los costos totales de cada cliente a cada centro, es decir, los valores finales de la matriz, multiplicamos los resultados obtenidos por el volumen de producción de cada cliente. Tanto la suma del costo del centro al puerto, como la multiplicación del volumen de cada cliente son realizados por métodos implementados en el grafo.

En una lista aparte almacenamos los costos fijos que tiene cada centro para utilizarlos en cálculos posteriores.

Teniendo los costos mínimos finales que tiene cada cliente para que su materia prima llegue al puerto el siguiente paso es realizar una serie de cálculos que nos permitirán tomar las decisiones de qué centro construir y qué centro no más adelante. Estos cálculos se implementan en la clase “CO”, la cual hace referencia a los centros en sus distintos estados (puede representar a centros aprobados, desaprobados o en revisión) y cuenta con los siguientes atributos:

Lista de enteros “x”, la cual irá representando la situación de cada centro durante el análisis.

Matriz de enteros “mapa”, la cual representa a la matriz calculada por la clase “Grafo”, que es la matriz de costos finales para cada cliente a cada centro.

Lista de enteros “costosFijos”, la cual representa a la lista calculada por la clase “Grafo”, que contiene los costos fijos de cada centro.

Lista de enteros “costosMinimos”, que representa los costos mínimos de cada cliente a cada centro construido. Esta lista será utilizada para calcular el valor pesimista, conocido como “U”.

Lista de enteros “costosMinimosPosibles”, que representa los costos mínimos de cada cliente a cada centro construido y/o eventual. Esta lista será utilizada para calcular el valor optimista, conocido como “C”.

Variable de entero “U”, que representa el valor pesimista.

Variable de entero “C”, que representa el valor optimista.

Variable de entero “reduccionMinima”, que representa el ahorro mínimo para cada cliente si se construye el centro.

Variable de entero “reducción Máxima”, que representa el ahorro máximo para cada cliente si se construye el centro.

La clase “CO” cuenta con métodos que calculan las variables que no se pasan como parámetro:

setCostosMinimos -> calcula la variable “U” = (suma de los costos mínimos de cada cliente a cada centro construido) + el costo fijo de los centros construidos.

setCostosMinimosPosibles -> calcula la variable “C” = (suma de los costos mínimos de cada cliente a cada centro construido y/o eventual) + el costo fijo de los centros construidos.

setReduccionMinima -> calcula la reducción mínima = valor mínimo para cada cliente - el segundo valor mínimo para cada cliente (solo teniendo en cuenta centros construidos)

setReduccionMaxima -> calcula la reducción máxima = valor actual (si es menor al mínimo costo de los centros construidos para ese cliente) - el mínimo costo de los construidos para ese cliente.

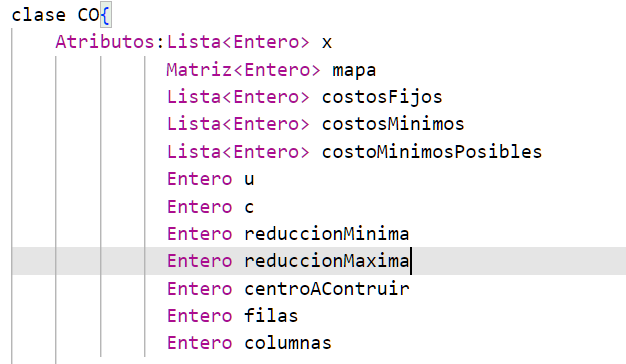
El método que usamos para analizar qué centro aprobar o desaprobar es el conocido como “Branch & Bound”, el cual se basa en un árbol binario, cuyos nodos padres representaría el análisis de construcción y sus respectivos nodos hijos las distintos estados del cada centro (aprobado o desaprobado). Este método nos permite anticipar la decisión en base a dos cálculos concretos:

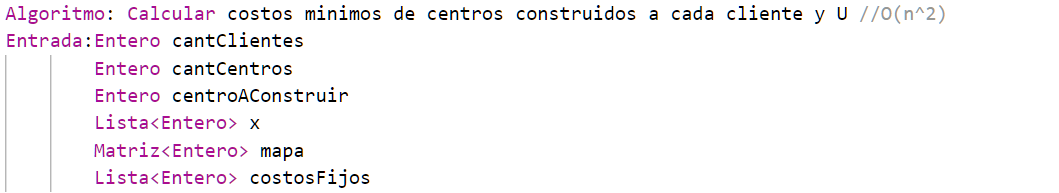
- Si el costo fijo del centro es menor a la reducción mínima, este se aprueba.

