

ANÁLISIS DEL RETO 4

Catalina Isabel Muñoz, 202121606, ci.munoz@uniandes.edu.co

Yaisa Catalina Ramírez, 202021914, y.ramirez@uniandes.edu.co

Ana María Patrón, 201714291, am.patron@uniandes.edu.co

Carga de Datos

Para la carga de datos se usaron tres estructuras de datos: dos tablas de hash y un grafo dirigido. En primer lugar, se recorrió el archivo “bus_stops” y cada una de las paradas se almacenó en la tabla “vértices” junto con su información de latitud, longitud y vecindario. Adicionalmente, se consideró que si la parada era de tipo “S”, esta también se debía agregar a la tabla “transbordos” en la cual si el transbordo ya se había incluido, se añadía el bus a la lista de buses asociados a la estación de transbordo. Posteriormente, se recorre el archivo “bus_edges” a partir del cual se toman los vértices de inicio y fin, se añaden al grafo, se calcula su distancia (peso) y finalmente se añade el arco entre dichos vértices. Una vez creados todos los arcos del archivo, se toma la tabla “transbordos”, se crean vértices para cada uno de las llaves de la tabla (estaciones de transbordo) y se generan arcos (en ambas direcciones) entre la llave y todos los elementos de su lista.

Resultados carga de datos:

```
El rango del area rectangular de la forma [Longitud minima, Longitud maxima] y [Latitud minima, Latitud maxima] es:
Longitudes: [ 2.055835 2.221753]
Latitudes: [ 41.32164 41.46759]
Total de estaciones exclusivas 1244
Total de transbordos 1080
Numero de arcos utilizados: 15834
Nota: Key representa el Code-Ruta
Las 5 primeras estaciones son:
```

Identificador estacion	Geolocalizacion	Numero de estaciones conectadas
8-106	['2.159124', '41.45014']	1
9-106	['2.160857', '41.42953']	2
10-106	['2.154352', '41.43958']	2
11-106	['2.17833', '41.46362']	2
12-106	['2.175149', '41.46383']	2

```
Las 5 últimas estaciones son:
```

Identificador estacion	Geolocalizacion	Numero de estaciones conectadas
2331-V9	['2.12288', '41.40709']	1
2330-V9	['2.150898', '41.38442']	2
2329-V9	['2.152377', '41.3784']	2
2328-V9	['2.130285', '41.39952']	2
2327-V9	['2.154602', '41.37665']	2

Requerimiento 1

Descripción

Se hace uso del algoritmo DFS para buscar todas las posibles rutas que existen entre la estación de origen y de destino. Asimismo, una vez obtenido el camino usando "PathTo", se obtuvieron las distancias a cada estación siguiente usando la función "haversine".

Entrada	La estación de origen y de destino registradas de la forma Code-IdBus
Salidas	Si existe camino, informar la distancia total del recorrido, el total de estaciones y de transbordos y las estaciones del recorrido indicando su identificador y distancia a la siguiente estación.
Implementado (Sí/No)	Si. Yaisa C. Ramírez

Análisis de complejidad

Se define V como el número de vértices y E como el número de arcos

Pasos	Complejidad
Paso 1. Hacer el recorrido DFS sobre el grafo usando como punto de partida el origen	$O(E + V)$
Paso 2. Obtener si existe camino o no hacia el vértice destino	$O(1)$
Paso 3. Recorrer la pila de estaciones que corresponden al camino	$O(N)$ donde N es el número de estaciones del camino, $N \ll V$
Paso 4. Obtener la longitud y latitud de cada estación	$O(N)$
Paso 5. Obtener la distancia entre las estaciones usando haversine	$O(1)$
TOTAL	$O(E+V)$

Pruebas Realizadas

Se realizó la prueba con un computador Dell Inspiron core i7 con 12GB de RAM y 1.3GHz de procesamiento. Se buscó camino de 35-109 a 58-V7

Entrada	Tiempo (s)
large	0.356

Tablas de datos

```
Estacion inicial: 35-109
Estacion final: 58-V7
Distancia del camino planteado: 80.37531970292524
El camino planteado tiene 67 estaciones
El camino planteado tiene 2 transbordos
Las estaciones del camino planteado son:
67
```

