

2022



Alejandro Arenas Vasco

Movilidad eficiente al alcance  
de todos.

# Una breve presentación

Alejandro Arenas Vasco (aarenas2@eafit.edu.co)

Ingeniero de producción – Universidad EAFIT

Máster en estadística e investigación operativa – Universidad  
Politécnica de Catalunya

Estudiante de doctorado de ingeniería matemática – Universidad  
EAFIT





# Motivación

---

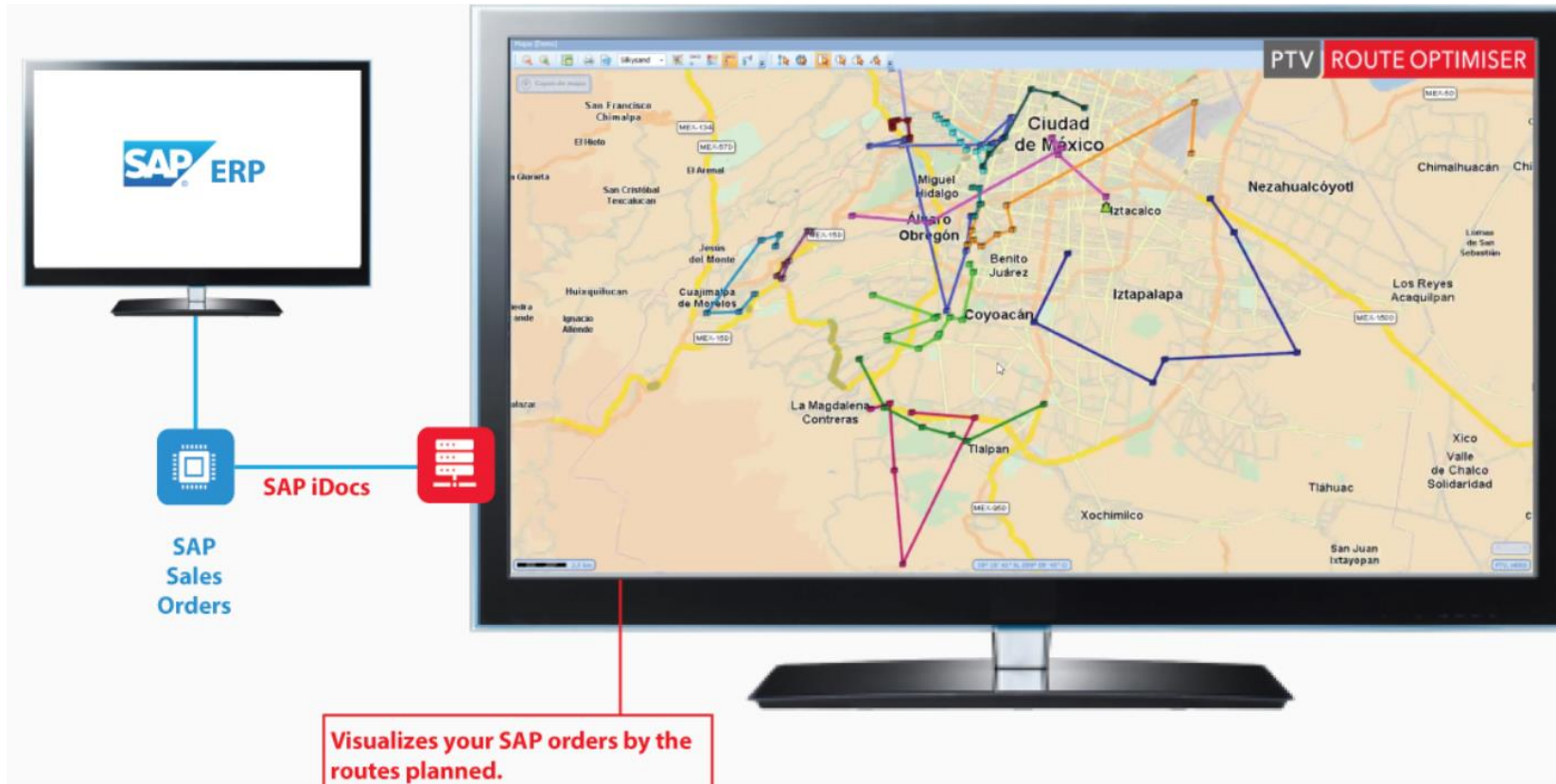
# Motivación

Respondamos la siguiente pregunta:

Si estuviesen manejando y quisieran ir a Sabaneta, ¿cómo buscarían la mejor ruta?



# Motivación



Cambiamos la pregunta:

¿Cómo organiza una multinacional las rutas de sus vehículos?