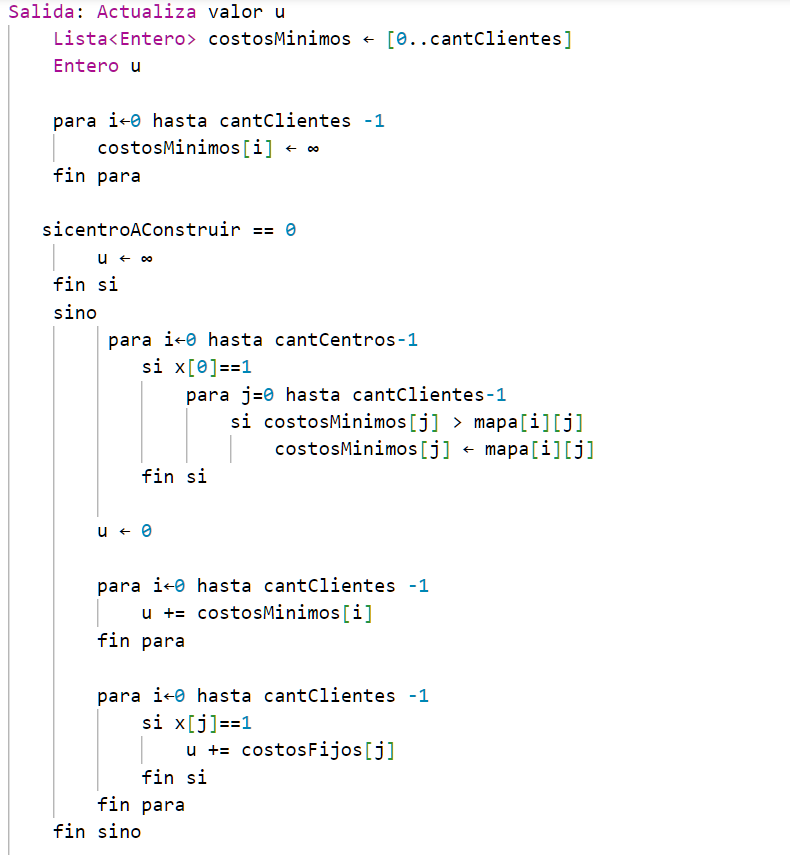
- Si el costo fijo del centro es mayor a la reducción máxima, este no se aprueba.

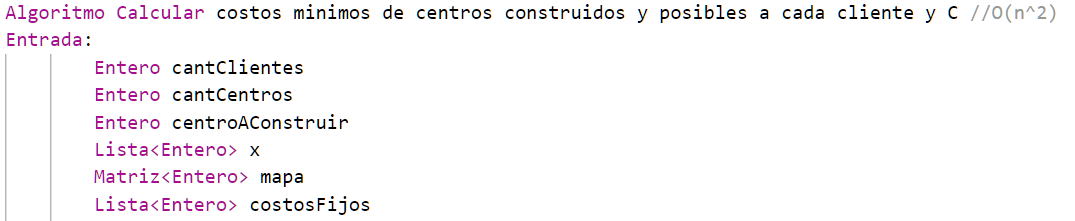
En caso de no cumplirse esas condiciones se analiza cada caso. Se llega a una conclusión cuando el valor optimista y el minimista es el mismo (C=U).

