Universidad Politécnica de Madrid

Escuela Técnica Superior de Ingenieros Informáticos

Proyecto de diseño de una aplicación que halle el trayecto óptimo entre dos estaciones de metro.

Caso particular: estaciones de metro de Lyon

CURSO 2023-2024

GRUPO 8:

- Marcos Ayuso Camarena (200369)
- Jaime Perán De Montes-Jovellar (200274)
- Alberto González López (140332)
- Lucas Panfil Balmaseda (180342)
- Francisco Chicote Martín (200256)

Índice

Descripción general del proyecto.

El objetivo del proyecto ha consistido en el diseño de la aplicación que permita hallar el trayecto óptimo entre dos estaciones del metro de Lyon teniendo en cuenta que, para llegar de una estación A a una estación B, pueden concurrir diferentes circunstancias que modifiquen ese trayecto óptimo como pueden ser, la existencia de un trasbordo entre estaciones, el horario en el que se esté realizando el trayecto, y la heurística calculada para esos dos puntos del trayecto.

Estructura de información. Datos.

Para la realización del trabajo se han tenido en cuenta diversos parámetros que han constituido la base de hechos de información de la que se ha servido la implementación para el cálculo y la evolución de los grafos.

Se ha utilizado un formato de tabla en la que hemos recogido la información estructurándola en pestañas (conjuntos de datos homogéneos), filas y columnas.

Primer conjunto de datos. Ubicación.

Se han recogido las ubicaciones en coordenadas de grados decimales (DD), ej. Gare de Vaise (45,7835154061337; 4,80483329334771).

Coordenadas	Gare de Vaise	Valmy	Gorge d Loup	Vieux lyon	Bellecour	Guillotiere	Saxe Gambetta	Garibaldi	Sans-Souci	Monplaisir-Lumièr	Grange Blanche	Laënnec	Mermoz - Pinel
X	45,78	45,78	45,77	45,76	45,76	45,76	45,75	45,75	45,75	45,75	45,74	45,74	45,73
Y	4,80	4,81	4,80	4,83	4,83	4,84	4,85	4,85	4,86	4,87	4,88	4,89	4,89

Segundo conjunto de datos. Trasbordos.

Se han identificado y, otorgado un valor numérico que identifica la penalización de uso, los trasbordos posibles en función de los requisitos marcados en el mapa del metro de Lyon.

		Gare de Vaise	Valmy	Gorge d Loup	Vieux lyon	Bellecour	Guillotiere	Saxe Gambetta	Garibaldi	Sans-Souci	Monplaisir-Lumi	Grange Blanche	Laënnec	Mermoz - Pinel	Parilly	Gare De Venissieux	Vaulx-en-Vela	Laurent Bonnev	Cusset	Flachet	Grattе-Ciel	Hôtel de Ville	Charpennes	Massena	Foch
5	Vieux lyon	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	Bellecour	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	Guillotiere	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	Saxe Gambetta	0	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	Garibaldi	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tercer conjunto de datos. Horas.

Se ha creado un mapa horario en el que se reflejan los retrasos producidos por tiempo en hora punta con una penalización determinada en función de la estación de metro, de tal forma que en una estación concurrida puede tener una confluencia mayor en hora punta y por tanto mayor penalización que otra estación menos transitada.

HORA	Gare de Vaise	Valmy	Gorge d Loup	Vieux lyon	Bellecour	Guillotiere	Saxe Gambetta	Garibaldi	Sans-Souci	lonplaisir-Lumiè	Grange Blanche	Laënnec	Mermoz - Pinel	Parilly	Gare De Venissieux	Vaulx-en-Velin	aurent Bonneva
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	2	3	4	6	9	7	6	6	6	5	4	3	3	2	2	1	1
2	4	6	7	11	17	14	13	11	11	10	9	6	6	4	4	3	3
3	6	9	11	17	26	21	19	17	17	15	13	9	9	6	6	4	4
4	9	11	14	23	34	29	26	23	23	20	17	11	11	9	9	6	6
5	11	14	18	29	43	36	32	29	29	25	21	14	14	11	11	7	7

Cuarto conjunto de datos. Distancias.

En este caso se ha registrado el valor de todas las distancias entre las estaciones contiguas y aquellas a las que se puede acceder mediante trasbordo directo. En el caso de no ser contiguas o no tener trasbordo los valores se han informado a cero.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1		Gare de Vaise	Valmy	Gorge d Loup	Vieux lyon	Bellecour	Guillotiere	Saxe Gambetta	Garibaldi	Sans-Souci	Monplaisir-Lumière	Grange Blanche	Laënnec	Mermoz - Pinel	Parilly	Gare De Venissieux	Vaulx-en-Velin	aurent Bonneva
2	Gare de Vaise	0	60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Valmy	60	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	Gorge d Loup	0	120	0	180	0	0	0	0	0	0	0	0	0	0	0	0	0
5	Vieux lyon	0	0	180	0	120	0	0	0	0	0	0	0	0	0	0	0	0
6	Bellecour	0	0	0	120	0	60	0	0	0	0	0	0	0	0	0	0	0
7	Guillotiere	0	0	0	0	60	0	60	0	0	0	0	0	0	0	0	0	0
8	Saxe Gambetta	0	0	0	0	0	60	0	60	0	0	0	0	0	0	0	0	0
9	Garibaldi	0	0	0	0	0	0	60	0	120	0	0	0	0	0	0	0	0

Quinto conjunto. Líneas.

