

|  |
| --- |
| App metro CDMX utilizando Dijkstra |
| Estructura de datos |
| Alumno: Jesús Toxqui Ortega  Prof.: David Bárcenas |
|  |





Introducción

# ¿qué es el dijkstra?

El Algoritmo de Dijkstra, también denominado Algoritmo de caminos mínimos, es un modelo que se clasifica dentro de los algoritmos de búsqueda. Su objetivo, es determinar la ruta más corta, desde el nodo origen, hasta cualquier nodo de la red. Su metodología se basa en iteraciones, de manera tal que, en la práctica, su desarrollo se dificulta a medida que el tamaño de la red aumenta, dejándolo en clara desventaja, frente a métodos de optimización basados en programación matemática.

## ¿Qué tipos de redes pueden resolverse por medio del Algoritmo de Dijkstra?

Este algoritmo, al igual que el método de Floyd, tienen la capacidad de resolver el problema de la ruta más corta, tanto para redes cíclicas, como para redes cíclicas. De manera tal que los bucles que presente una red, no restringen el uso del algoritmo.

El algoritmo de Dijkstra hace uso y define etiquetas a partir del nodo origen y para cada uno de los nodos subsiguientes. Estas etiquetas contienen información relacionada con un valor acumulado del tamaño de los arcos y con la procedencia más próxima de la ruta.

Las etiquetas corresponden a los nodos, no a los arcos. En el algoritmo de Dijkstra, estas etiquetas son temporales y permanentes. Las etiquetas temporales son aquellas que son susceptibles de modificarse mientras exista la posibilidad de hallar para sí, una ruta más corta; de lo contrario, dicha etiqueta pasa a ser permanente.



## Edsger Dijkstra

Edsger Wybe Dijkstra fue un científico de la computación de origen holandés. Recibió un importante premio otorgado por la Asociación de Sistemas Computarizados, por sus aportes en la rama de computación distribuida durante su trabajo en el tema de auto-estabilizadores de programas de computación. Este premio anual pasó a llamarse a partir del siguiente año el Premio Dijkstra en honor del notable científico.

## Complejidad

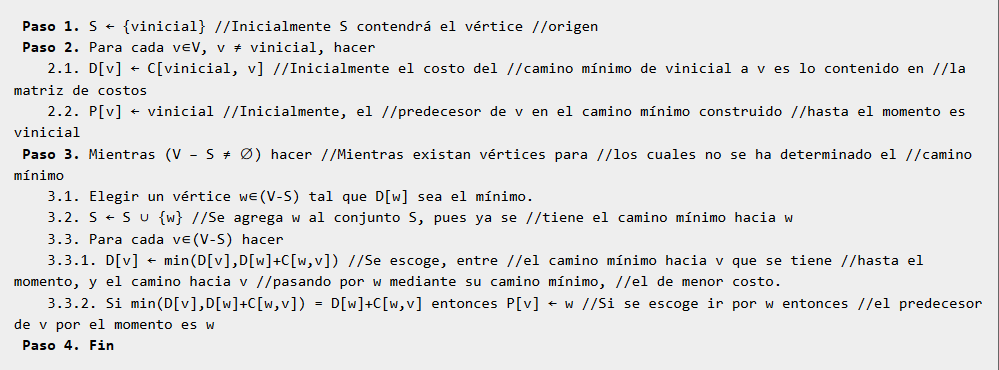
Orden de complejidad del algoritmo: O(|V|2+|E|) = O(|V|2) sin utilizar cola de prioridad, O((|E|+|V|) log |V|) utilizando cola de prioridad (por ejemplo, un montículo).

Podemos estimar la complejidad computacional del algoritmo de Dijkstra (en términos de sumas y comparaciones). El algoritmo realiza a lo más n-1 iteraciones, ya que en cada iteración se añade un vértice al conjunto distinguido. Para estimar el número total de operaciones basta estimar el número de operaciones que se llevan a cabo en cada iteración. Podemos identificar el vértice con la menor etiqueta entre los que no están en Sk realizando n-1 comparaciones o menos. Después hacemos una suma y una comparación para actualizar la etiqueta de cada uno de los vértices que no están en Sk. Por tanto, en cada iteración se realizan a lo sumo 2(n-1) operaciones, ya que no puede haber más de n-1 etiquetas por actualizar en cada iteración. Como no se realizan más de n-1 iteraciones, cada una de las cuales supone a lo más 2(n-1) operaciones, llegamos al siguiente teorema.

TEOREMA: El Algoritmo de Dijkstra realiza O(n2) operaciones (sumas y comparaciones) para determinar la longitud del camino más corto entre dos vértices de un grafo ponderado simple, conexo y no dirigido con n vértices.