Identificador estacion	Distancia a la siguiente estacion
35-109	5.897942417406026
36-109	5.715554521923522
37-109	1.3908692094375423
38-109	7.637311670422826
39-109	2.73982760950849
40-109	2.475523097609836
41-109	1.0878674816637002
42-109	0.6373548219186733
43-109	5.545455847538612
44-109	1.9041325239623466
45-109	1.9145436746923608
46-109	0.3645024060923372
47-109	0.4298792915079227
48-109	0.4208265184647341
49-109	0.4520124537171439
50-109	0.3817814862042315
51-109	0.14630202237933776
52-109	3.0468263591508222
53-109	0.5214794433317042
54-109	0.2744815818672842
55-109	0.4537715385823353
56-109	0.429707912943077
57-109	1.9469757794049458
58-109	5.311033804278349

58-109	5.311033804278349
853-109	0.693880300292971
854-109	0.5026647800026656
855-109	1.2304761301188711
856-109	0.49767831047709293
857-109	0.3752381188342453
858-109	1.2495138681118572
859-109	0.6030517192846034
860-109	0.7234013936776537
861-109	1.088429049137055
862-109	1.8856171599970397
863-109	2.8552033843417526
864-109	2.0216906922337805
865-109	1.8036035294070683
866-109	0.25716068108608625
909-109	0.8141030000763735
910-109	0.09477477636938511
911-109	0.27504818852775176
912-109	0.11768025381311034
913-109	2.857948722914829
46-CJ	0.010424916543695229
44-CJ	0
T-44	0
44-V7	0.010424916543695229
46-V7	0.4031151189350021
58-V7	0

Graficas

No se realizaron gráficas dado que se hizo una única prueba con large para obtener el grafo completo

Análisis

Este requerimiento tiene una complejidad temporal alta. Sin embargo, si se plantean muchas búsquedas para un mismo origen, solo se tiene una complejidad de $O(1)$ ya que solo se debe ejecutar una vez el algoritmo DFS sobre el grafo. En cuanto a la complejidad espacial, esta es de $O(V)$.

Requerimiento 2

Descripción

Implementamos el algoritmo BFS para encontrar el camino con menos paradas. Para encontrar las distancias entre cada estación que hacía parte del camino y la anterior extraemos las respectivas longitudes y latitudes, y con esas hallamos la distancia con la función haversine

Entrada	Identificadores de estación de origen y de llegada en la forma code-bus
Salidas	Estaciones que hacen parte del camino, distancia total de este, número de transbordos y distancias entre estaciones
Implementado (Sí/No)	Sí, Ana M.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1. Hacer el recorrido BFS sobre el grafo usando como punto de partida el origen	$O(E + V)$, $E = \#$ arcos, $V = \#$ vértices
Paso 2. Obtener si existe camino o no hacia el vértice destino	$O(1)$
Paso 3. Recorrer la pila de estaciones que corresponden al camino	$O(N)$, $N =$ tamaño de la pila
Paso 4. Obtener la longitud y latitud de cada estación	$O(N)$, $N =$ tamaño de la pila
Paso 5. Obtener la distancia entre las estaciones usando haversine	$O(1)$
TOTAL	$O(E+V)$

Pruebas Realizadas

Se realizó la prueba en computador con procesador 1.6 GHz Intel Core i5, memoria RAM de 8GB y sistema operativo MacOS High Sierra, usando la base large y con estación inicial 220-116 y final 890-87

Entrada	Tiempo (s)
large	0.284

Tablas de datos

estacion inicial: 220-116	
estacion final: 890-87	
Distancia del camino planteado: 5.915273122782805	
El camino planteado tiene 15 estaciones	
El camino planteado tiene 1 transbordos	
Las estaciones del camino planteado son:	
Identificador estacion	Distancia a la siguiente estacion
221-116	0.36488587909348086
222-116	0.14066173330454448
223-116	0.36492760190125545
224-116	0.6356213110432379
225-116	0.2970830148499219
226-116	0.7907751926465045
227-116	1.079870701694794
228-116	0.09132163827427284
883-116	0.48672340021696525
T-883	0
883-87	0.0
884-87	0.3329406113592043
885-87	0.4470533339629668
886-87	0.5138380777801579
890-87	0.36957062665549983

Graficas

Se omite dado que solo se evaluó con large

Análisis

Se omite dado que solo se evaluó con large

Requerimiento 3

Descripción

Se desea conocer si existen componentes conectados en el grafo y de ser así cuántos son y conocer sus características.

