Universidad Politécnica de Madrid

Proyecto de Inteligencia Artificial:

“Tokyo RP Metro App”

Grupo 3:
Montserrat Silva Cordero
Paloma Luna Cito
Pauline Isabela Conti
Stephanie Mennle
Jorge Arroyo Martínez
Alejandro Nieto Jeux

Madrid, España a 15 de diciembre de 2019

Contenido

Introducción

El objeto de este proyecto es diseñar una aplicación para hallar el trayecto óptimo entre dos estaciones del plano "Japan Railpass", el cual contiene algunas líneas del metro en Japón, teniendo en cuenta el número de transbordos, así como las distancias entre estaciones. En este documento serán presentados los parámetros y líneas seguidas para el desarrollo del proyecto, así como el algoritmo utilizado para cumplir el objetivo.

Algoritmo A*

El algoritmo A* es un algoritmo heurístico de búsqueda empleado para el cálculo de caminos mínimos en una red de nodos. Utiliza una función de evaluación heurística F(n) que etiqueta los nodos de la red y determina la probabilidad de cada nodo estudiado de pertenecer al camino óptimo.
La función heurística que utiliza el algoritmo está compuesta por otras dos funciones:

- **g(*n*)** indica la distancia del camino desde el nodo origen *s* al *n* (*s* es el nodo de origen y *n* el nodo que estamos estudiando).
- **h(*n*)** expresa la distancia estimada desde el nodo n hasta el nodo destino t (n es el nodo que estamos estudiando).

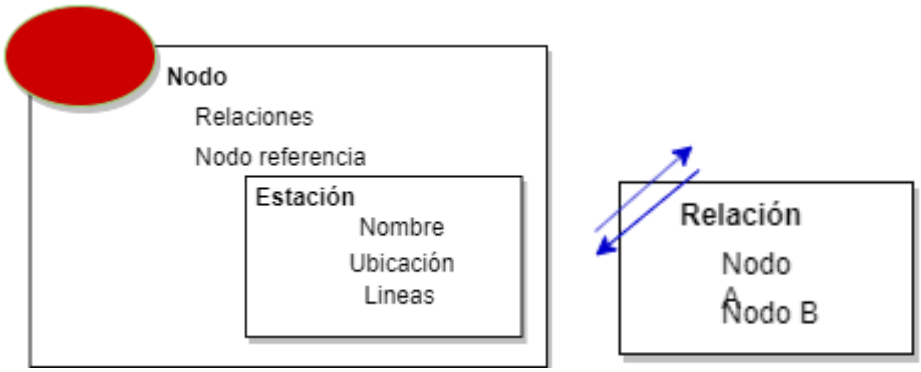
El algoritmo sigue los siguientes pasos:

1. Establecer el nodo *s* como origen.
2. Calcular el valor de F(*s*) y mover el nodo *s* al conjunto de lista abierta y dado que es el único nodo de la lista abierta pasaría a la lista cerrada.
3. Se calcula la F(*n*) de los nodos adyacentes al último nodo introducido en la lista cerrada calculando su g(*n*) y h(*n*) y los metemos en la lista abierta.
4. Comprobamos todos los valores de evaluación de los nodos de la lista abierta y seleccionamos el de menor valor, lo metemos en la lista cerrada y lo quitamos de la lista abierta.
5. Comprobamos si el nodo introducido en la lista cerrada es la meta. Si es la meta finaliza, si no lo es se repite desde el paso 3.
6. El algoritmo acaba cuando el nodo meta está en la lista cerrada.

