Proyecto:

Metro Japón

Grupo 8

Diego José Abengózar Vilar (z17m063)
Alejandro García Castellanos (z17m008)
Ignacio Javier Encinas Ramos (z17m057)
Alejandro Gil Ferrer (z17m017)
Jaime Vallejo Benítez-Cano (z17m002)

Índice

1. Modelización

Para la implementación de la estructura del metro aportada en la descripción del proyecto generamos un grafo en el cual los nodos representarán las estaciones y las aristas las líneas del metro. Los pesos de estas últimas equivalen al tiempo medio de ir de una estación (nodo) a otra.

Figura 1

Caso particular para transbordos

La principal dificultad aparece al tratar aquellos nodos que representan estaciones que ofrecen la posibilidad de transbordo. Para solucionarlo, ponemos a cero el tiempo entre el nodo simbólico y los nodos de los transbordos para que, en el caso de que este sea el nodo inicial, el algoritmo trate de forma equitativa todas las líneas que convergen en él.

Antes de la ejecución se eliminarán todos los nodos simbólicos que no sean el origen y el destino para que no afecte al resto situaciones.

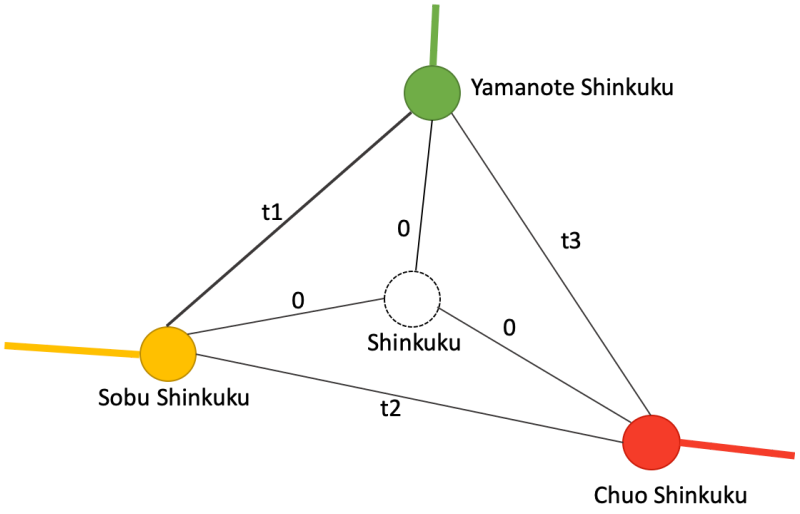


Figura 2

2. Obtención de datos

Los datos de las distancias entre estaciones se han obtenido de la página de Wikipedia correspondientes a cada una de ellas:

Para Yamanote: https://en.wikipedia.org/wiki/Yamanote_Line
Para Chūō: https://en.wikipedia.org/wiki/Ch%C5%AB%C5%8D_Line_(Rapid)
Para Sōbu: https://en.wikipedia.org/wiki/Ch%C5%AB%C5%8D-S%C5%8Dbu_Line

La información acerca de los tiempos y las distancias entre transbordos se han obtenido de las aplicaciones: Google Maps y Train Info de JR-East. Usamos la velocidad media de 5km/h de una persona andando para obtener la distancia.

Velocidad media al andar: https://es.wikipedia.org/wiki/Kil%C3%B3metro_por_hora

Tras la recopilación de datos calculamos el tiempo entre estaciones y el tiempo de transbordo de la siguiente manera:

$$T_{entre\_estaciones} = (D_{entre\_estaciones} / V_m) + t_m$$

$$T_{transbordo} = T_{entre\_líneas} + t_L$$

Donde V_m representa la velocidad media del tren de cada respectiva línea, t_m el tiempo medio de espera de un tren en una estación y t_L el tiempo medio de llegada del tren en la nueva línea.

3. Heurística

Se ha tomado como **heurística, el tiempo medio** que tarda el tren más rápido (se trata del que tiene una velocidad de 100 km/h) en ir en línea recta desde una estación a otra.