Entrada	Ninguno
Salidas	<ul style="list-style-type: none">El total de componentes conectados dentro del grafo.Mostrar los 5 componentes conectados más grandes (de mayor a menor número de estaciones en la componente fuertemente conectada):<ul style="list-style-type: none">El número de estaciones que pertenecen a dicho componente

	<ul style="list-style-type: none"> ○ Los identificadores de las tres primeras y tres últimas estaciones pertenecientes al componente.
Implementado (Sí/No)	Si, catalina Muñoz

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1 kosaraju	$O(E+V)$
Paso 2 ciclo sobre los identificadores	$O(V)$
Paso 3 merge	$O(V \log V)$
TOTAL	<i>$O(N \log N)$ donde N es el numero de estaciones que tiene el componente mas grande.</i>

Pruebas Realizadas

Descripción de las pruebas de tiempos de ejecución y memoria utilizada. Incluir descripción del procedimiento, las condiciones, las herramientas y recursos utilizados (librerías, computadores donde se ejecutan las pruebas, entre otros).

Entrada	Tiempo (s)
Large	7.296031951904297

Tablas de datos

```
El total de componentes conectados dentro del grafo es: 6

-----
En el componente 6 hay 5557 elementos.
Identificadores de las Primeras tres estaciones en el componente:
1174-65
1017-H10
280-V29
-----
Identificadores de las ultimas tres estaciones en el componente:
1496-67
803-47
1294-55
-----
En el componente 4 hay 98 elementos.
Identificadores de las Primeras tres estaciones en el componente:
561-128
565-128
327-118
-----
Identificadores de las ultimas tres estaciones en el componente:
559-128
321-118
314-118
-----
En el componente 5 hay 38 elementos.
Identificadores de las Primeras tres estaciones en el componente:
1921-BCTO
1922-BCTO
T-1905
-----
Identificadores de las ultimas tres estaciones en el componente:
1898-BCTE
1913-BCTO
1918-BCTO
-----
En el componente 1 hay 23 elementos.
Identificadores de las Primeras tres estaciones en el componente:
85-111
96-111
78-111
-----
Identificadores de las ultimas tres estaciones en el componente:
76-111
92-111
91-111
-----
En el componente 3 hay 6 elementos.
Identificadores de las Primeras tres estaciones en el componente:
9-106
13-106
11-106
-----
Identificadores de las ultimas tres estaciones en el componente:
12-106
10-106
8-106
-----
```

Requerimiento 4

Descripción

Para obtener las estaciones mas cercanas, se recorren todos los vértices para obtener las menores distancias a cada punto. A partir de ello, se recorre el grafo con dijkstra y se obtiene la ruta para llegar a la estación de destino más cercana al punto deseado.

Entrada	Localización geográfica del origen y del destino.
Salidas	Distancia entre las localizaciones y sus estaciones más cercanas, distancia total del recorrido, total de estaciones y estaciones de transbordo del recorrido y finalmente, una lista con la información de las estaciones usando identificador y distancia a la siguiente estación.
Implementado (Sí/No)	Sí. Yaisa C. Ramírez

Análisis de complejidad

En este caso se define V como el número de vértices y E como el número de arcos

Pasos	Complejidad
--------------	--------------------

Paso 1. Recorrer las Keys de la tabla de hash donde estaban almacenados los vértices	$O(V)$
Paso 2. Obtener las distancias entre los puntos dados y cada uno de las estaciones usando la función haversine	$O(1)$
Paso 3. Implementar Dijkstra sobre el grafo usando como origen el vértice origen hallado	$O(E \log V)$
Paso 4. Revisar si existe camino entre hacia el vértice destino	$O(1)$
Paso 5. Si hay camino, obtener la lista de vértices a recorrer junto con sus propiedades	$O(1)$
Paso 6. Si no hay camino, ejecutar de nuevo la función pero eliminando los vértices previamente hallados de la copia del grafo.	$O(V)$
TOTAL	$O(V)$