# Motivación

Lugar	Demanda (cajas)
UPB	0
Colegio San ignacio	5
Inem	3
Colegio Maria Montessori	2
Colegio la Inmaculada	4
Universidad de Medellín	5
Universidad Nacional	7
Politecnico Jaime Isaza Cadavid	5
Universidad de Antioquia	9
Universidad CES	1
EAFIT	3

Y... si la flota son dos motocargueros para entregar los siguientes paquetes en 10 clientes diferentes:



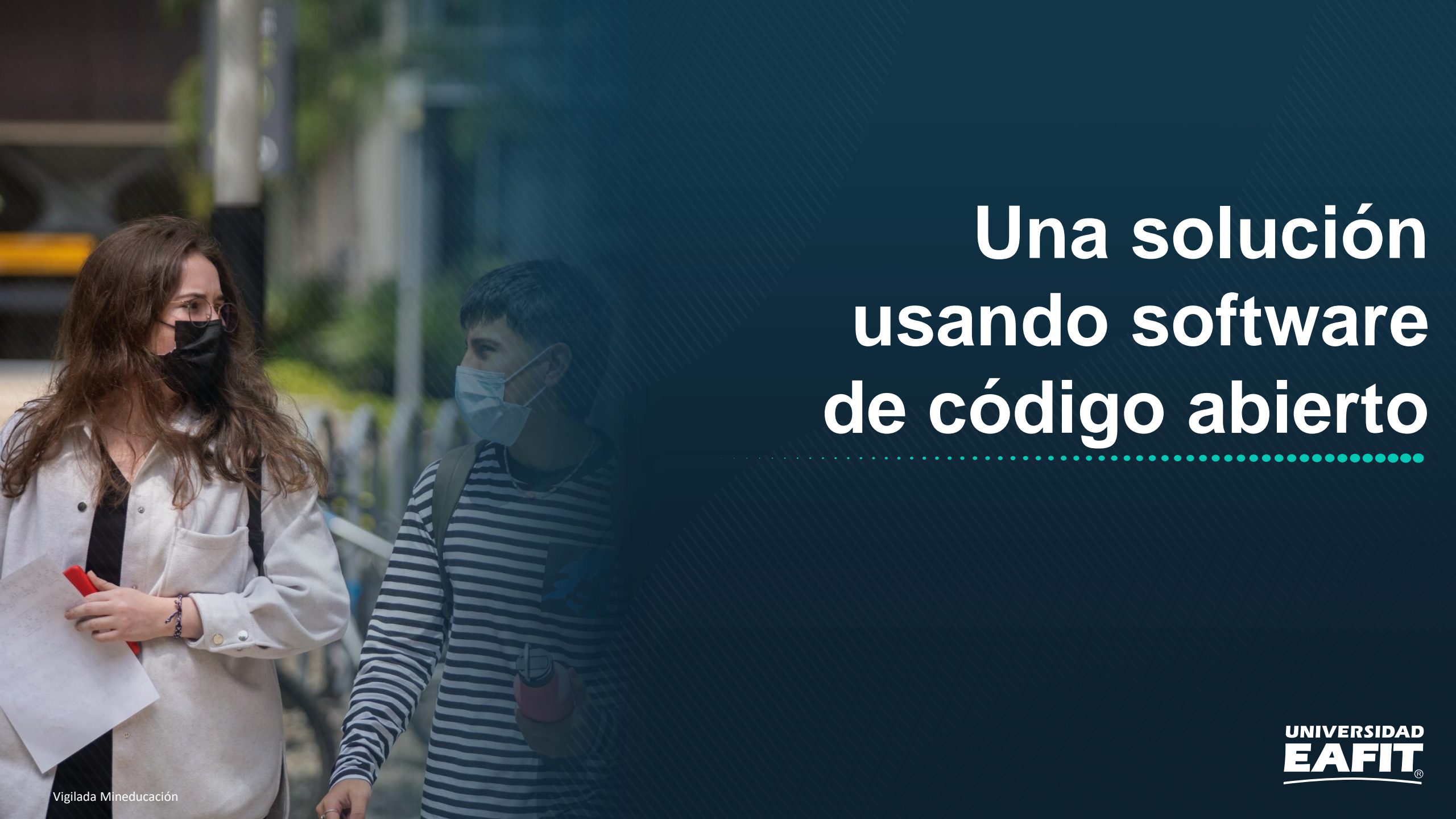
# Motivación



Lugar	Demanda (cajas)
UPB	0
Colegio San ignacio	5
Inem	3
Colegio Maria Montessori	2
Colegio la Inmaculada	4
Universidad de Medellín	5
Universidad Nacional	7
Politecnico Jaime Isaza Cadavid	5
Universidad de Antioquia	9
Universidad CES	1
EAFIT	3

Tienen 15 minutos para organizar la ruta de menos kilómetros usando dos vehículos con capacidad máxima para 30 cajas.