Para la obtención de la **distancia** en línea recta, primero se han buscado las coordenadas decimales de todas las **estaciones** en ***Google Maps*.**
A continuación se ha aplicado la **fórmula de *Haversine*** que permite calcular la **distancia entre dos puntos en una esfera.**

Fórmula de *Haversine*

$$a = sin^2(\frac{\varphi_B - \varphi_A}{2}) + cos(\varphi_A) * cos(\varphi_B) * sin^2(\frac{\lambda_B - \lambda_A}{2})$$
$$c = arctan2(\sqrt{a}, \sqrt{1-a})$$
$$d = R * c$$

$$Donde:$$
$$\varphi_i \rightarrow latitud$$
$$\lambda_i \rightarrow longitud$$
$$R \rightarrow radio\ de\ la\ Tierra$$

Figura 3
Referencia:: https://community.esri.com/groups/coordinate-reference-systems/blog/2017/10/05/

Dado que la **distancia entre nuestros nodos es menor o igual a la distancia del trayecto** del tren entre estaciones y que la **velocidad usada es mayor o igual a la velocidad media** en el trayecto entre estaciones o transbordos; podemos asegurar que $h(n) \leq h*(n)$, y por lo tanto, **nuestra modelización es válida** para aplicarle el **algoritmo A***.

4. Implementación (Lenguaje de programación y librerías)

Para el desarrollo de está práctica hemos optado por el lenguaje de programación ***Python*.** Las diversas razones han sido las siguientes:

- Su **versatilidad** y facilidad para la **adaptación al proyecto.**
- La **multitud de librerías útiles** que nos han permitido manejar, analizar y mostrar los datos fácilmente. De entre las cuales:
    - TKinter
    - MatPlotlib
    - NetworkX
    - CSV
- La **facilidad** a la hora de hacer ***test* y *debug,*** así como para la **implementación de la interfaz gráfica.**
- Además, nos ha permitido aprender a hacer un proyecto en *Python*

5. Aplicación del algoritmo A*

El algoritmo A* que se ha implementado consta de **tres entradas** (origen, destino y la opción de si se desea calcular la ruta con el mínimo número de transbordos), **dos fuentes de datos** (Un CSV con la heurística de las estaciones y el grafo basado en el mapa de las estaciones proporcionado) y **dos salidas** (El camino mínimo que se ha de seguir y el tiempo estimado que se tardará en seguirlo).

A continuación se dan algunos detalles y explicaciones sobre los elementos usados en el algoritmo A*:

- **La función g** viene definida por el **tiempo desde el origen hasta el nodo actual.**
- **La función h** representa a la **heurística** (es decir, la estimación de tiempo hasta el nodo destino).
- **La función f** se trata de la **suma de g y h**
- **El puntero direccionador** señalará al anterior nodo
- **El grafo G** ha sido implementado gracias a la librería ***networkX*.**
- **La lista abierta** ha sido implementada como una ***Priority Queue*** con la librería **heapq**. La lista abierta guarda las duplas **< f , estación >** donde tienen **mayor prioridad las que tienen menor f.**
- **La lista cerrada** almacena los **nodos ya visitados.**