Pruebas Realizadas

Se realizó la prueba con un computador Dell Inspiron core i7 con 12GB de RAM y 1.3GHz de procesamiento. Se probó con la siguiente entrada:

[Long, Lat] (inicio): [2.14807, 41.3742]

[Long, Lat] (destino): [2.13802, 41.3742]

Entrada	Tiempo (s)
large	8.4561

Tablas de datos

```

Longitud del origen: 2.14807
Latitud del origen: 41.3742
Longitud del destino: 2.13802
Latitud del destino: 41.3742
Distancia entre localizacion de origen y estacion de bus mas cercana: 0.03497582300820631
Distancia total entre estaciones del recorrido: 2.1222165882385653
Distancia entre estacion destino y localizacion de origen: 0.13512296503911966
Total de estaciones del camino: 10
Total de transbordos del camino: 3
Las estaciones del camino planteado son:
+-----+-----+
| Identificador estacion | Distancia a la siguiente estacion |
+-----+-----+
| T-1494 | 0 |
| 1494-CJ | 0.843236117801968 |
| 1636-CJ | 0 |
| T-1636 | 0 |
| 1636-78 | 0.15164638860128732 |
| 1637-78 | 0 |
| T-1637 | 0 |
| 1637-115 | 0.20657216965817057 |
| 984-115 | 0.9207619121771392 |
| T-202 | 0 |
+-----+-----+

```

Graficas

No se realizaron gráficas dado que se evaluó únicamente con large para tener el grafo completo.

Análisis

A comparación de los primeros requerimientos, por ejemplo, con este se obtuvo la mayor complejidad temporal (a pesar de que el algoritmo Dijkstra tiene una complejidad considerablemente inferior a DFS y BFS). Esto se debe a que se recorrieron todos los vértices para obtener las menores distancias. Una posible solución a dicho problema podría ser plantear de forma temporal los puntos de origen y destino como vértices y recorrer desde cada uno el algoritmo Dijkstra para obtener la menor distancia a cada vértice.

Requerimiento 5

Descripción

Hallamos todos los caminos posibles que logran que el camino sea más rápido (pues el enunciado menciona “le puedo facilitar a los usuarios visitar múltiples lugares de la manera más rápida posible”) y a estos caminos le aplicamos la condición de que su número de conexiones sea menor o igual que el máximo permitido y una vez obtenemos los que cumple la condición, obtenemos su información

Entrada	Estación de origen (code-bus) y número máximo de conexiones
Salidas	Identificador de estación alcanzable, su geolocalización y la longitud
Implementado (Sí/No)	Sí, Ana M.

Análisis de complejidad

Análisis de complejidad de cada uno de los pasos del algoritmo

Pasos	Complejidad
Paso 1. Algoritmo Dijkstra para obtener caminos posibles más rápidos	$O(E \log V)$, $E = \#$ arcos, $V = \#$ vértices
Paso 2. Acotar con la condición del máximo número de estaciones	$O(N)$, $N = \#$ total de estaciones alcanzables posibles
Paso 3. Obtener la información de las estaciones alcanzables	$O(M)$, $M = \#$ estaciones alcanzables con la condición
TOTAL	$O(E \log V)$

Pruebas Realizadas

Se realizó la prueba en computador con procesador 1.6 GHz Intel Core i5, memoria RAM de 8GB y sistema operativo MacOS High Sierra, usando la base large y con estación inicial 220-116 y 4 conexiones máximas

Entrada	Tiempo (s)
large	0.736

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

```
estacion de origen: 220-116
Número de conexiones permitidas desde la estación origen: 4
las estaciones model
4484
['222-116', '219-116', '221-116', '218-116', '220-116', '216-116', '223-116', '224-116', '217-116']
['41.41351', '41.41117', '41.41265', '41.41065', '41.41208', '41.41172', '41.41398', '41.40834', '41.411']
['2.153389', '2.151857', '2.154626', '2.1514', '2.150317', '2.157128', '2.15772', '2.156479', '2.1556']
[2, 1, 1, 2, 0, 4, 3, 4, 3]
Las estaciones alcanzables son:
['222-116', '219-116', '221-116', '218-116', '220-116', '216-116', '223-116', '224-116', '217-116']
```