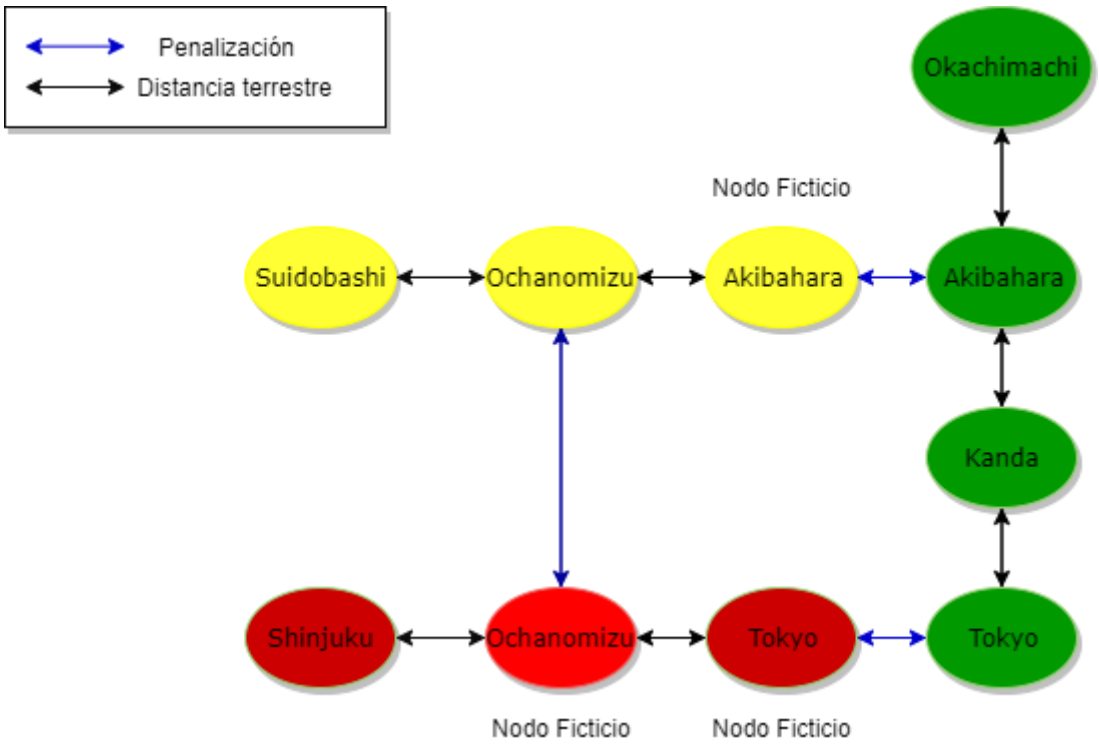
Implementación

El proyecto consta de una aplicación de escritorio desarrollada en el lenguaje de programación JAVA, por lo cual el desarrollo es orientado a objetos y su representación se enfoca en las siguientes clases: metro, estación, grafo, nodo y relación, diseñadas en función de la implementación del algoritmo presentado anteriormente.

Para la implementación y representación del metro se creó un grafo, donde cada nodo representa una estación del metro, donde cada nodo tiene de 1 a N vecinos, los cuales son las estaciones siguientes en el trayecto ya sea en la línea inicial o en las que exista intercambio.



En el caso de aquellas estaciones que tienen intercambio entre líneas fueron representadas como nodos ficticios dentro de nuestra red, a continuación, se presenta una explicación grafica de dichos nodos y sus respectivas relaciones.



Como se puede observar el nodo ficticio representa a la misma estación, sin embargo, éste esta en otra línea (distinguidas por colores), con respecto al peso existente entre las relaciones, pueden ser la distancia terrestre o la penalización, esta ultima es otorgada a los nodos que representan un cambio de línea.

De igual manera es importante mencionar que las distancias terrestres entre dos nodos son consideradas iguales para cualquier sentido.

Durante la ejecución del algoritmo es importante que los nodos evaluados siempre contengan referencia al nodo antecesor con el que esta siendo evaluado, dado que al ser encontrado el nodo objetivo se recorrerá el trayecto mediante estas referencias.

Recolección y obtención de datos

Como se mencionó con anterioridad el algoritmo A* hace uso de una función heurística compuesta por dos distancias g(n) y h(n), las cuales para fines de este proyecto fueron representadas y obtenidas de la siguiente manera:

- **g(*n*)** como distancia terrestre o distancia férrea
    - Se hizo una búsqueda en internet para la obtención de estas distancias, posterior a la recolección en distintas fuentes, se hizo una comprobación eligiendo de manera aleatoria las distancias y haciendo uso de la herramienta proporcionada por *Google Earth*, que permite calcular la longitud de un trazo; con ello se seleccionaron las distancias representativas de g(n).
- **h(*n*)** como distancia área
    - En el caso de estas distancias fue necesario buscar por medio de *Google Maps* la ubicación de cada una de las estaciones evaluadas, esta información debía contener latitud y longitud.
    - Durante la ejecución del algoritmo esta distancia es calculada de acuerdo con la ubicación de la estación A y estación B, estos cálculos fueron realizados dado la fórmula de distancia área.