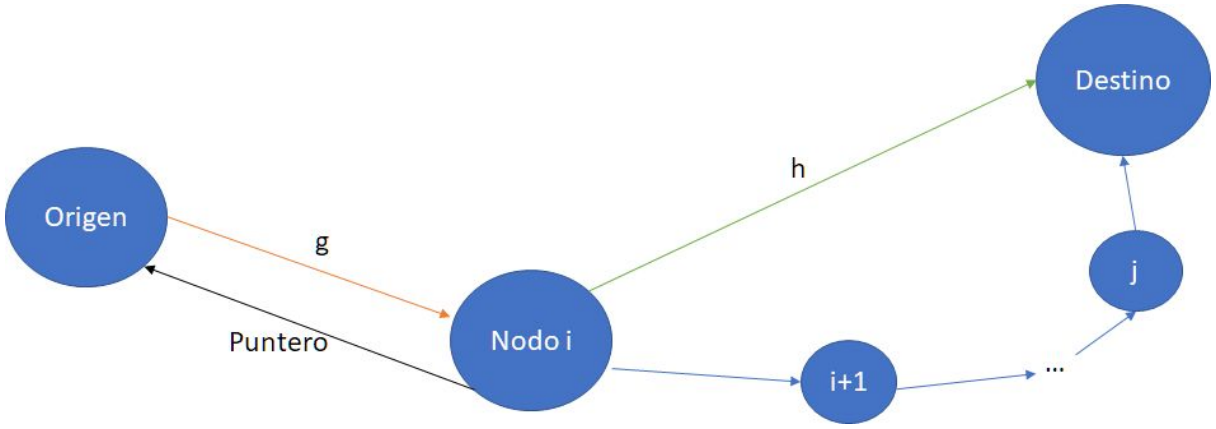


Figura 4

- Pseudocódigo

Inicialización

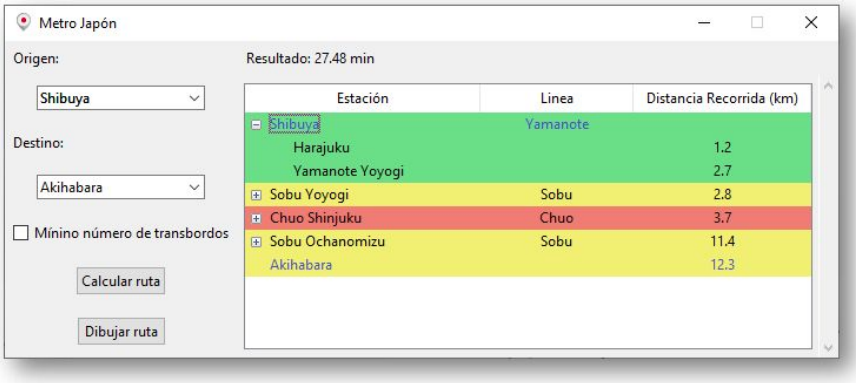
Mientras (no *hemosLlegado*)
Sacar el más prioritario de *ListaAbierta* y meterlo en *ListaCerrada*
Si es el destino -> *hemosLlegado*
Si no, para cada *nodo adyacente* de G (que no sea antecedente)
-> Calcular su f y g (y añadir penalización a transbordos)
Si está en *ListaAbierta*, si el nuevo f es mejor
-> Actualizar f, g, el puntero direccionador y *ListaAbierta*
Si no está,
-> Actualizar f, g, el puntero y añadirlo a *ListaAbierta*
Volver