Identificador estacion	latitud	longitud	conexiones	distancias
222-116	41.41351	2.153389	2	0.30151663317889354
219-116	41.41117	2.151857	1	0.16349975669925887
221-116	41.41265	2.154626	1	0.36488587909348086
218-116	41.41065	2.1514	2	0.18286798178549113
220-116	41.41208	2.150317	0	0.0
216-116	41.41172	2.157128	4	0.5694004178409237
223-116	41.41398	2.15772	3	0.6524996164654996
224-116	41.40834	2.156479	4	0.6610767571200107
217-116	41.411	2.1556	3	0.4566430306881312

Graficas

Se omite dado que solo se evaluó con large

Análisis

Se omite dado que solo se evaluó con large

Requerimiento 6

Descripción

Se creó un mapa cuyas llaves eran los vecindarios y los valores eran las listas con las estaciones. Después se filtraba por vecindario y para cada elemento de esa lista se codificada la llave única code-bus, y se calculaba el camino de mínima distancia entre la estación de partida y cada estación posible. Finalmente se comparaban estas distancias y se elegía la de menor valor.

Entrada	Estación de origen (code-bus) y vecindario
Salidas	Estaciones que hacen parte del camino, distancia total de este, número de transbordos y distancias entre estaciones
Implementado (Sí/No)	Sí, Ana M.

Análisis de complejidad

Pasos	Complejidad
Paso 1. Crear el mapa de vecindarios	O(1)
Paso 2. Filtrar por vecindario	O(1)

Paso 3. Hallar distancias mínimas entre cada estación posible y estación de origen con algoritmo Dijkstra	$O(E \log V)$, $E = \# \text{ arcos}$, $V = \# \text{ vértices}$
Paso 4. Extraer información de la estación	$O(1)$
TOTAL	$O(E \log V)$

Pruebas Realizadas

Se realizó la prueba en computador con procesador 1.6 GHz Intel Core i5, memoria RAM de 8GB y sistema operativo MacOS High Sierra, usando la base large y con estación inicial 220-116 y final 890-87

Entrada	Tiempo (s)
large	0.793

Tablas de datos

Las tablas con la recopilación de datos de las pruebas.

Distric Name : Les Corts , Neighborhood : les corts }
 Distancia del camino planteado: 19.572334038161873
 El camino planteado tiene 26 estaciones
 El camino planteado tiene 5 transbordos
 Las estaciones del camino planteado son:

Identificador estacion	Distancia a la siguiente estacion	ID_Vecindario
220-116	0.36488587909348086	la Salut
221-116	0.14066173330454448	la Salut
222-116	0.36492760190125545	la Salut
223-116	0.6356213110432379	la Salut
224-116	0.2970830148499219	la Vila de GrÀ cia
225-116	0.7907751926465045	la Vila de GrÀ cia
226-116	1.079870701694794	la Vila de GrÀ cia
227-116	0.09132163827427284	la Salut
228-116	0.48672340021696525	la Salut
883-27	0.3329406113592043	la Vila de GrÀ cia
884-27	0.3445550698025558	la Vila de GrÀ cia
890-87	0.07237139686254128	el Putxet i el FarrÀ³
926-87	0.20728986137210603	la Vila de GrÀ cia
954-87	0.2684526620282339	Vallcarca i els Penitents
985-27	3.3450795107869684	Vallcarca i els Penitents
984-115	0.20657216965817057	la Nova Esquerra de l'Eixample
1637-V7	1.1299462873399242	Sants
1639-V7	1.0206393202600534	les Corts
1640-V7	0.5028234698846936	Sants
1641-V7	0.41274738820668405	les Corts
1642-V7	1.645200717161601	Sants
2298-V7	0.28659798799691255	les Tres Torres
2299-V7	1.2862129307437598	SarriÀ
2300-V7	1.2796075318000315	SarriÀ
2301-V7	0.7794744748165241	les Tres Torres
2302-V7	2.1999521750569317	les Corts

Bienvenido

Graficas

Se omite dado que solo se evaluó con large

Análisis

Se omite dado que solo se evaluó con large

Requerimiento 7

Descripción

Como pasajero deseo encontrar un camino circular saliendo de una estación inicial. Esto me sirve para planear rutas turísticas rápidas dentro de la ciudad. E importante, este camino debe permitirme volver al origen (estación inicial) a través de una estación de transbordo de ser necesario.