Sólo pueden usar un mapa de la ciudad.



# Una solución usando software de código abierto

---



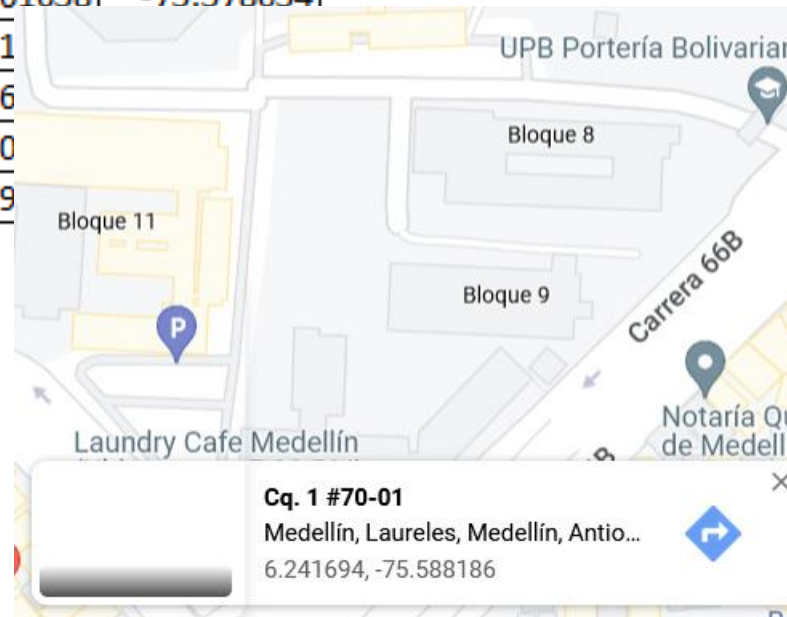
# Solución

Para la solución es necesario:



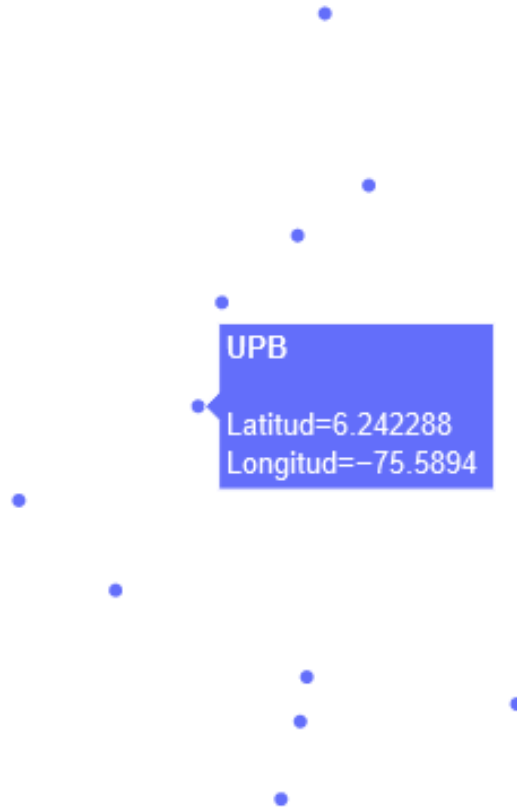
# Solución

Lugar	Latitud	Longitud
UPB	6.242288	-75.589395
Colegio San ignacio	6.254069	-75.586685
Inem	6.206538	-75.577755
Colegio Maria Montessori	6.286836	-75.57494
Colegio la Inmaculada	6.221419	-75.598803
Universidad de Medellín	6.231613	-75.609844
Universidad Nacional	6.261658	-75.578054
Politecnico Jaime Isaza Cadavid	6.21	
Universidad de Antioquia	6.26	
CES	6.20	
EAFIT	6.19	



Inicialmente se pueden obtener las geocordenadas de cada uno de los puntos requeridos. Este proceso es sencillo usando Google Maps.