Se establecen el conjunto de paradas que tiene cada línea y se identifican con su color del mapa.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		Gare de Vaise	Valmy	Gorge d Loup	Vieux lyon	Bellecour	Guillotiere	Saxe Gambetta	Garibaldi	Sans-Souci	onplaisir-Lumi	Grange Blanche	Laënnec	Mermoz - Pinel	Parilly	Gare De Venissieux	Vaulx-en-Velin	aurent Bonnev	Cusset
2	Gare de Vais	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
3	Valmy	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
4	Gorge d Lou	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
5	Vieux lyon	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
6	Bellecour	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
7	Guillotiere	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
8	axe Gambet	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
9	Garibaldi	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
10	Sans-Souci	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
11	nplaisir-Lum	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
12	range Blanc	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
13	Laënnec	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
14	Mermoz - Pin	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
15	Parilly	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
16	e De Venis	green	green	green	green	green	green	green	green	green	green	green	green	green	green	green	red	red	red
17	Vaulx-en-Vel	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red
18	aurent Bonne	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red
19	Cusset	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red
20	Flachet	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red
21	Grattte-Ciel	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red
22	Jean-Ville	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red	red

Entorno y Desarrollo

Durante la realización de la práctica se han valorado diversas opciones tanto de lenguajes como de entornos en los que realizar el proyecto, pero al final hemos optado, pero llevarlo a cabo en Python, y ayudarnos de HTML y CSS para la interfaz que nos han permitido hacer estas librerías:

- **Pandas**: Para la obtención y tratamiento de los datos.
- **Networkx**: Para la creación de los grafos sobre los que nos basamos.
- **Math**: Para los cálculos geográficos.
- **Folium**: Para mostrar el mapa y hacerlo más atractivo
- **Ipywidgets**: Para la interfaz

A la hora de ejecutar el código hemos optado por hacerlo en Jupyter, ya que así podíamos sacar una interfaz gráfica atractiva y una ejecución sencilla en local sin reparar en la creación de mapas etc.

El código está dividido en tres partes principalmente.

La primera en la cual obtenemos los datos de las hojas de Excel y utilizamos los datos para hacer un grafo ponderado sobre el cual aplicar el algoritmo A*.

La segunda en la que Calculamos el camino más corto mediante el algoritmo A* ayudándonos de distintas funciones auxiliares para calcular retraso por la hora, heurística, color de la línea, retraso por el transbordo y por último para generar el camino una vez ejecutado el algoritmo.

La tercera es en la que mostramos la interfaz, creamos los mapas para poder mostrar en distintas capas tanto las líneas del metro de Lyon como el camino más corto, la interfaz para introducir los datos de la ruta deseada y el tratamiento de errores por si no se introduce una parada valida o a una hora en la que el metro esté cerrado.

Algoritmo utilizado, A*

Se escoge el primer nodo origen. A continuación, "*El mejor el primero",* se escoge el nodo más prometedor, de acuerdo con la función heurística asignada para cada una de las estaciones.

Se utiliza una función de reordenamiento de los nodos de la lista abierta

f*(n) será el coste real de un camino óptimo desde el nodo origen a un nodo meta, restringido a que dicho nodo pase por el nodo n.

g será el coste del camino que va desde el nodo inicial origen al nodo meta en el árbol de búsqueda, este camino es el de coste más bajo encontrado hasta el momento.

h será nuestra función heurística.

Buscamos con esta búsqueda A*:

COMPLETITUD.

Un algoritmo es completo si termina con una solución si ésta existe.

ADMISIBILIDAD.

Un algoritmo es admisible si asegura encontrar una solución óptima, si esta existe. A* es admisible.

EFICIENCIA.

Un algoritmo A1 es más eficiente que otro A2 si A1 está más informado que A2, así cada nodo desarrollado por A1 también es desarrollado por A2. A2 desarrolla, al menos tantos nodos como A1

OPTIMIDAD.

Un procedimiento es óptimo, dentro de un conjunto de procedimientos, si es más eficiente que todos los elementos de dicho conjunto.

Cálculos efectuados -> f*(n) = g*(n) + h*(n)

Ésta establece su valor heurístico h(n) de acuerdo con las coordenadas de la estación (DD:X, Y) y la velocidad media del metro de Lyon y cualquier otro punto con conexión.

g(n) se calcula en función del recorrido, el destino seleccionado, las distancias por estaciones e incrementado su valor con h(n) para obtener su f(n).

Se utilizan dos estructuras de ordenamiento de los nodos de acuerdo con el algoritmo A*, una lista abierta, para nosotros nodosAbiertos [] y lista cerrada, nodosCerrados.