Calcular ruta usando los punteros direccionadores

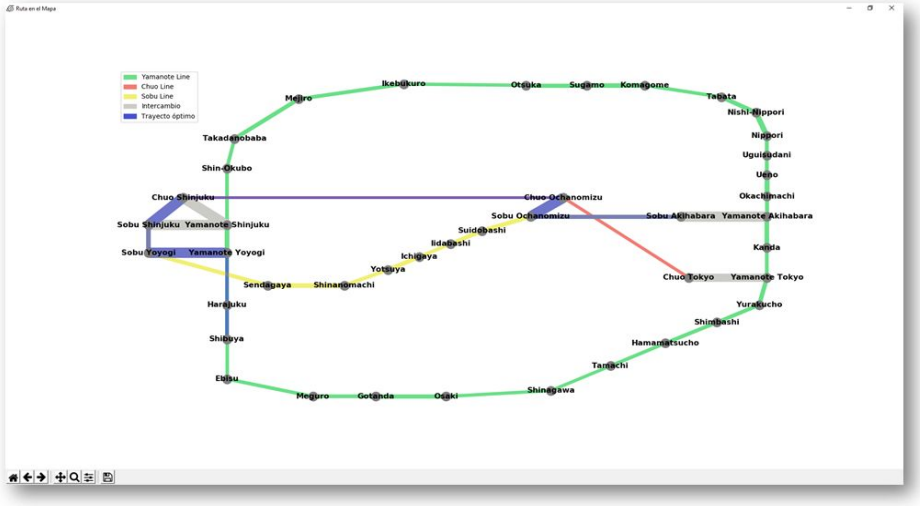
6. Interfaz

Hemos diseñado una interfaz de tal forma que sea muy sencilla de utilizar. En el margen izquierdo se encuentran las casillas de selección para el origen y el destino de nuestra ruta. Podremos seleccionar cualquiera de las estaciones disponibles del metro Japón.
Inmediatamente debajo, podremos seleccionar con un botón si se quiere calcular el mínimo número de transbordos posibles de la ruta especificada.

A continuación, se dispone del botón “Calcular ruta” que va a calcular y mostrar la lista de estaciones que se encuentran en el camino mínimo. Estas aparecerán representadas en la parte derecha de la pantalla junto con el nombre de la línea, su color, la distancia recorrida entre estaciones en kilómetros, y el tiempo que se estima que se va a tardar en realizar el trayecto.
Finalmente, pulsaremos en “Dibujar ruta” para que la ruta se muestre representada sobre el grafo que aparecerá en una nueva ventana.



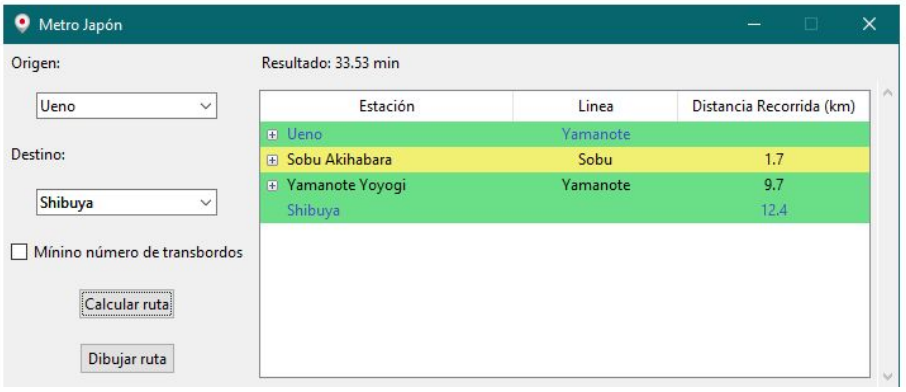
En la siguiente imagen podemos ver el grafo junto a una pequeña leyenda que indica de qué color son las líneas del metro. En la esquina inferior izquierda existen diversas opciones como hacer zoom sobre el grafo o guardar la imagen de la ruta en nuestro ordenador.