## Pasos del algoritmo

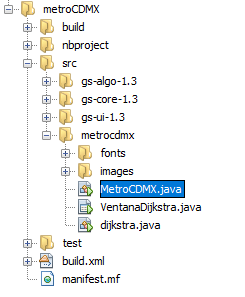
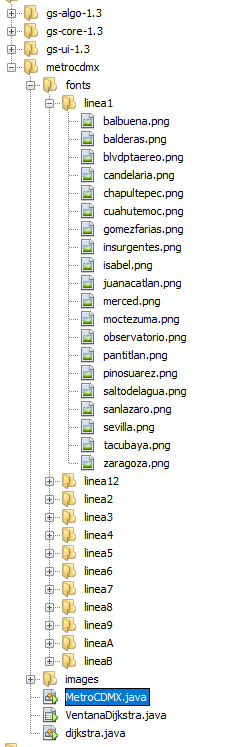
* Sea V un conjunto de vértices de un grafo.
* Sea C una matriz de costos de las aristas del grafo, donde en C[u,v] se almacena el costo de la arista entre u y v.
* Sea S un conjunto que contendrá los vértices para los cuales ya se tiene determinado el camino mínimo.
* Sea D un arreglo unidimensional tal que D[v] es el costo del camino mínimo del vértice origen al vértice v.
* Sea P un arreglo unidimensional tal que P[v] es el vértice predecesor de v en el camino mínimo que se tiene construido.
* Sea v inicial el vértice origen. Recordar que el Algoritmo Dijkstra determina los caminos mínimos que existen partiendo de un vértice origen al resto de los vértices.



## Aplicación desarrollada

Utilizando los conocimientos previos como son listas, arboles, y grafos, se planteó un problema el cual consistía en encontrar la ruta más corta entre estaciones del metro de la ciudad de México. Para dar solución a este problema se optó por utilizar el lenguaje de programación java, concretamente utilizando el Framework Graphstream.

Para esto la distribución del proyecto fue la siguiente:



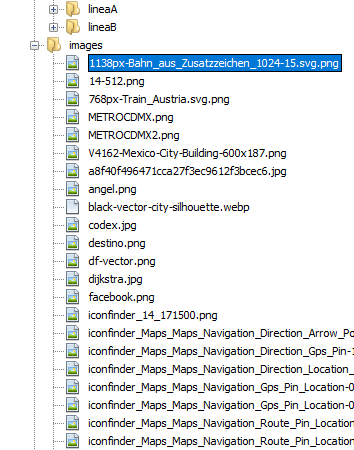


Ilustración Root de archivos del proyecto

MetroCDMX.java: este archivo contiene la información referente al llamado de la ventana. Es el archivo principal que se ejecuta antes que todos.

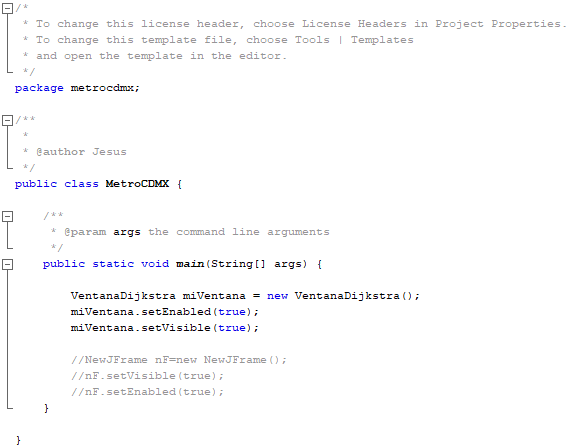


Ilustración 2 Codigo ventana MetroCDMX.java

VentanaDijkstra.java: aquí se muestra toda la información referente a interfaz y la función de cada uno de los objetos de la misma



Ilustración 3 Interfaz gráfica del proyecto

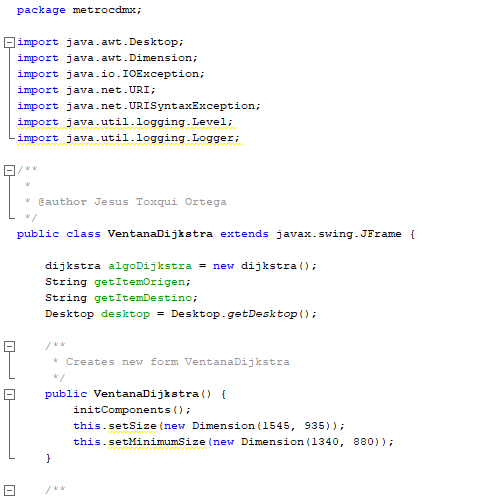


Ilustración 4 declaración de variables y objetos, así como dimensiones de la ventana



Ilustración 5 funciones del botón calcDijkstra obtiene los elementos seleccionados de cada lista desplegable y los manda a dijkstra.java

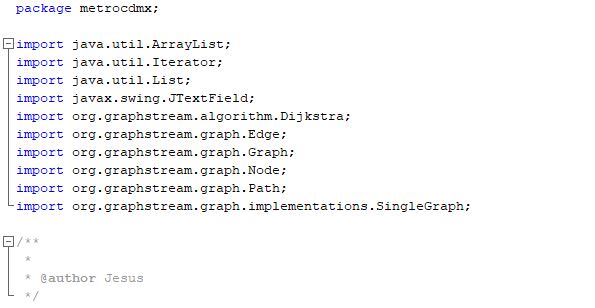


Ilustración 6 Impostación de módulos, paquetes y/o librerías



Ilustración 7 Declaración de nueva función graph, importación de estilos como atributos y adición de nodos línea 1 con posiciones en plano cartesiano (x,y)