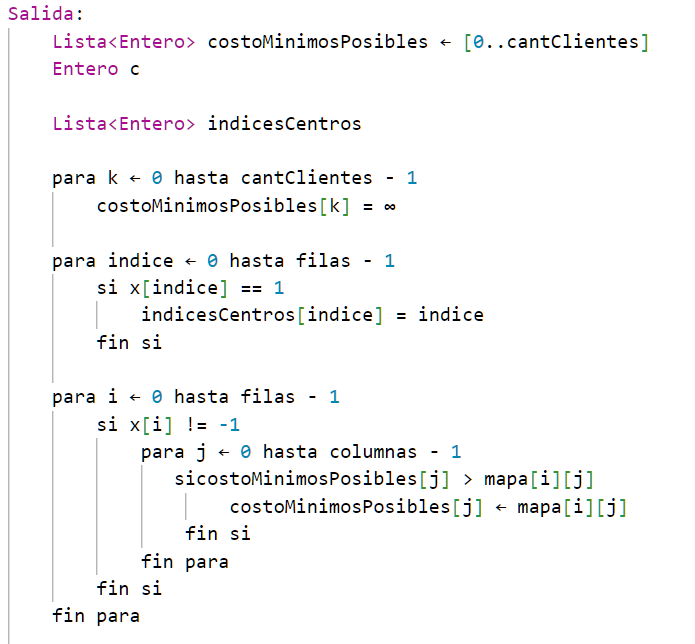
Pseudocódigo

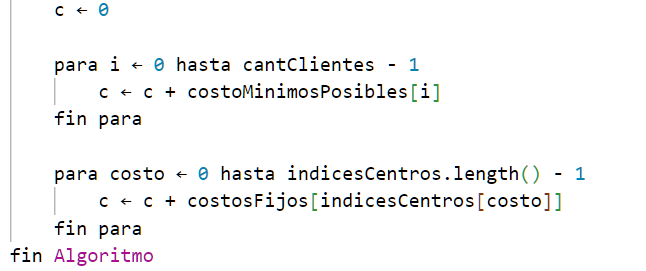


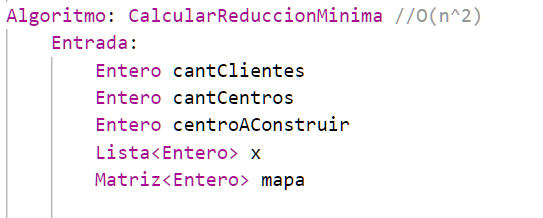


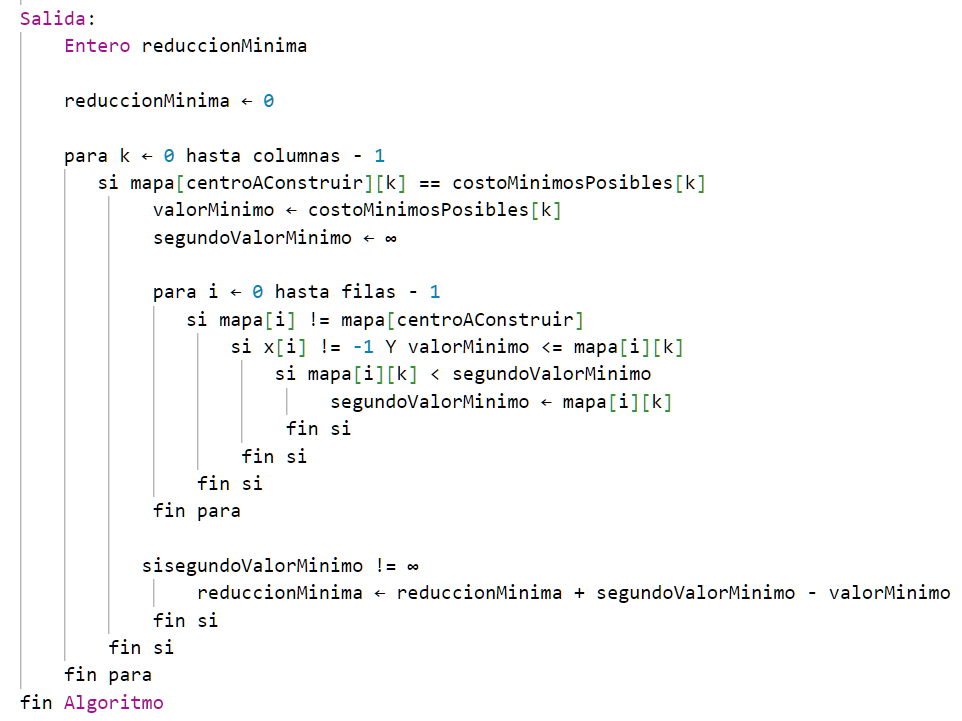


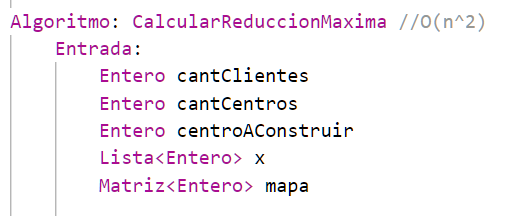


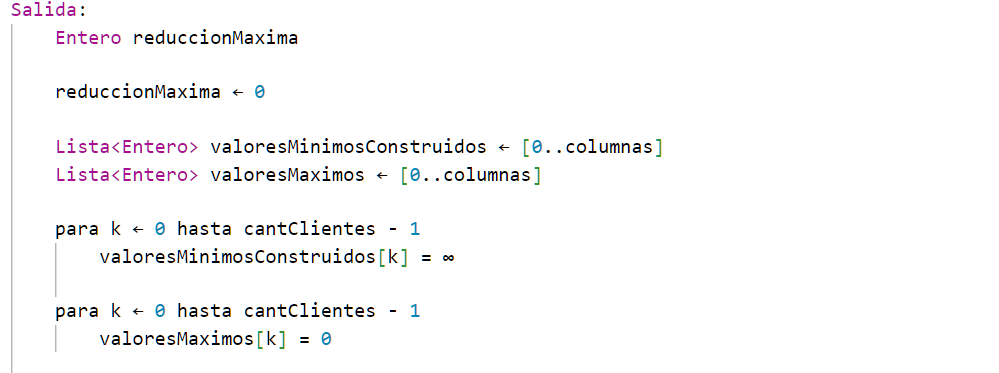