Cálculo del camino más corto

En base a las estructuras anteriormente mencionadas, se utiliza una estructura recursiva que irá construyendo el camino a recorrer tratando la pila de nodos abiertos y cerrados. El procedimiento en la implementación sería el que sigue:

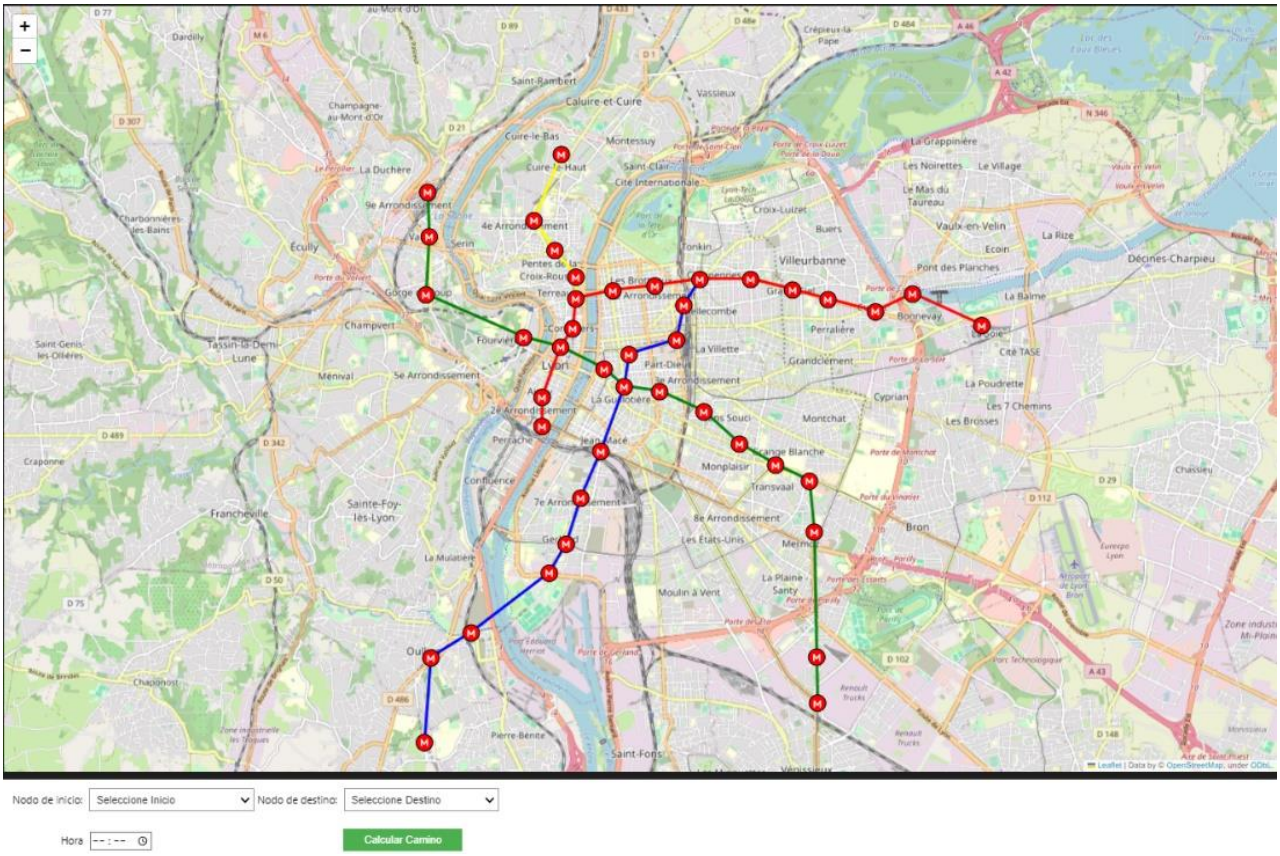
1. recibe un nodo como entrada y lo trata según este en el grafo (la primera vez recibe el nodo inicial)
2. si no está en nodos abiertos no cerrados se añade a la lista de nodos abiertos
3. Se calcula el coste de la arista, entendida ésta como el recorrido al siguiente nodo, y se guarda el valor como CosteActual
4. A continuación, se calcula el coste al siguiente nodo. Si el coste es cero o mayor que el coste calculado anteriormente nos quedamos con los valores almacenados en CosteActual.
5. Mientras haya elementos en nuestra estructura de nodosAbiertos seguimos iterando. El siguiente nodo de nodosAbiertos lo metemos en la lista de nodosCerrados e iteramos.

La lista de nodos abiertos está vacía, se ha creado el camino en función de los elementos de la lista de nodos cerrada, y se ha construido el path en pathAux para poder mostrar el recorrido en la interfaz del usuario.

Casos de uso

En este apartado se presentarán dos casos de uso de funcionamiento.