Ilustración Obtención de nodos antes declarados y modificación de estilos con imágenes de fondo en cada nodo

Ilustración 9 Ilustración 8 Programación de Uniones entre nodos ya declarados con una distancia de 1

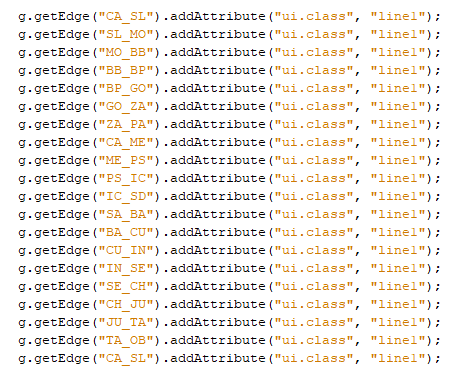


Ilustración 10 Obtención de edges ya declarados y inserción de clases como atributo

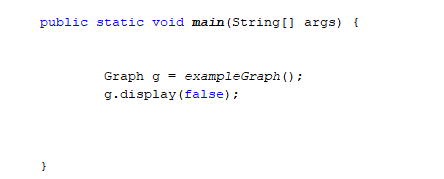


Ilustración 9 se desactiva el display en la función main



Ilustración 10 Cálculo del dijkstra obteniendo de parámetros en la función los enlazados con la ventanaDijkstra.java