Hemos asumido, que la velocidad media de los metros en Japón es de 35 km/h, adicionalmente hemos considerado que cada trasbordo duraría 5 minutos representados como penalización al tiempo total del trayecto, durante la implementación del algoritmo en el grafo construido se trabaja con los pesos como representación de distancias, es por ello por lo que la penalización es calculada en base a la relación ente la velocidad y el tiempo estipulado.

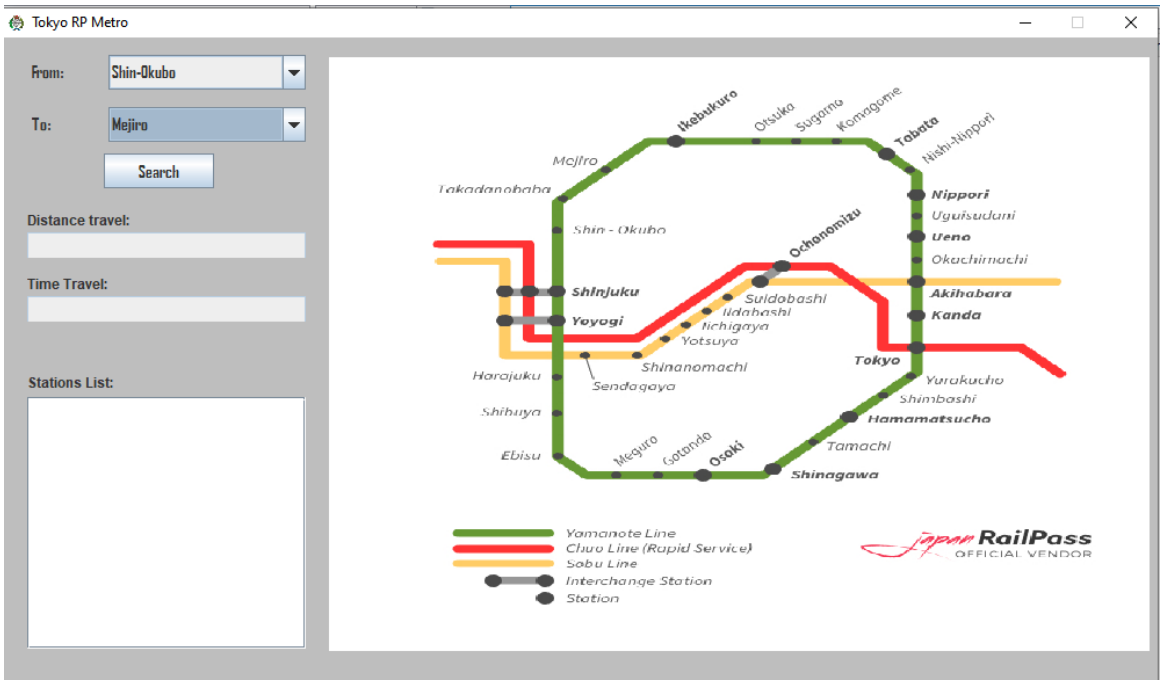
Tokyo RP Metro App

La aplicación desarrollada lleva por nombre “Tokyo RP Metro”, la cual cuenta con una interfaz gráfica, ésta está compuesta por:

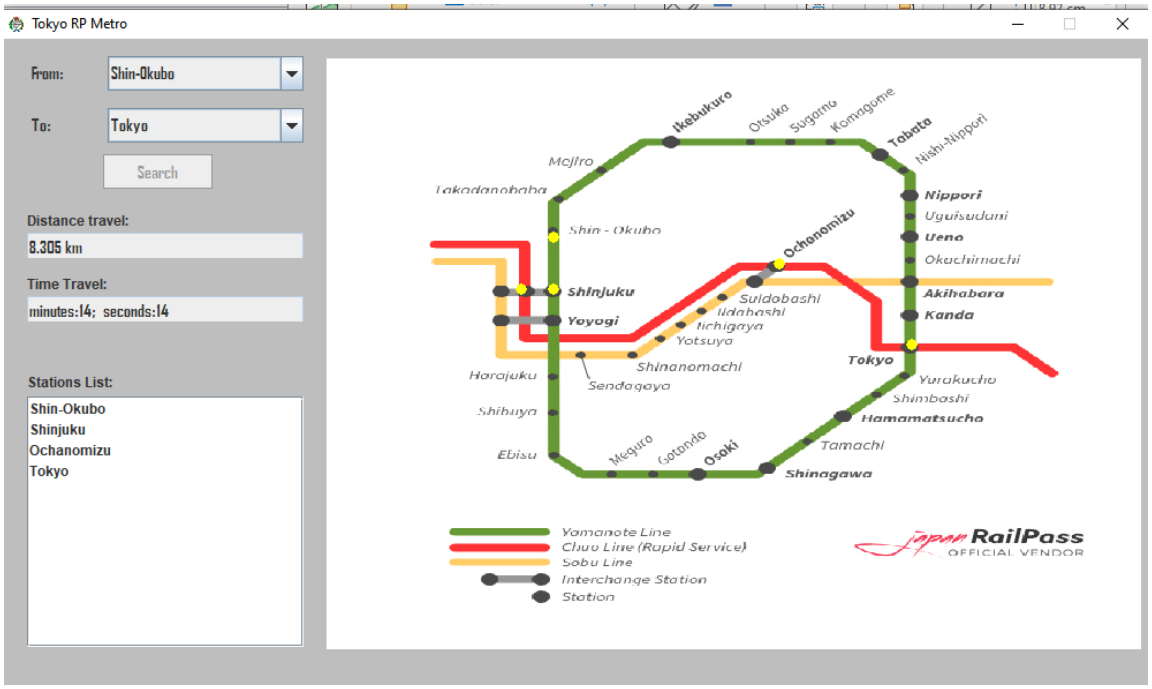
- El mapa del metro de Tokyo,
- Parámetros de entrada
    - Estación de origen
    - Estación de destino
- Parámetros de salida (calculados por el algoritmo)
    - Distancia aproximada del viaje
    - Duración aproximada del viaje
    - Listado de estaciones recorridas desde el origen hasta e destino
    - Indicación de estaciones dentro del mapa

Esta aplicación permite realizar la consulta de cualquier trayecto relacionado con las estaciones existentes.

Presentamos a continuación capturas de la aplicación y su funcionamiento:



La interfaz de la aplicación es intuitiva y sencilla, se presentan todas las estaciones en cajas de selección. Una vez elegidas las estaciones se solicita la búsqueda o calculo por medio del botón presentado. Automáticamente la aplicación presenta la distancia y tiempos aproximados, de igual forma el listado de estaciones a recorrer. En el plano estas mismas estaciones son resaltadas con puntos amarillos, tal y como se muestra en la siguiente ilustración.



Referencias

Inc., G. (s.f.). *Google Earth*. Obtenido de https://www.google.com/intl/es/earth/

Inc., G. (s.f.). *Google Maps*. Obtenido de https://www.google.es/maps/?hl=es

Japan Railpass. (s.f.). Obtenido de https://www.jrailpass.com/blog/es/linea-yamanote

Veness, C. (s.f.). *Movable Type Scripts*. Obtenido de Calculate distance, bearing and more between Latitude/Longitude points: https://www.movable-type.co.uk/scripts/latlong.html

Wikipedia. (s.f.). Obtenido de Línea Chūō (Rápida): https://es.wikipedia.org/wiki/L%C3%ADnea_Ch%C5%AB%C5%8D_(R%C3%A1pida)

Wikipedia. (s.f.). *Wikipedia*. Obtenido de Línea Chūō-Sōbu: https://es.wikipedia.org/wiki/L%C3%ADnea_Ch%C5%AB%C5%8D-S%C5%8Dbu

Wikipedia. (s.f.). *Wikipedia*. Obtenido de Línea Yamanote: https://es.wikipedia.org/wiki/L%C3%ADnea_Yamanote

Wikipedia. (s.f.). *Wikipedia*. Obtenido de Algoritmo de búsqueda A*: https://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_A*