7. Demostración del funcionamiento

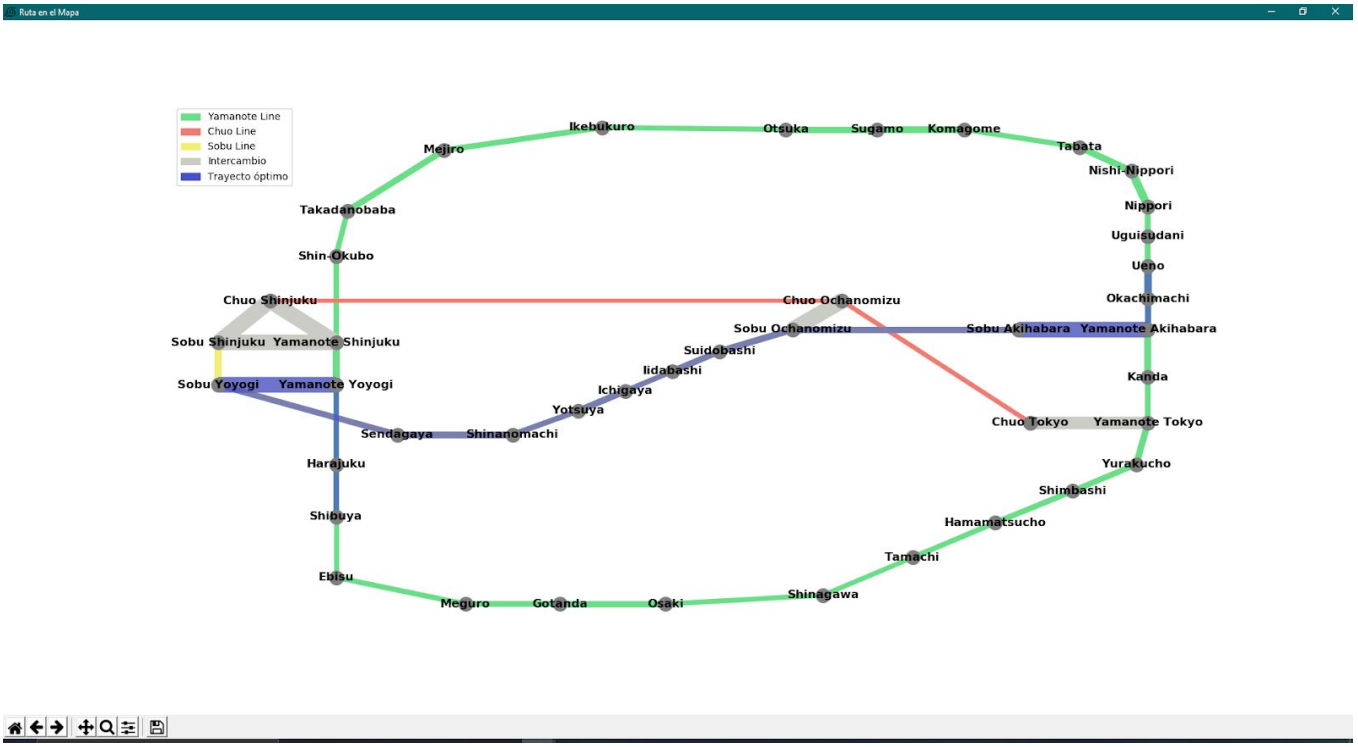
Describimos a continuación un ejemplo que muestra el funcionamiento del programa. Para ello, hemos escogido una ruta larga donde se puede observar con claridad cómo está funcionando el algoritmo. La ruta sleccionada es Ueno-Shibuya.

El primer paso es seleccionar el origen y el destino en las casillas de selección. Y pulsar “Calcular ruta”. Como se puede observar aparecen de forma comprimida a la derecha, las líneas que debemos tomar para hacer el viaje.