## Resultados

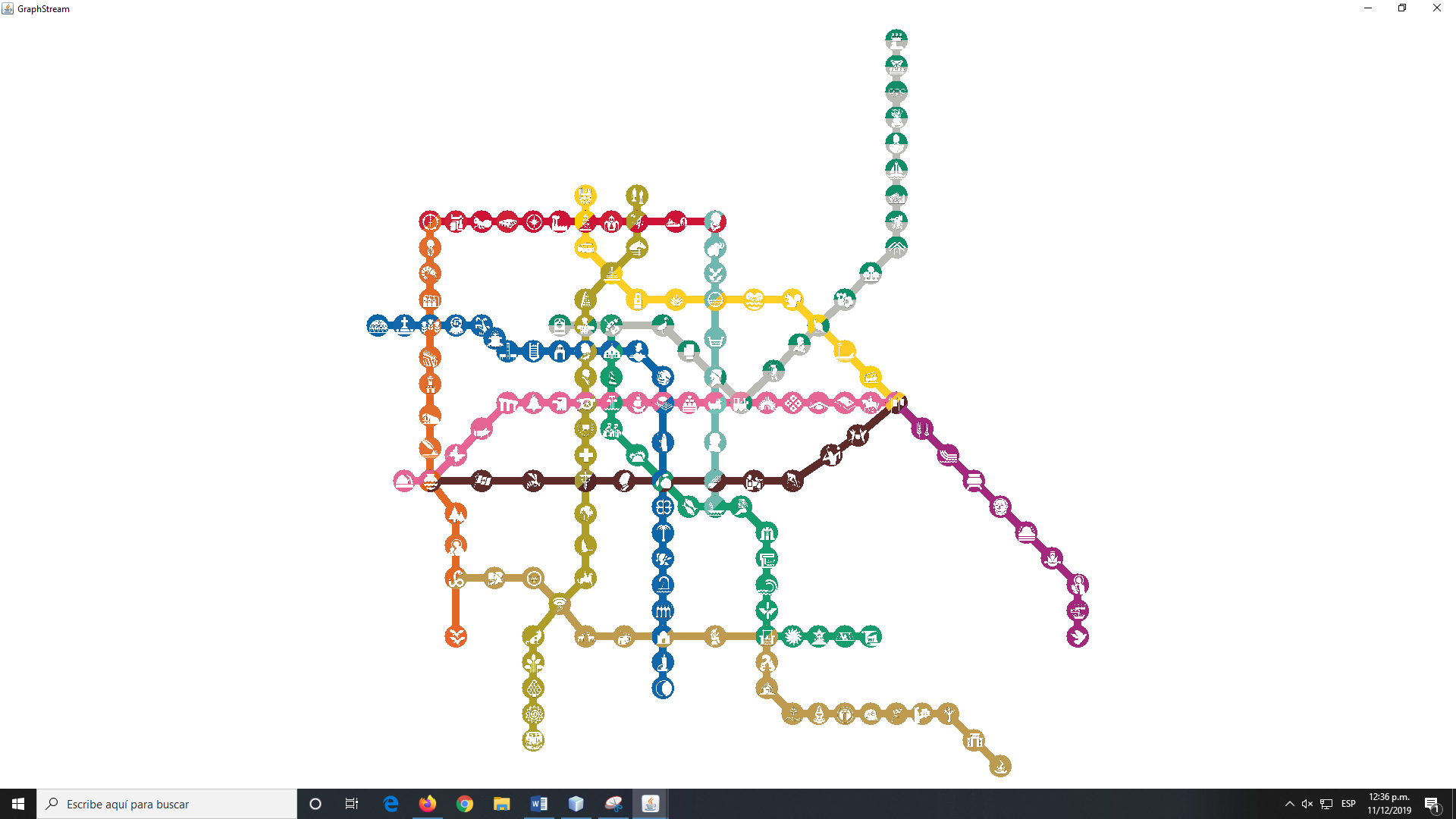


Ilustración 11 Mapa original sin rutas CORTAS

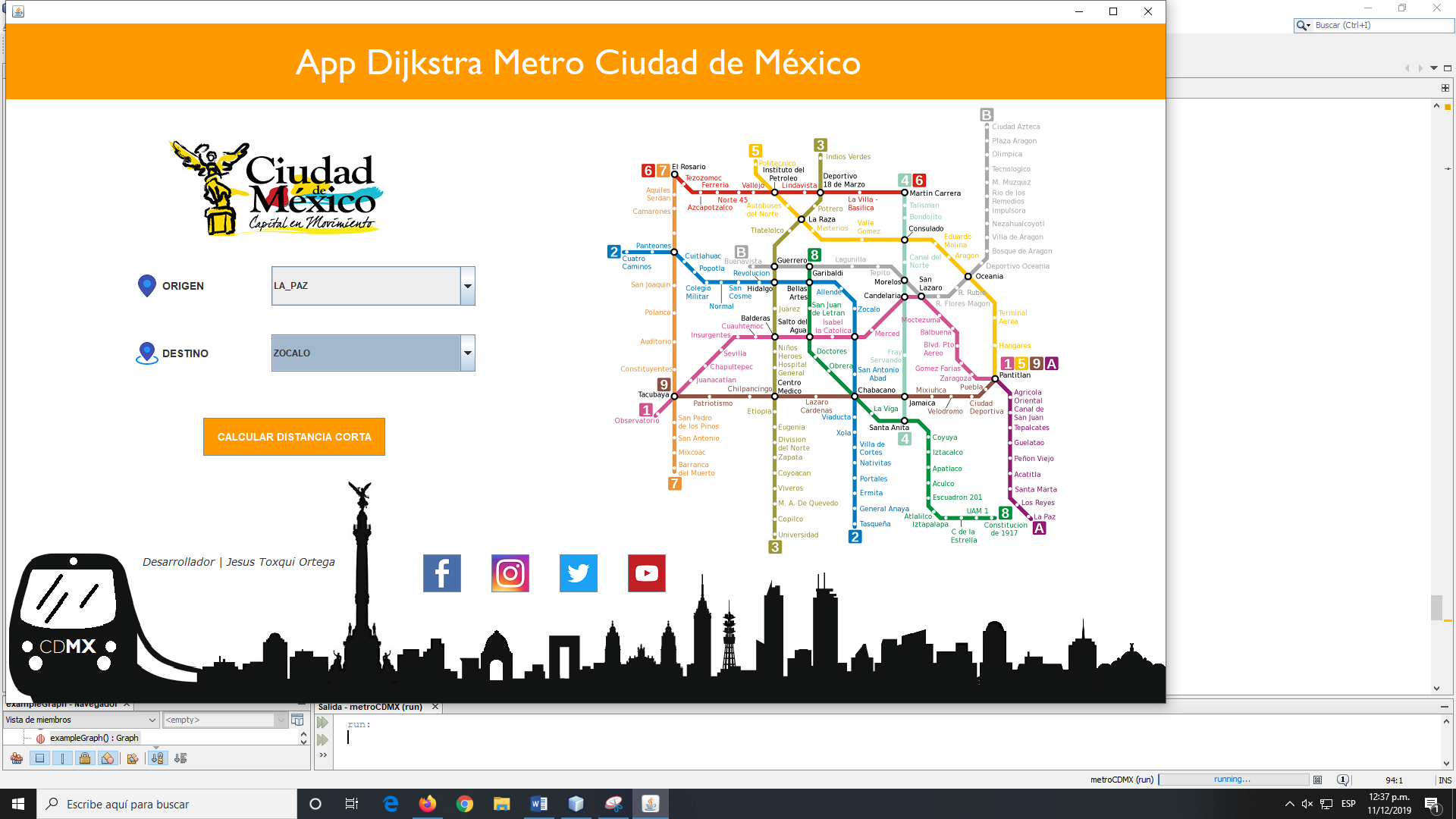