# Solución

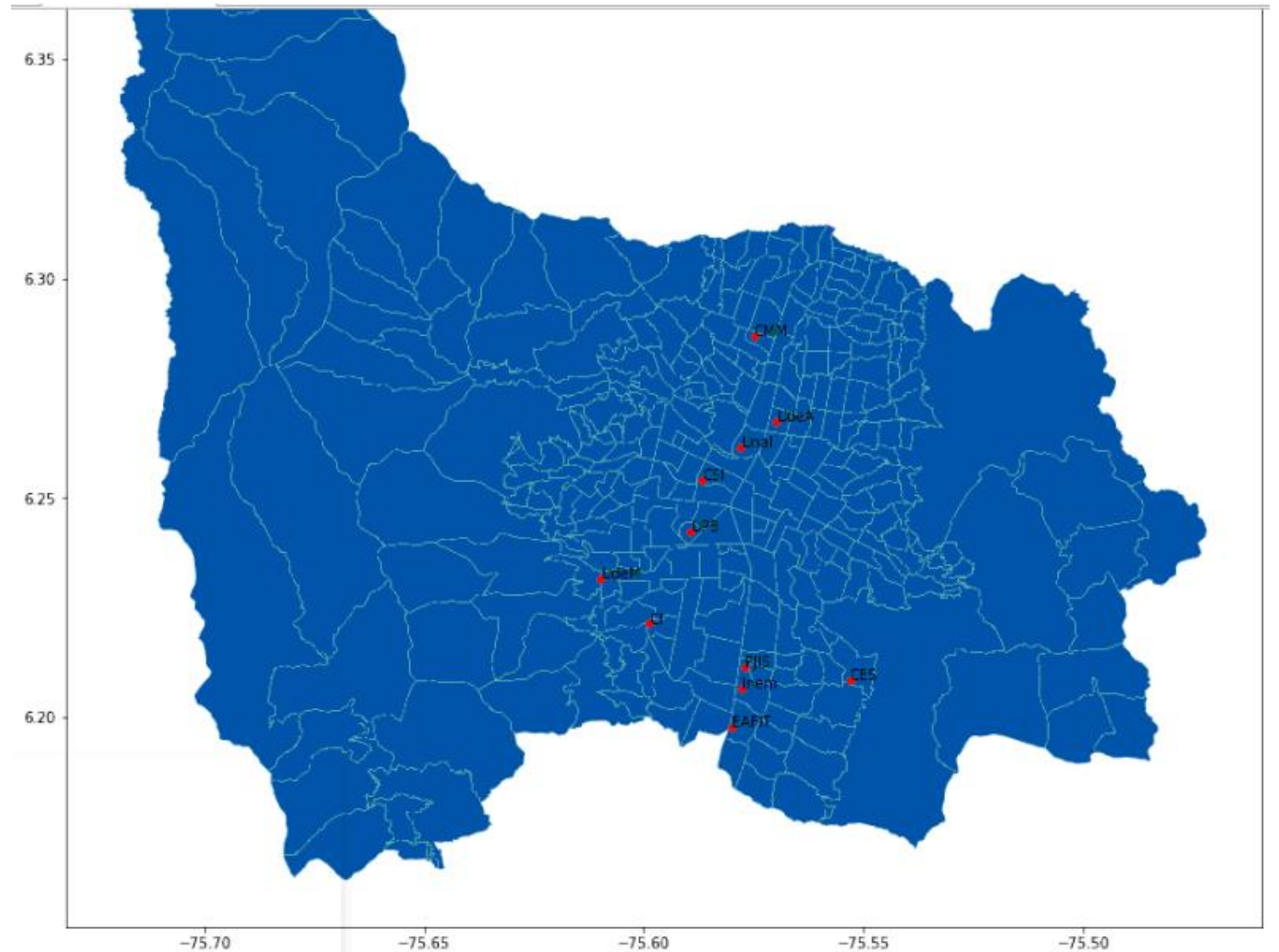


Se puede obtener una  
noción espacial de nuestros  
clientes usando un sencillo  
Script:



# Solución

Si la idea es conocer la ubicación dentro de Medellín, se puede usar un Script diferente:



# Solución

Ya tenemos mejores referencias, sin embargo algunas preguntas surgen:

¿Qué es ser más eficiente?

¿Qué información requiero?

¿Cómo lo soluciono?