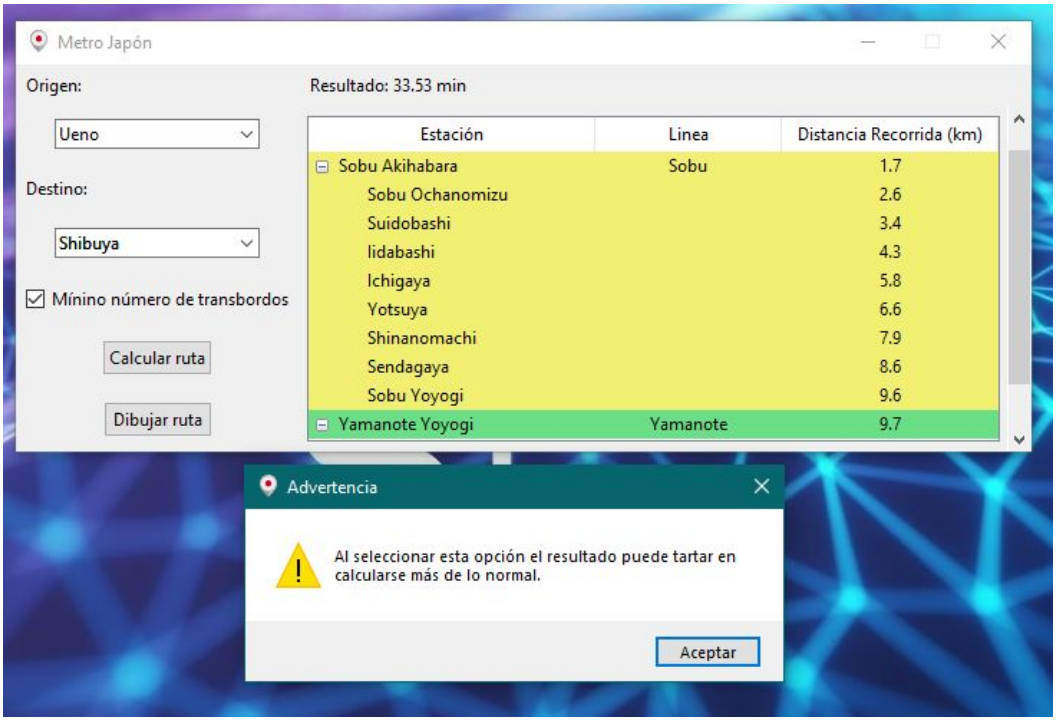


Con el desplegable se pueden ver en extensión todas y cada una de las estaciones que visitaremos. Esta opción es útil ya que también nos proporciona el tiempo estimado entre las distintas estaciones.

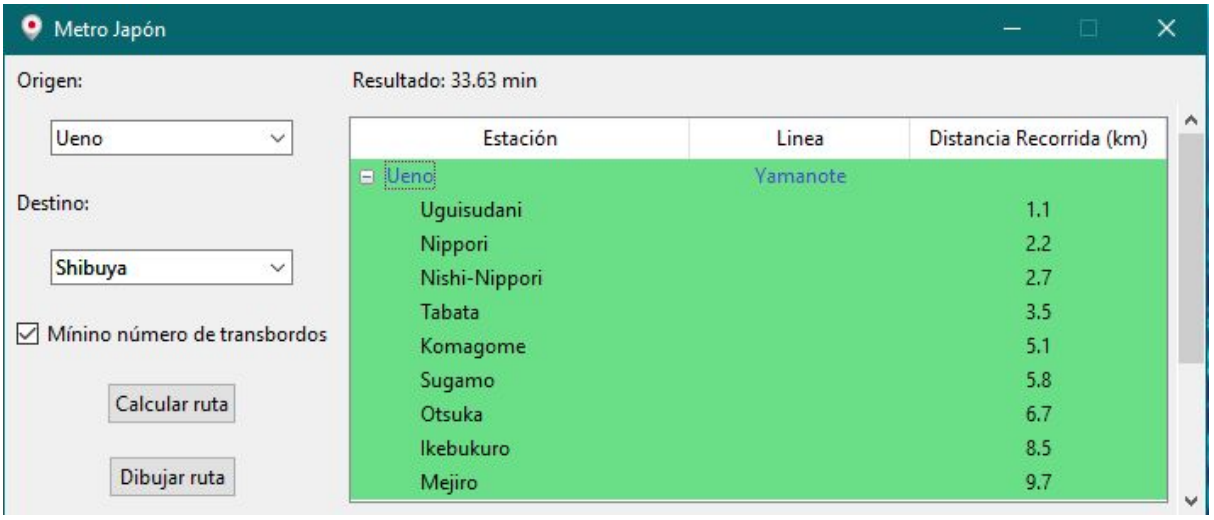
Pulsamos en el botón para dibujar una ruta y nos aparece la ventana del grafo.
La línea azul es la que indica la ruta. Las líneas gruesas indican los transbordos entre estaciones.



Ahora, seleccionamos la opción de utilizar los mínimos transbordos posibles.



Y volvemos a calcular la ruta. Como se puede observar, ya no hay que tomar la línea Sobu (amarilla) para llegar a nuestro destino.



Sacamos el grafo por pantalla y efectivamente nos damos cuenta de que la ruta es diferente.