Ilustración 12 Ingreso de datos por listas desplegables

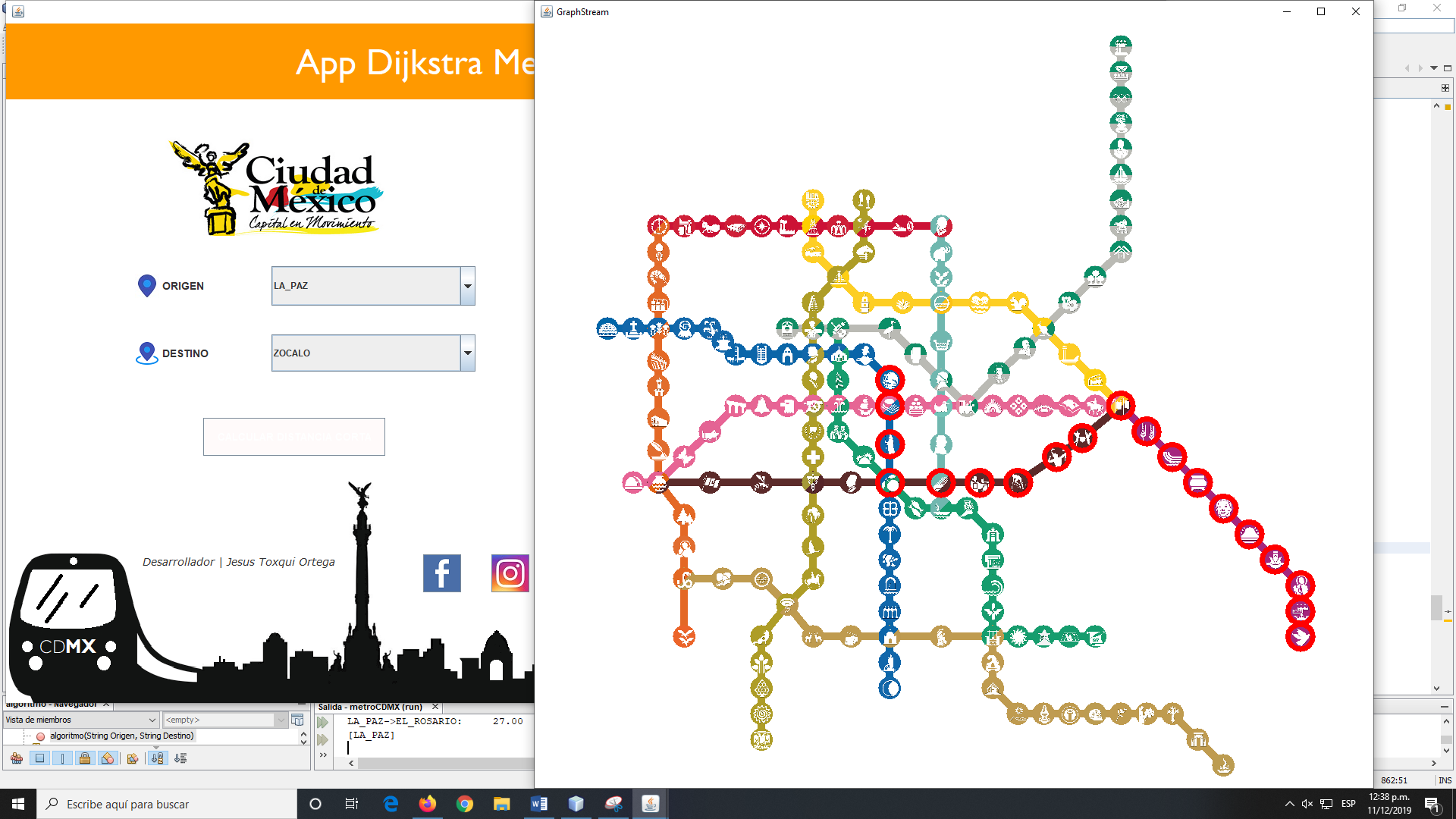


Ilustración 13 Calcula la distancia más corta Entre la estación LA\_PAZ y el ZOCALO