# Ruteo

---



# Ruteo

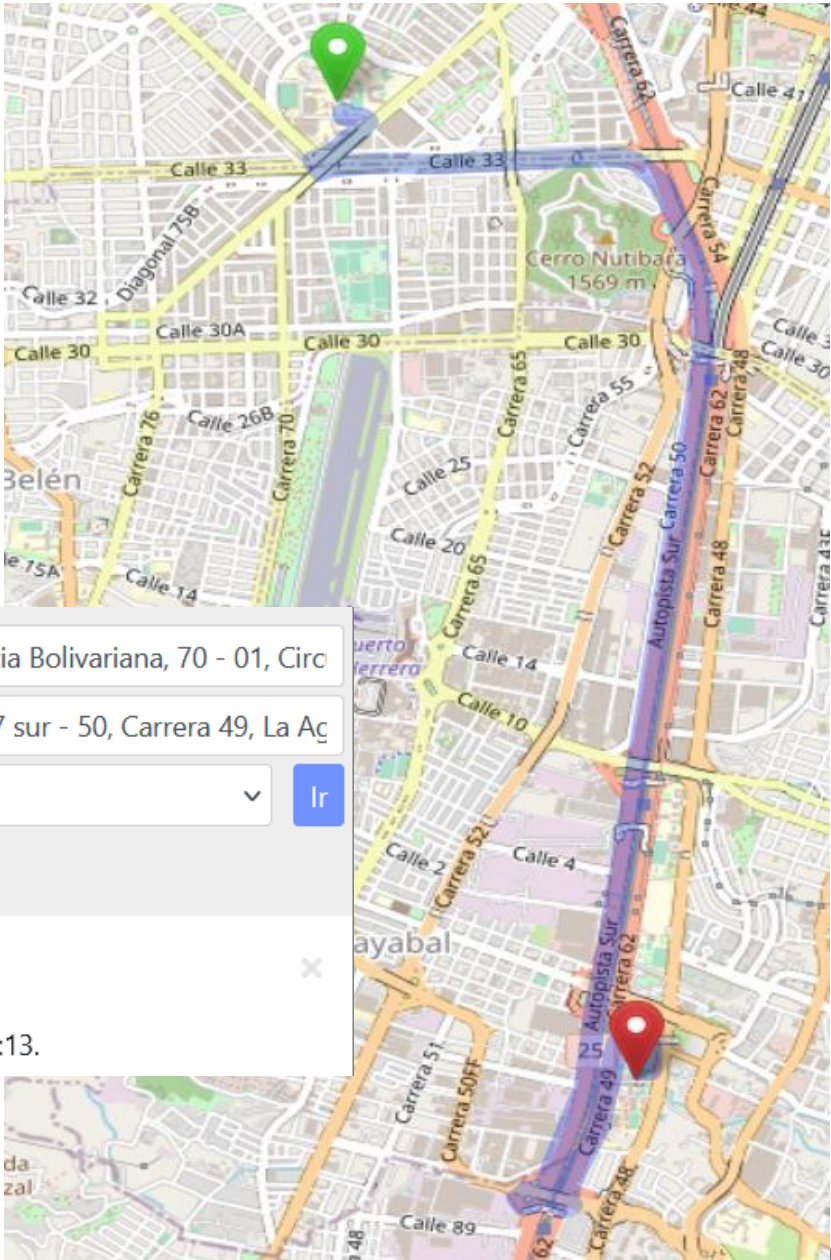
Una buena forma de empezar es saber cuántos kilómetros hay entre cada uno de los puntos del problema.

¿Qué consideraciones debe tener esta distancia?

	UPB	CSI	INEM	CMM	CI	UdeM	Unal	PJIC	UdeA	CES	EAFIT	UPB
UPB	0	2.401	7.22	7.703	3.945	3.923	3.061	6.668	7.168	8.122	8.355	0
CSI	1.617	0	7.627	5.285	6.004	5.219	2.092	7.075	6.532	9.051	8.469	1.617
INEM	6.5	6.417	0	10.548	6.575	6.937	7.045	0.567	10.013	4.499	2.396	6.5
CMM	6.542	5.233	10.727	0	9.017	8.995	3.82	10.175	3.996	12.151	11.569	6.542
CI	3.887	6.257	4.573	10.339	0	2.688	5.822	5.141	9.929	7.758	4.685	3.887
UdeM	4.222	5.329	6.489	9.411	2.349	0	6.157	7.057	10.264	8.792	6.601	4.222
Unal	5.08	1.709	9.265	3.723	7.555	7.533	0	8.713	5.076	10.689	10.107	5.08
PJIC	7.137	7.054	1.822	11.185	6.908	7.574	7.682	0	10.65	3.207	2.729	7.137
UdeA	5.887	4.578	10.072	4.828	8.362	8.34	3.129	9.52	0	9.921	10.914	5.887
CES	8.338	8.483	5.265	12.638	7.498	8.783	9.135	4.034	12.103	0	5.312	8.338
EAFIT	9.568	9.485	3.075	13.612	6.26	8.132	10.109	3.642	13.077	6.753	0	9.568
UPB	0	2.401	7.22	7.703	3.945	3.923	3.061	6.668	7.168	8.122	8.037	0

# Ruteo

Obtener estas distancias no es difícil (ni costoso). Para ellos se pueden usar páginas de código libre como [www.openstreetmap.org](http://www.openstreetmap.org)