Entrada	Estacion origen
Salidas	<ul style="list-style-type: none">La distancia total que tomará el recorrido del camino circular. La distancia total de desplazamiento debe ser mayor a 0.0.El total de estaciones que contiene el camino. El total de estaciones debe ser mayor a 1.El total de transbordos de ruta que deben realizarse.El camino calculado entre las estaciones (incluyendo el origen y el destino) y para cada estación en <p>el camino se debe mostrar la siguiente información:</p> <ul style="list-style-type: none">El identificador de la estación.La distancia a la siguiente estación en el camino.
Implementado (Sí/No)	Sí, Catalina.

Análisis de complejidad

Pasos	Complejidad
Paso 1. dfs	$O(E+V)$
Paso 2. Se recorre el largo del ciclo para sacar el path	$O(N)$
Paso 3. Se recorre el largo del ciclo para imprimir	$O(N)$
TOTAL	$O(E+V)$

Pruebas Realizadas

Entrada	Tiempo (s)
large	0.28478384017944336

Tablas de datos

```
Identificador de la estación de origen: 238-V29
la distancia del camino es: 1.542551651430329
el numero de estaciones en el camino es: 2
el numero de transbordos en el camino es: 0
el camino que se siguió es:
Desde Estacion 238-V29 Hasta Estacion 277-V29 Peso a la siguiente estacion: 0.7712758257151645
Desde Estacion 277-V29 Hasta Estacion 238-V29 Peso a la siguiente estacion: 0.7712758257151645
```

Graficas

Se omite dado que solo se evaluó con large

Análisis

Se omite dado que solo se evaluó con large

Requerimiento 8

Se hace uso de la librería folium y una lista de longitudes y latitudes obtenida de cada requerimiento.

Con ello, se dibujan puntos en el mapa y según si son paradas de transbordo, origen o destino se pintan de color naranja, verde o rojo, respectivamente.

Entrada	Se definió como entrada que el usuario decidiera si deseaba visualizar el mapa una vez ejecutado cada requerimiento
Salidas	El mapa indicando las estaciones a recorrer
Implementado (Sí/No)	Si. Yaisa C. Ramírez

Análisis de complejidad

Se define N como el número de estaciones de la ruta obtenida en el requerimiento

Pasos	Complejidad
Paso 1. Recorrer la lista de latitudes y longitudes	$O(N)$, se sabe que $N \ll V$
Paso 2. Agregar coordenada al mapa	$O(1)$
TOTAL	$O(N)$