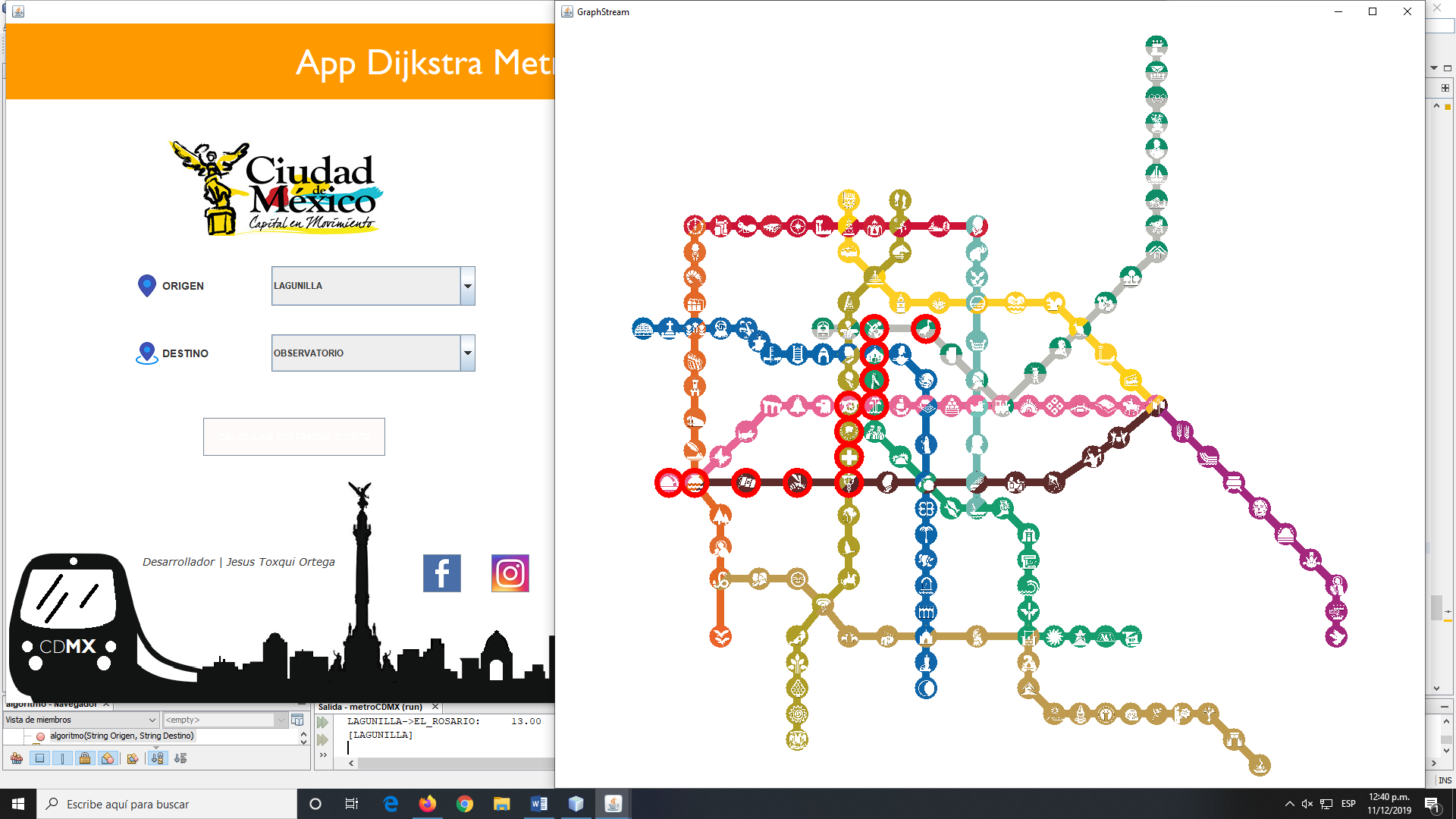


Ilustración 14 Calcula la distancia más corta Entre la estación LAGUNILLA y el OBSERVATORIO