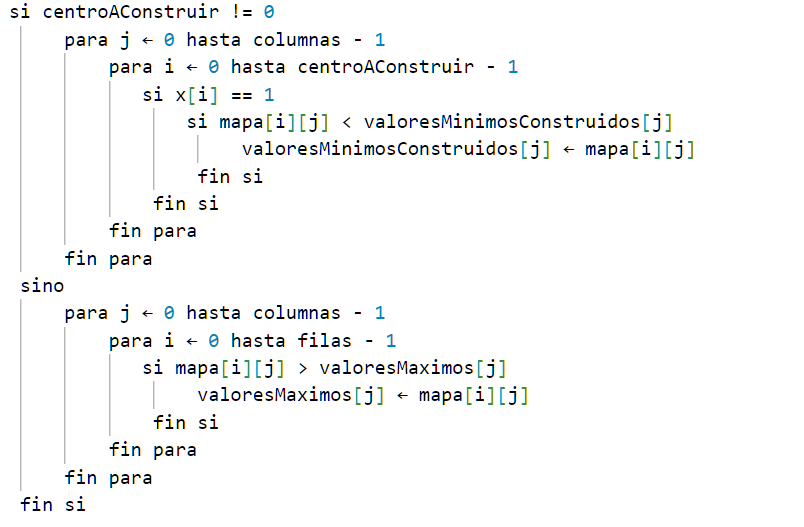


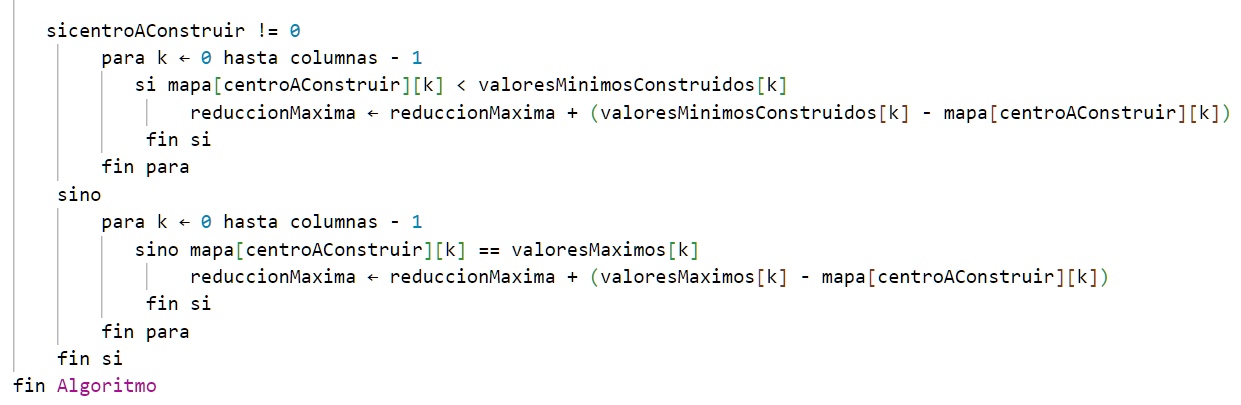


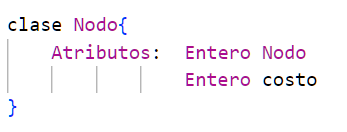


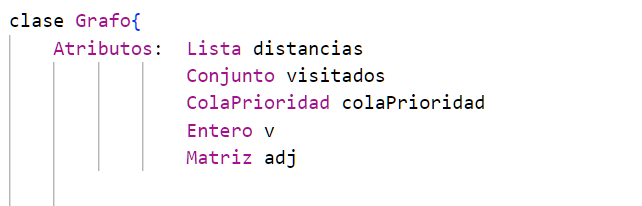


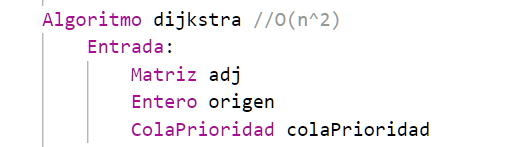


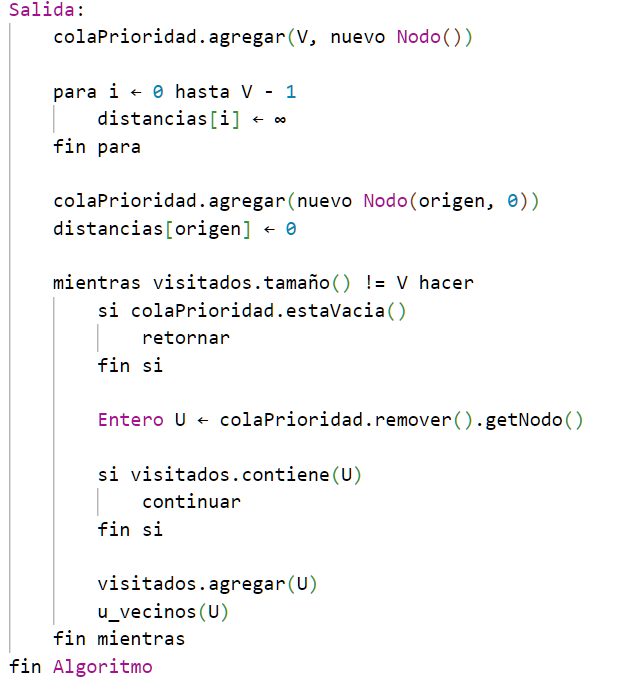