El planteamiento general será mostrar el camino recorrido de acuerdo con las opciones seleccionadas en el interfaz de usuario. El trayecto deberá variar como consecuencia de los pesos asignados en función de los horarios y trasbordos elegidos por el algoritmo A*



Primer caso de uso

Las opciones del trayecto para este primer caso de uso serán las siguientes:

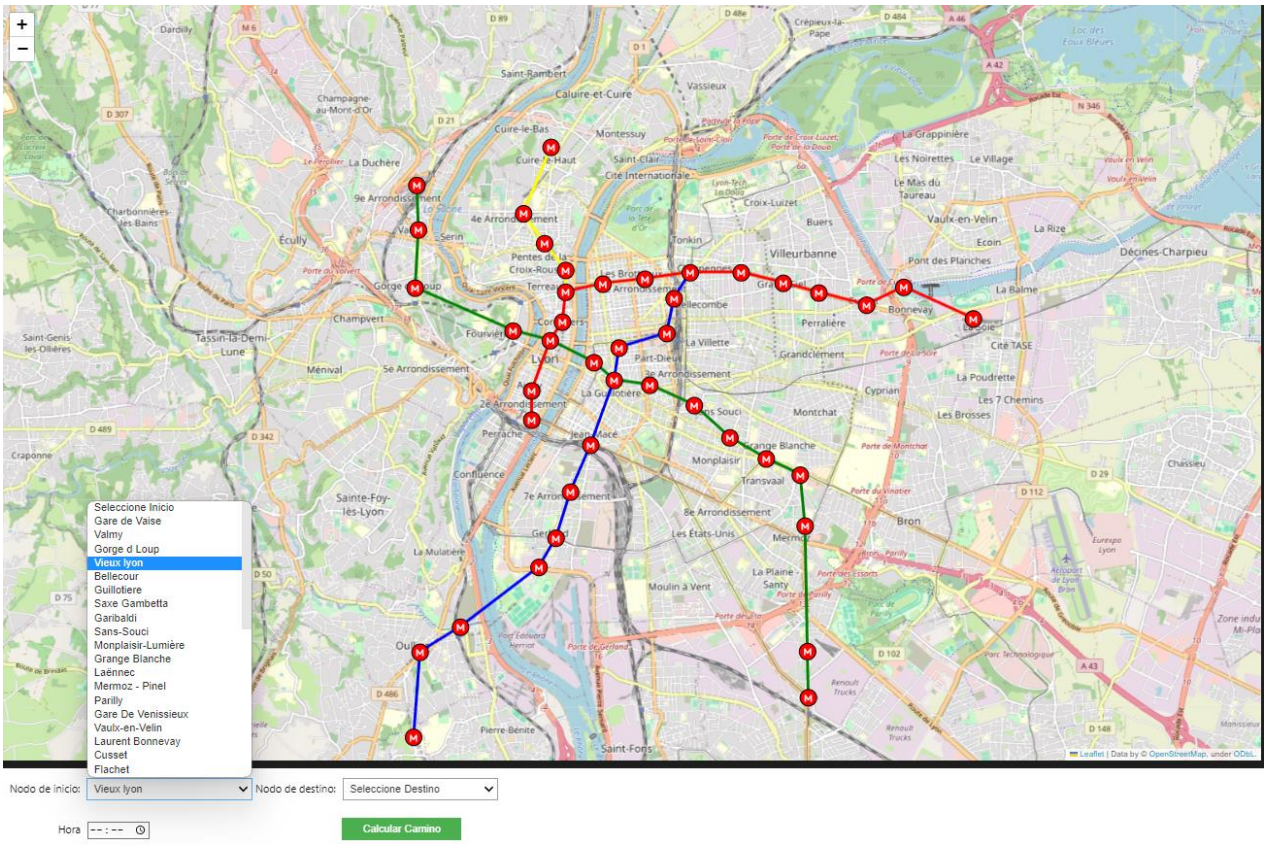
Estación origen: Vieux Lyon

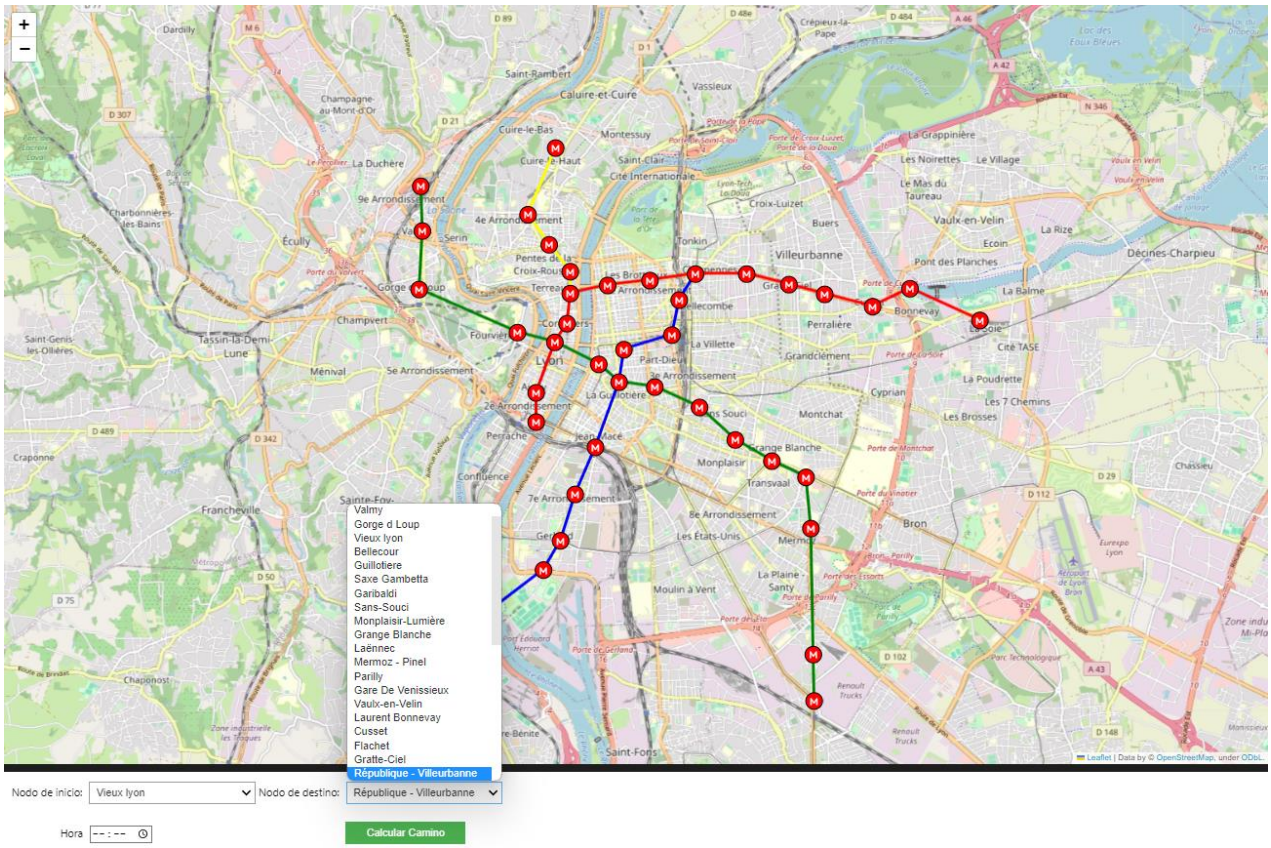
Estación destino: République Villeurbanne