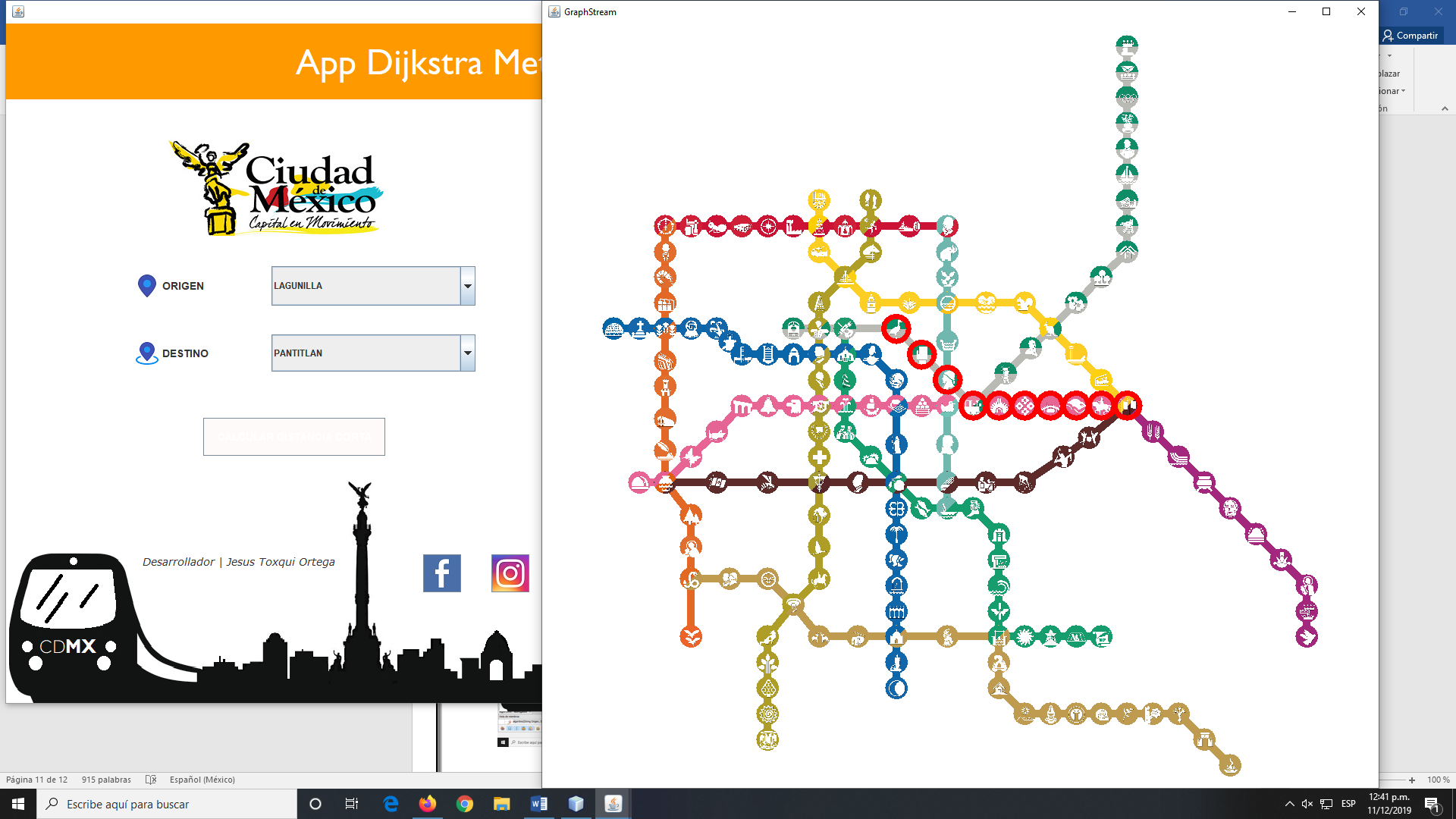


Ilustración 15 Calcula la distancia más corta Entre la estación LAGUNILLA y PANTITLAN

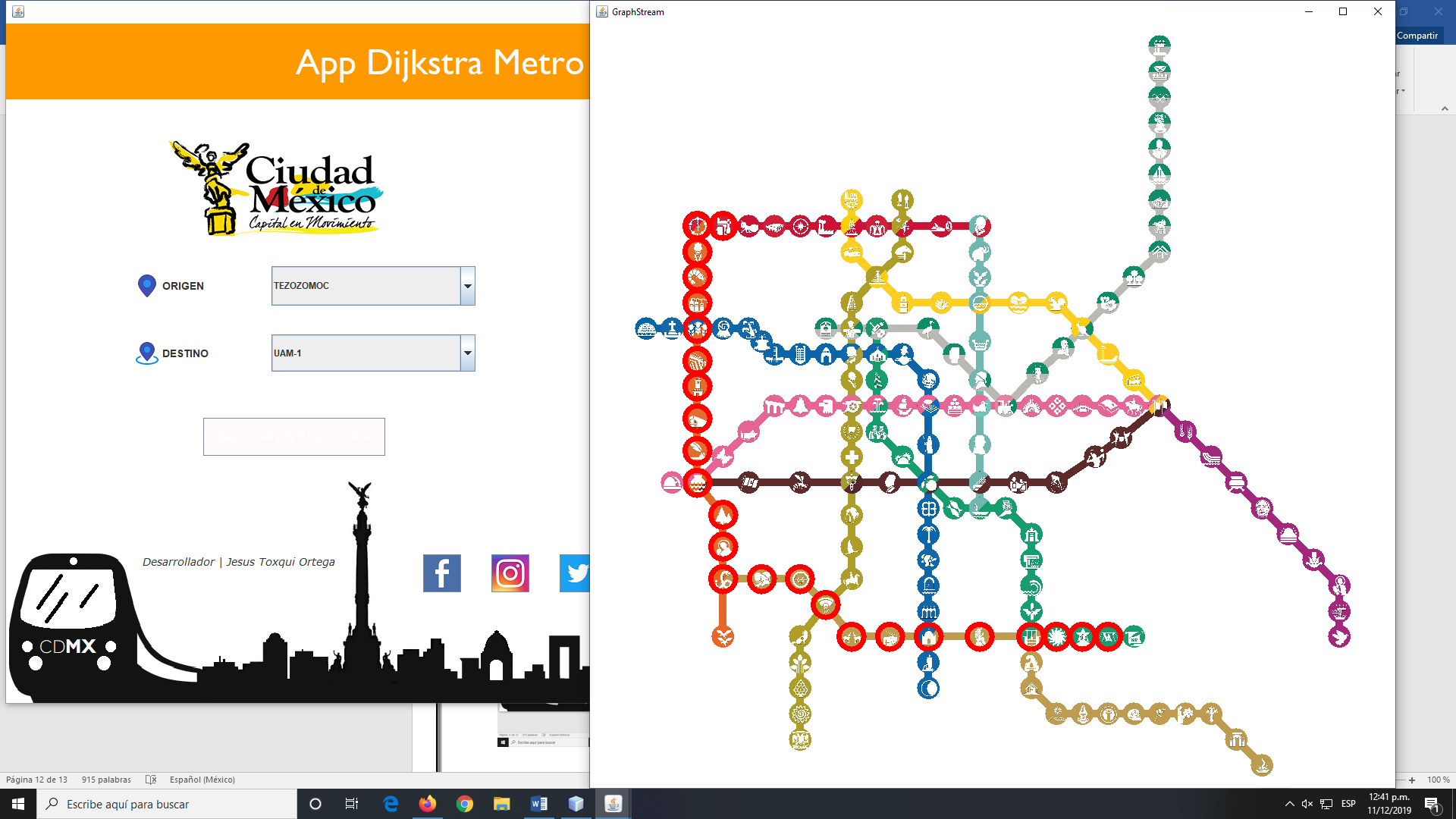


Ilustración 16 Calcula la distancia más corta Entre la estación TEZOZOMOC y UAM-1