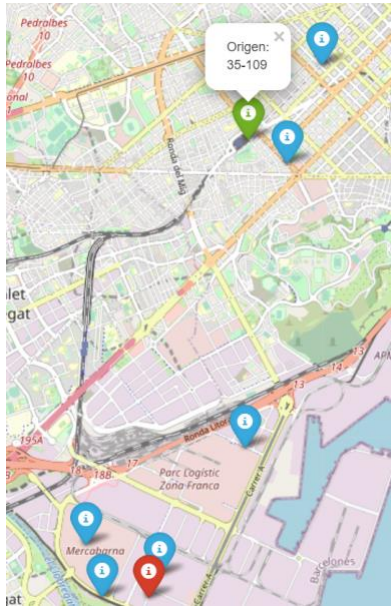
Pruebas Realizadas

Entrada	Tiempo (s)
large	0.566

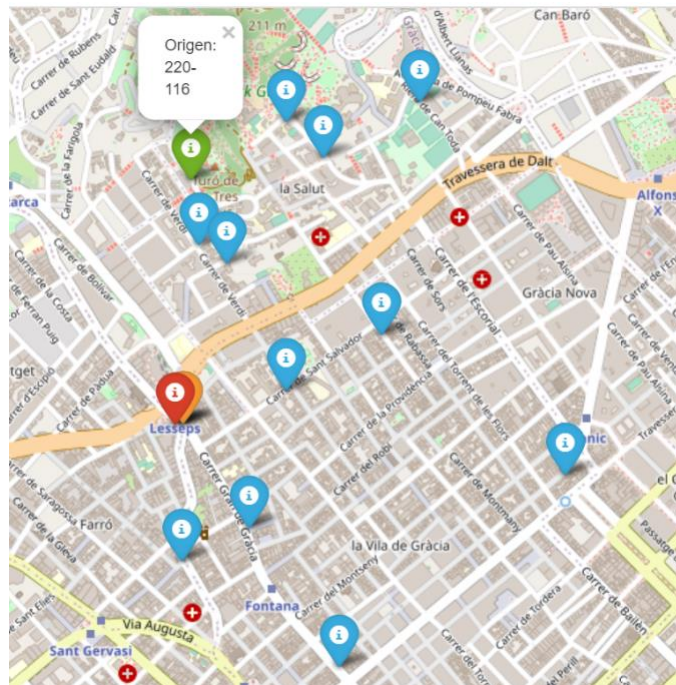
Tablas de datos

Es importante aclarar que, si en el recorrido existían estaciones de transbordo, estas se dibujaban de naranja y se omitían las paradas pertenecientes a la misma estación de transbordo.

Mapa del requerimiento 2 usando como entradas 35-109 y 42-109

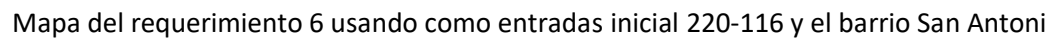
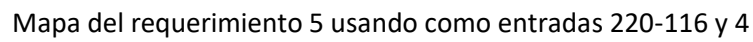


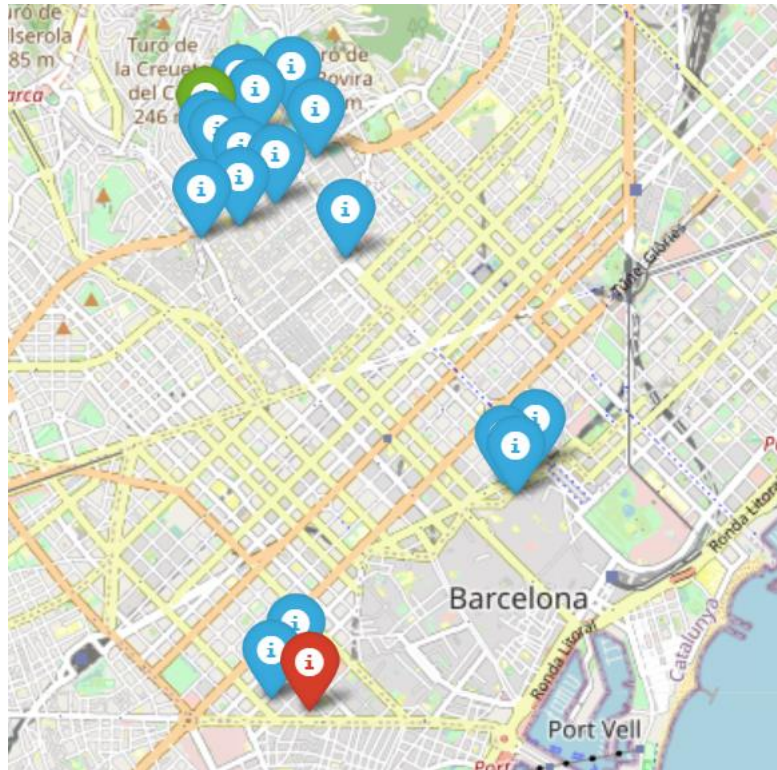
Mapa del requerimiento 2 usando como entradas 220-116 y 890-87



Mapa del requerimiento 4 usando como entradas

[Long, Lat] (destino): [2.13802, 41.3742]





Graficas

No se realizaron gráficas. Sin embargo, como era de esperarse, entre más estaciones tenía la ruta, mayor era el tiempo que tomaba la función en finalizar.