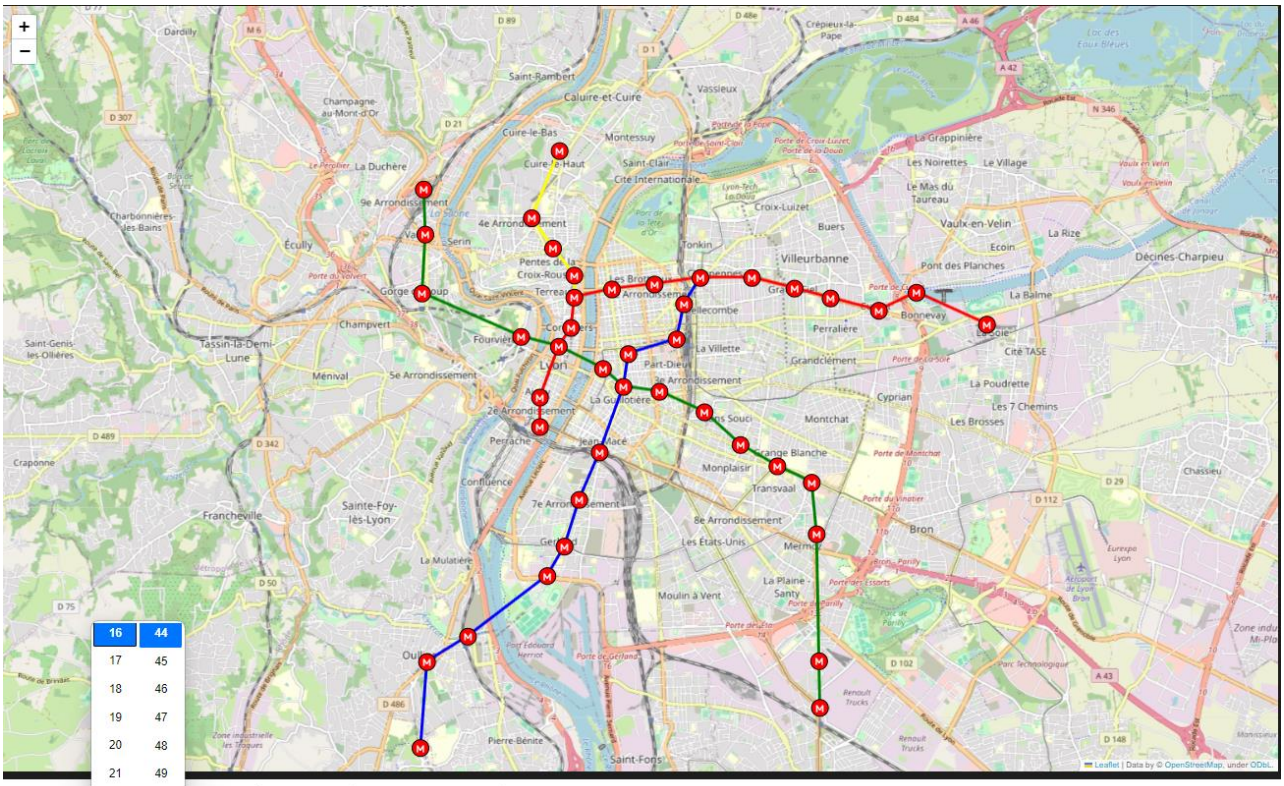
Horario de viaje: 16:44

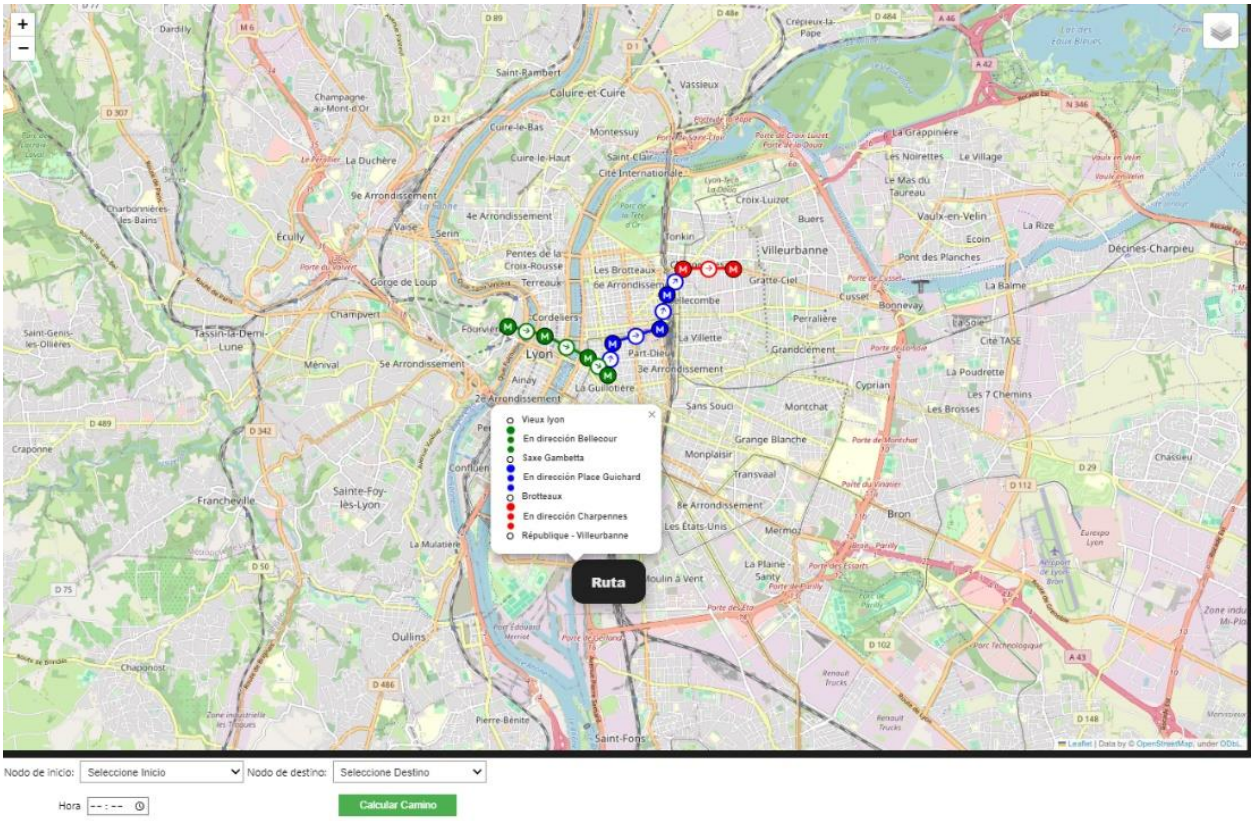
El trayecto que selecciona el algoritmo en base a estos parámetros pasa por escoger el trasbordo que tiene la línea verde con la línea azul, y posteriormente escogerá el trasbordo que tiene la línea azul y la línea roja. Aparentemente el usuario podría pensar que este trayecto le llevaría más tiempo, pero los resultados del algoritmo le llevan por este camino.

A continuación, se presentan las selecciones del usuario:


Seleccione Inicio
Gare de Vaise
Valmy
Gorge d Loup
Vieux lyon
Bellecour
Guillotiere
Saxe Gambetta
Garibaldi
Sans-Souci
Monplaisir-Lumière
Grange Blanche
Laënnec
Mermoz - Pinel
Parilly
Gare De Venissieux
Vaulx-en-Velin
Laurent Bonnevay
Cusset
Flachet
Nodo de inicio:
Vieux lyon
Nodo de destino:
Seleccione Destino
Hora
Calcular Camino


Valmy
Gorge d Loup
Vieux lyon
Bellecour
Guillotiere
Saxe Gambetta
Garibaldi
Sans-Souci
Monplaisir-Lumière
Grange Blanche
Laënnec
Mermoz - Pinel
Parilly
Gare De Venissieux
Vaulx-en-Velin
Laurent Bonnevay
Cusset
Flachet
Gratte-Ciel
République - Villeurbanne
Nodo de inicio:
Vieux lyon
Nodo de destino:
République - Villeurbanne
Hora
Calcular Camino


16
44
17
45
18
46
19
47
20
48
21
49
22
50
Nodo de inicio:
Nodo de destino:
République - Villeurbanne
Hora
16 : 44
Calcular Camino