## Otras funciones

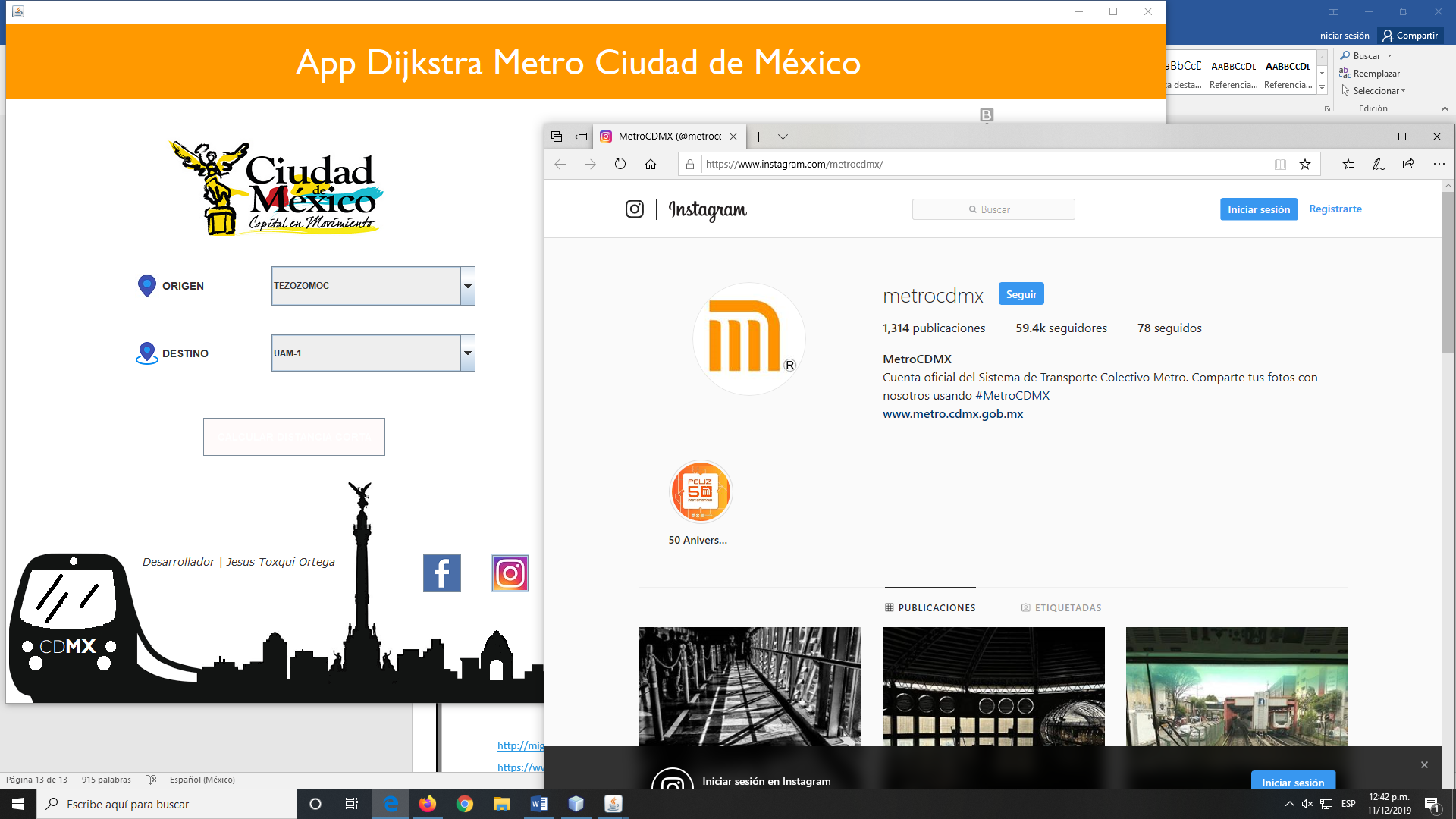


Ilustración 17 Funciones de botones de redes sociales

## Conclusión

La programación de grafos y su aplicación en la vida diaria pueda darse de muchas maneras. Sobre todo cuando se trata de representar modelos complejos con los cuales poder manipularlos o sacar provecho de ellos para resolver un problema concreto. El algoritmo Dijkstra nos ofrece buenas herramientas para el manejo de mapas, y su implementación en este proyecto nos permitió conocer su funcionamiento, así como su complejidad y al mismo tiempo darnos una idea de las cosas que se podían llegar a hacer.

## Bibliografía

Algoritmo de Dijkstra - EcuRed. (2019). Retrieved 11 December 2019, from https://www.ecured.cu/Algoritmo\_de\_Dijkstra

Edsger Dijkstra - EcuRed. (2019). Retrieved 11 December 2019, from https://www.ecured.cu/Edsger\_Dijkstra

Algoritmo de Dijkstra. (2019). Retrieved 11 December 2019, from http://migrupoc.blogspot.com/2012/10/algoritmo-de-dijkstra.html

Graphstream - Google Search. (2019). Retrieved 11 December 2019, from https://www.google.com/search?client=firefox-b-d&q=Graphstream