The screenshot shows a map of Medellín, Colombia, with a blue route highlighted. The route starts at a green pin at the top left and ends at a red pin at the bottom right. The map includes labels for various streets such as Calle 33, Calle 30, Calle 20, Calle 14, Calle 10, Calle 4, Calle 2, Calle 89, Carrera 65, Carrera 55, Carrera 52, Carrera 50, Carrera 48, Carrera 46, Carrera 44, Carrera 42, Carrera 40, Carrera 38, Carrera 36, Carrera 34, Carrera 32, Carrera 30, Carrera 28, Carrera 26, Carrera 24, Carrera 22, Carrera 20, Carrera 18, Carrera 16, Carrera 14, Carrera 12, Carrera 10, Carrera 8, Carrera 6, Carrera 4, Carrera 2, Carrera 0, Carrera -2, Carrera -4, Carrera -6, Carrera -8, Carrera -10, Carrera -12, Carrera -14, Carrera -16, Carrera -18, Carrera -20, Carrera -22, Carrera -24, Carrera -26, Carrera -28, Carrera -30, Carrera -32, Carrera -34, Carrera -36, Carrera -38, Carrera -40, Carrera -42, Carrera -44, Carrera -46, Carrera -48, Carrera -50, Carrera -52, Carrera -54, Carrera -56, Carrera -58, Carrera -60, Carrera -62, Carrera -64, Carrera -66, Carrera -68, Carrera -70, Carrera -72, Carrera -74, Carrera -76, Carrera -78, Carrera -80, Carrera -82, Carrera -84, Carrera -86, Carrera -88, Carrera -90, Carrera -92, Carrera -94, Carrera -96, Carrera -98, Carrera -100. The route is labeled with 'Autopista Sur' and 'Carrera 49'. A green pin is located at the top left, and a red pin is located at the bottom right. The route is highlighted in blue. The map also shows a river and some green spaces.

Universidad Pontificia Bolivariana, 70 - 01, Circ

Universidad EAFIT, 7 sur - 50, Carrera 49, La Ag

Automóvil (OSRM) Ir

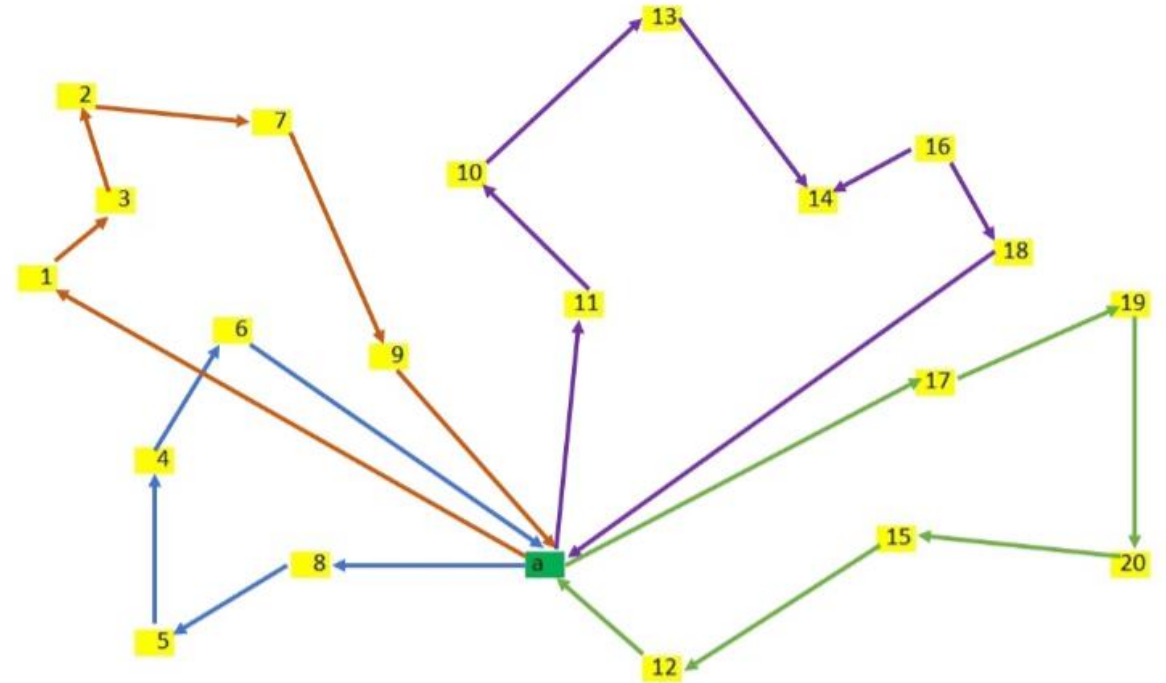
[Indicaciones inversas](#)

**Indicaciones**

Distancia: 8.4km. Hora: 0:13.

# Ruteo

Fue propuesto en 1959 por George Dantzig y J. H. Ramser. Consiste en atender todos los clientes en un grafo con una flota de vehículos y desde un depósito.





# Ruteo

El VRP, visto desde la optimización, es un problema NP-hard (no se puede resolver en tiempo polinomial).

A continuación se presenta una de sus formulaciones.