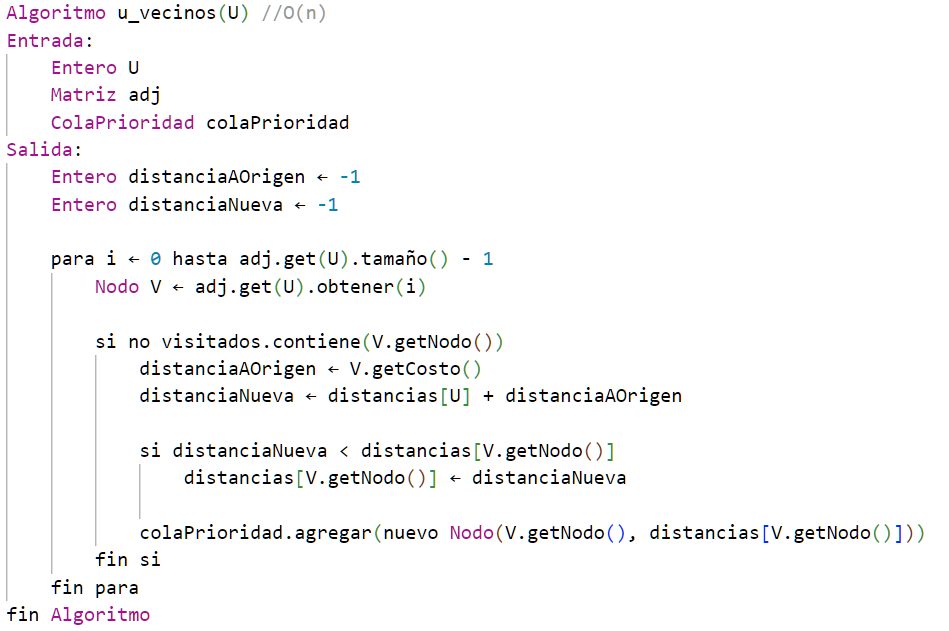


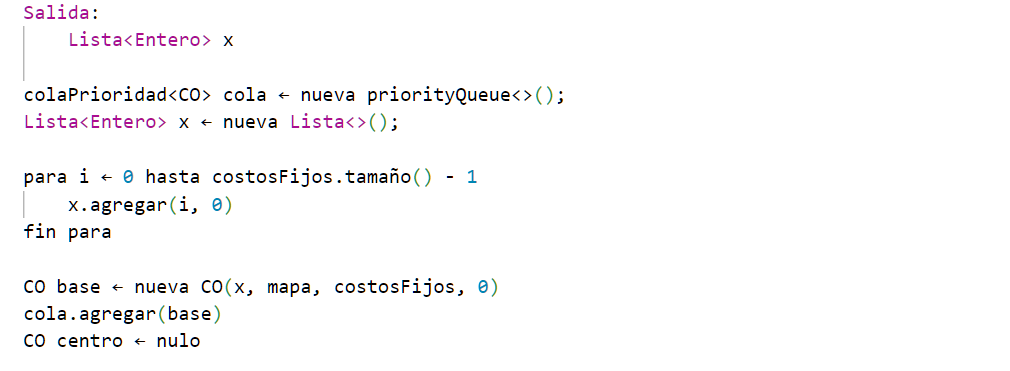


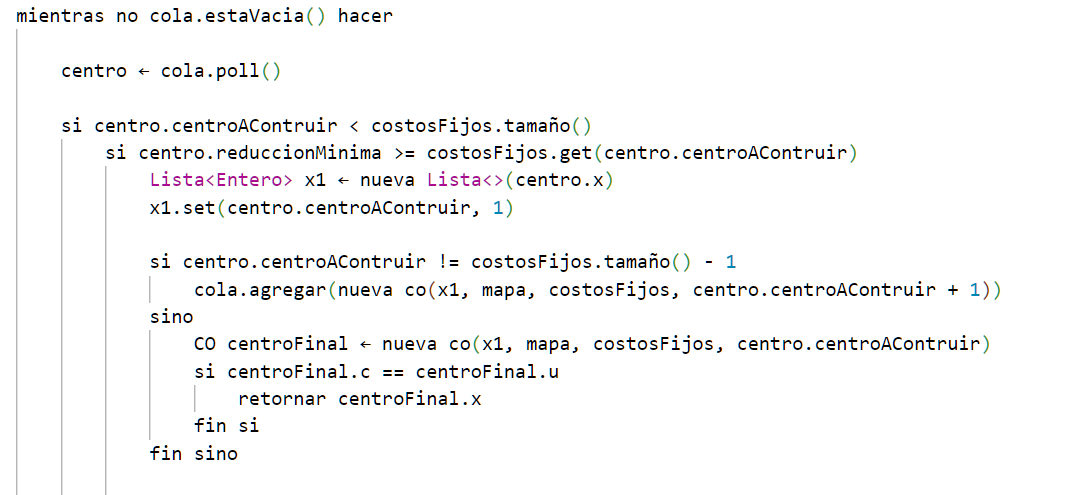


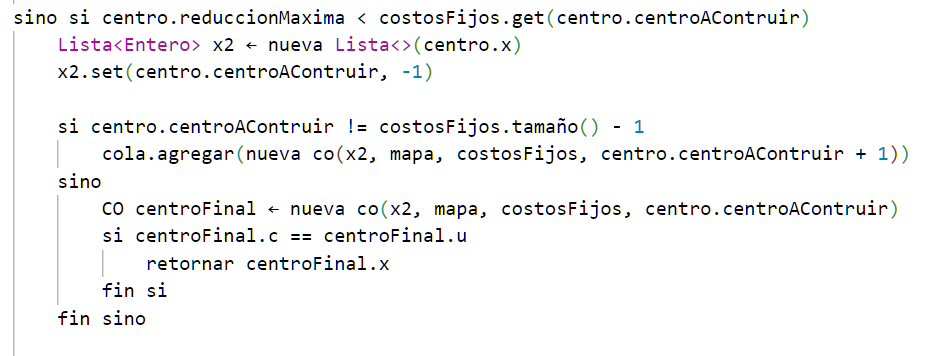


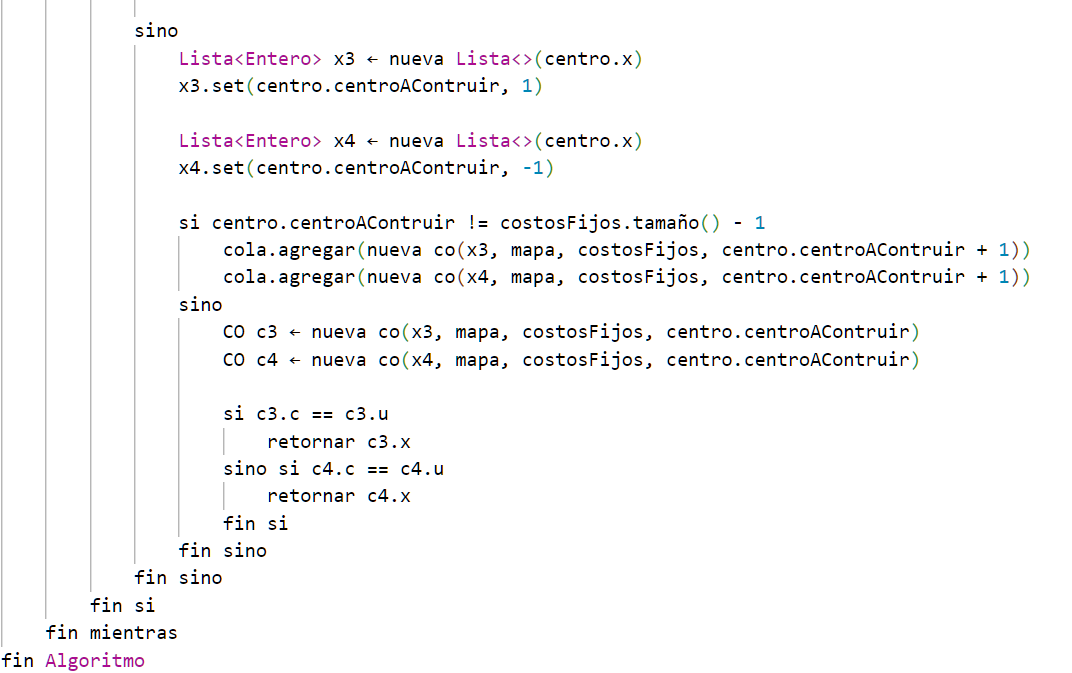












Conclusión

Para concluir, los resultados obtenidos tras implementar la estrategia planteada fueron los siguientes:

- Centros aprobados para ser construidos: centro 50, centro 52 y centro 57.

Hemos llegado a este resultado y teniendo en cuenta las herramienta que nos brindó nuestra estrategia, junto con los métodos implementados hemos llegado a la conclusión de que los centros indicados son los que más conviene construir. En el código especificamos qué centro le conviene a cada cliente.

Bibliografía:

Fuente para Dijkstra:

https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-in-java-using-priorityqueue/