Vieux lyon
En dirección Bellecour
Saxe Gambetta
En dirección Place Guichard
Brotteaux
En dirección Charpennes
République - Villeurbanne
Ruta
Nodo de inicio:
Seleccione Inicio
Nodo de destino:
Seleccione Destino
Hora
--:--
Calcular Camino

Segundo caso de uso

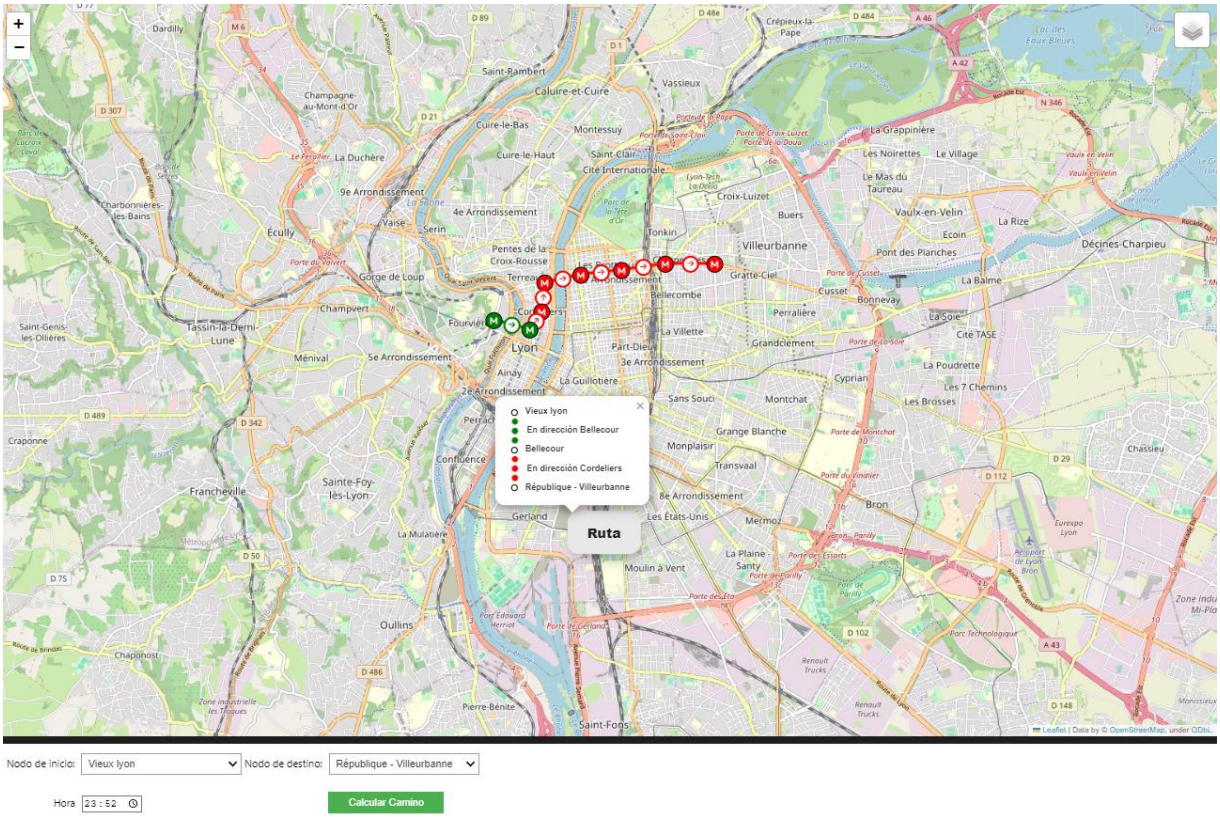
En este caso el trayecto que va a seleccionar el usuario va a ser el mismo que en el caso anterior, pero el horario de selección va a variar, de tal forma que al no ser un trayecto en hora punta los pesos del algoritmo le mostrarán una opción diferente.