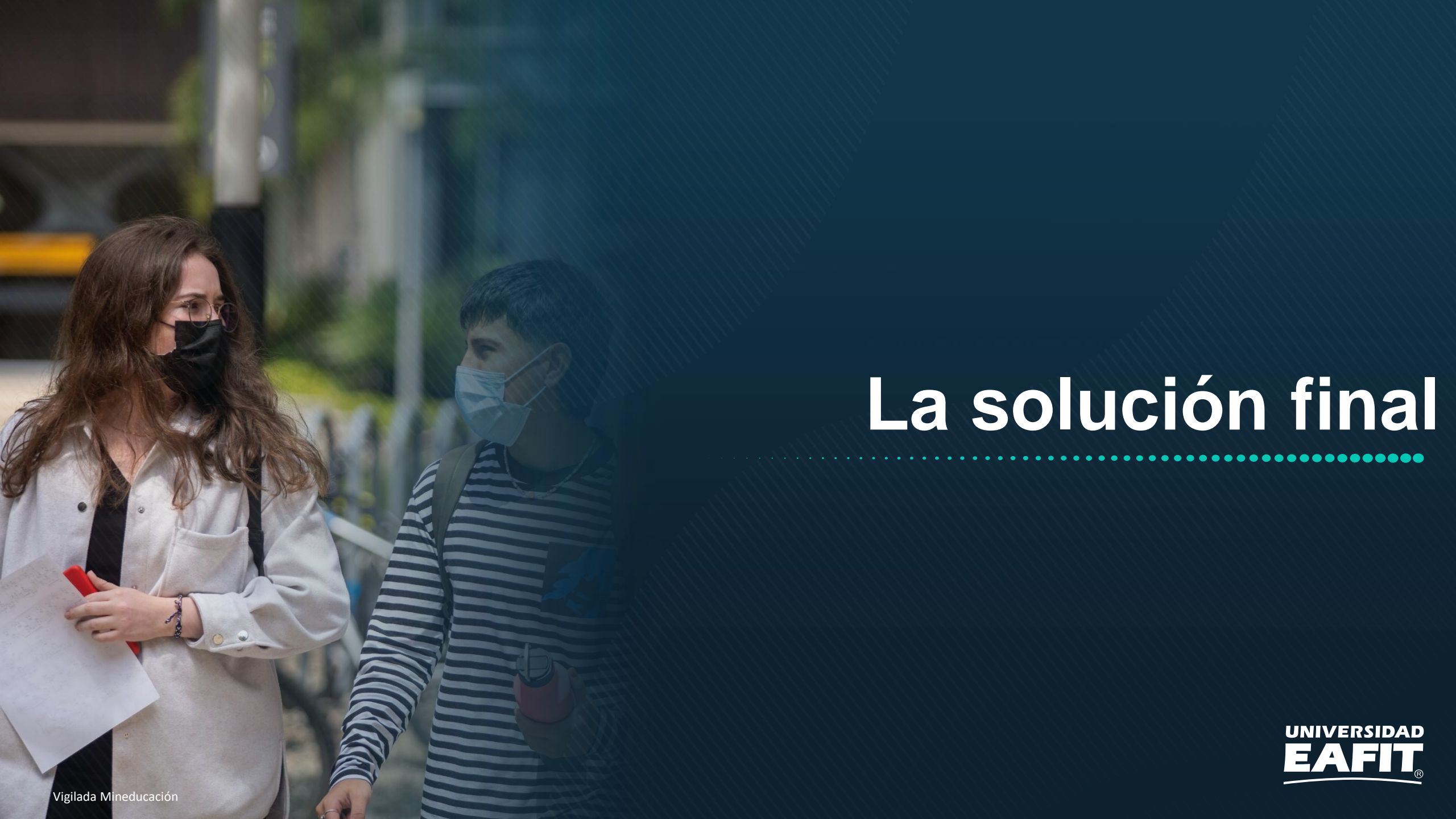
The variables for the MDMVRP are exactly the same as in the MDVRP. Meanwhile, the formulation is:

$$\begin{aligned} \text{MDMVRP} &:= \min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} C_{ij}^k X_{ij}^k \\ \text{subject to } &\sum_{k \in K} \sum_{j \in N} X_{ij}^k = 1 \quad \forall i \in C \\ &\sum_{i \in C} d_i \sum_{j \in N} X_{ij}^k \leq q_k \quad \forall k \in K \\ &\sum_{j \in N} X_{hj}^k = W_h^k \quad \forall h \in D, \forall k \in K \\ &\sum_{i \in N} X_{ih}^k = W_h^k \quad \forall h \in D, \forall k \in K \\ &\sum_{i \in N} X_{ih}^k - \sum_{j \in N} X_{hj}^k = 0 \quad \forall h \in C, \forall k \in K \\ &X_{ij}^k \in \{0,1\} \quad \forall i, j \in N, \forall k \in K \\ &\sum_{j \in N} W_i^k \leq \maxdep_o_i \quad \forall i \in D \\ &U_i = 1 \quad \forall i \in D \\ &2 \leq U_i \leq |N| \quad \forall i \in C \\ &U_j \geq (U_i + 1) - M \left( 1 - \sum_{k \in V} X_{ij}^k \right) \quad \forall i \in N, \forall j \in C \end{aligned} \quad (26)$$

# Ruteo

Sin embargo, el ruteo ha encontrado su camino en los softwares de código abierto y sin necesidad de la formulación abstracta:





# La solución final

---



# Solución

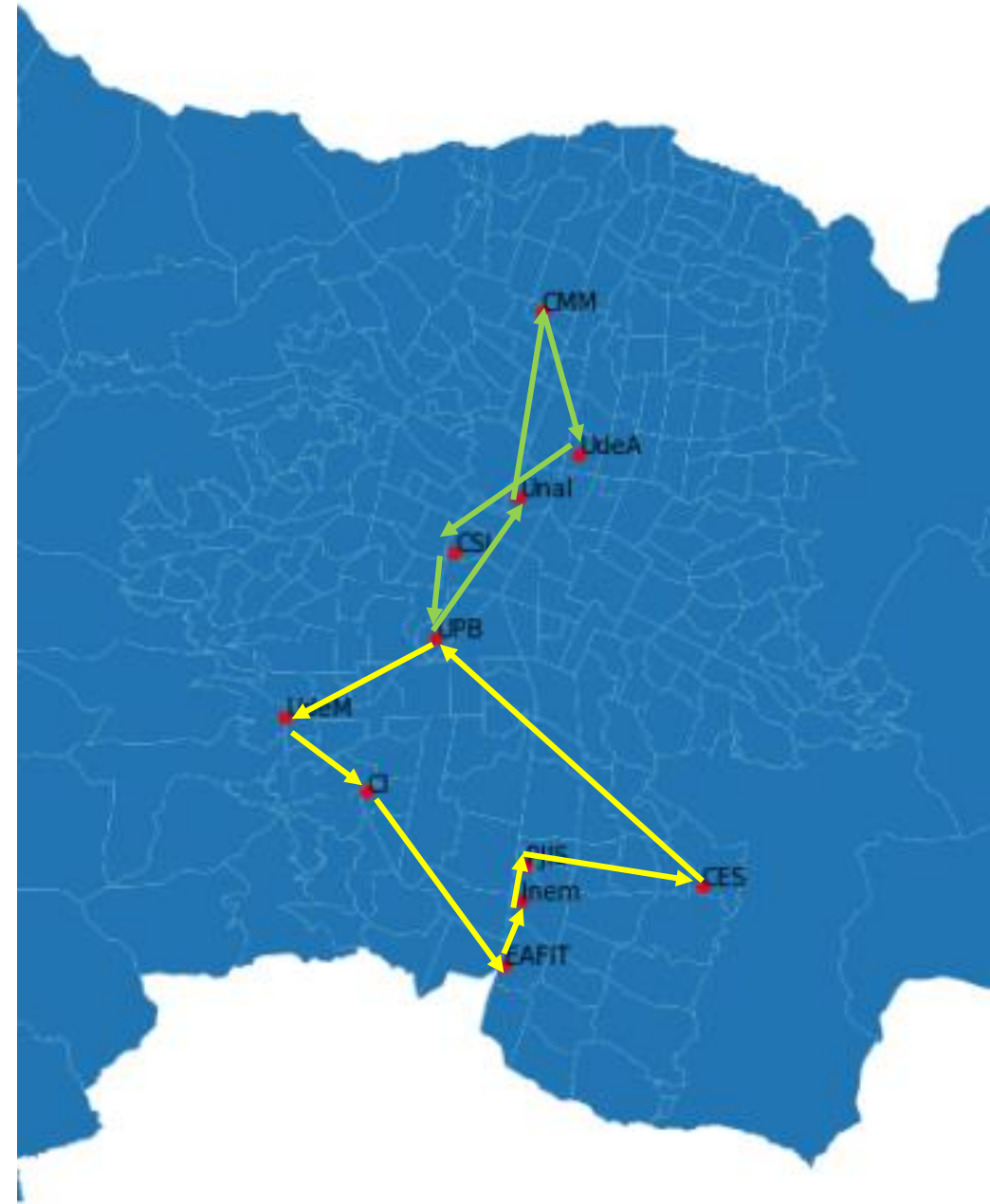
La solución con menor kilometraje es:

```
INFO:vrpy.master_solve_pulp:total cost = 43.119  
43.119  
{1: ['Source', 5, 4, 10, 2, 7, 9, 'Sink'], 2: ['Source', 6, 3, 8, 1, 'Sink']}
```

R1 26.144

R2 16.975

```
In [11]: print(prob.best_routes_load)  
{1: 21, 2: 23}
```





# ¿Alguna pregunta?

---





# GRACIAS

UNIVERSIDAD  
**EAFIT**<sup>®</sup>