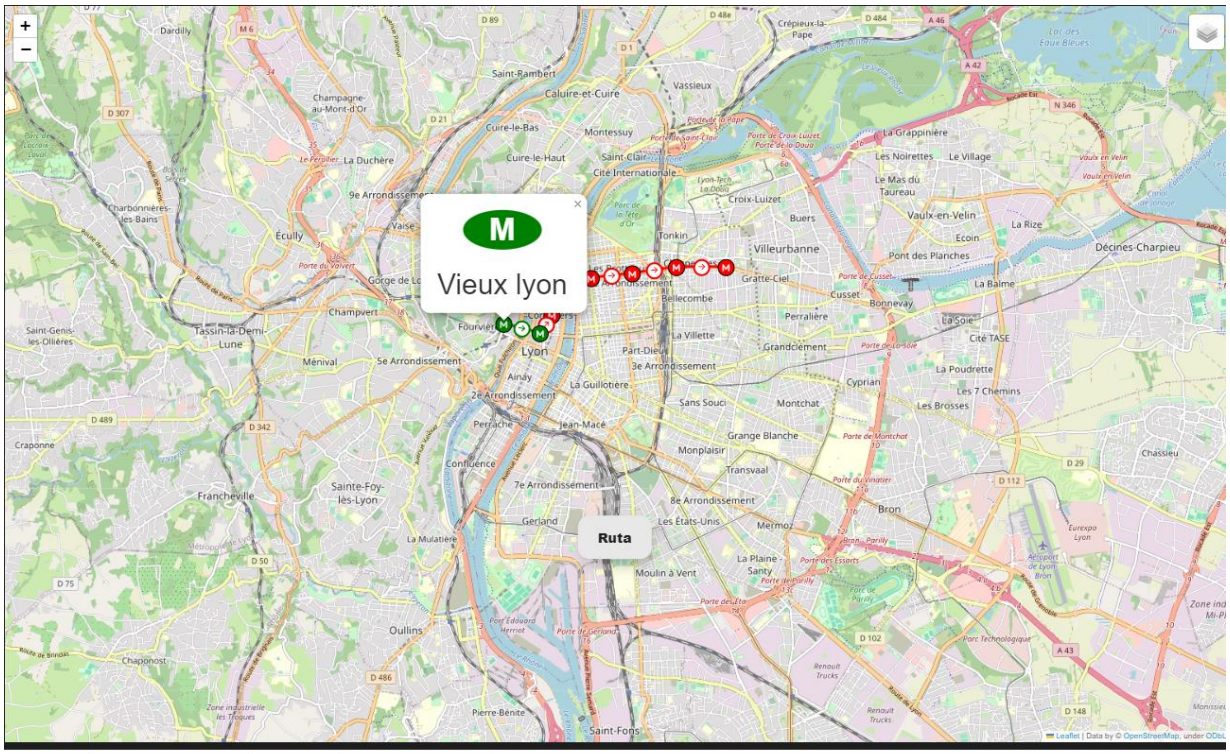
Estación origen: Vieux Lyon

Estación destino: República Villeurbanne

Horario de viaje: 23:52

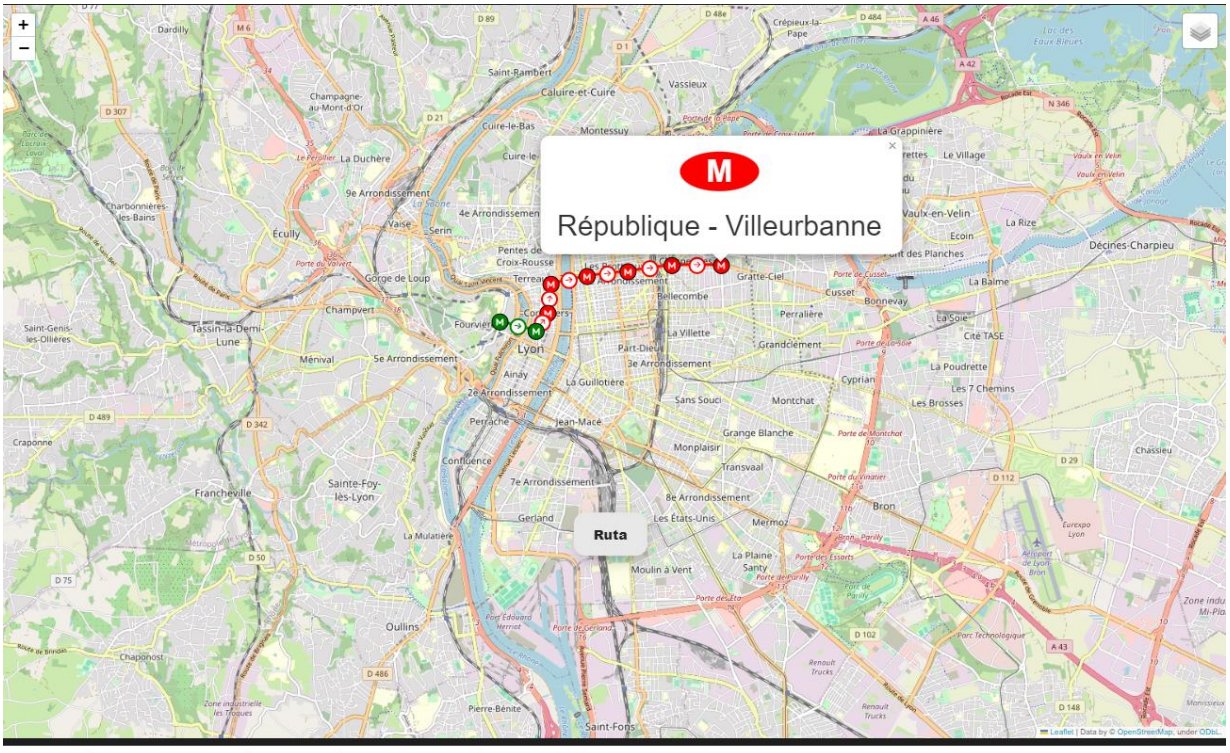
En este caso la ruta marcada le llevará al usuario utilizando únicamente el transbordo de la línea roja yendo directamente ya a su parada de destino.





Nodo de inicio: Vieux lyon Nodo de destino: République - Villeurbanne

Hora 23 : 52 Calcular Camino



Nodo de inicio: Vieux lyon Nodo de destino: République - Villeurbanne

Hora 23 : 52 Calcular Camino