

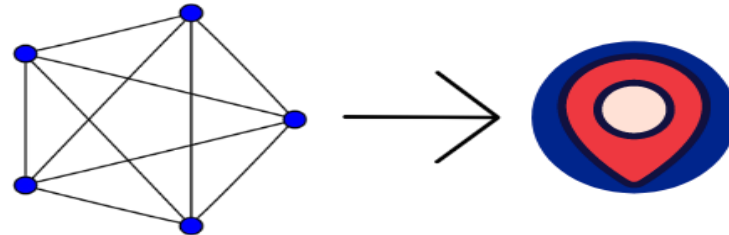
**Algoritmo para disminuir el tráfico urbano asignando
rutas de vehículos compartido a los empleados de una
empresa**

***Juan Camilo Guerrero Alarcón
Juan Jose Escudero Valencia
Santiago Pulgarín Vásquez***

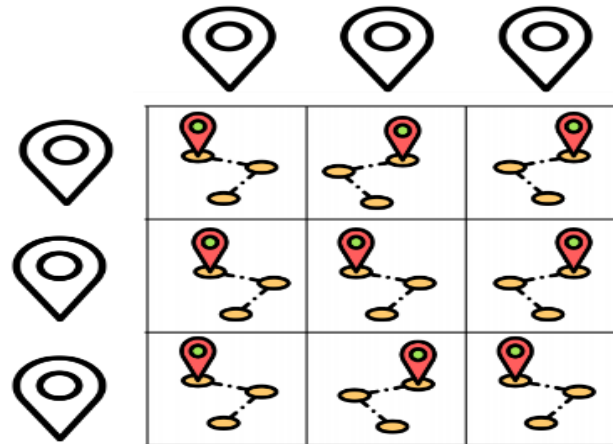
Medellín, Antioquia

Mayo 16, 2019

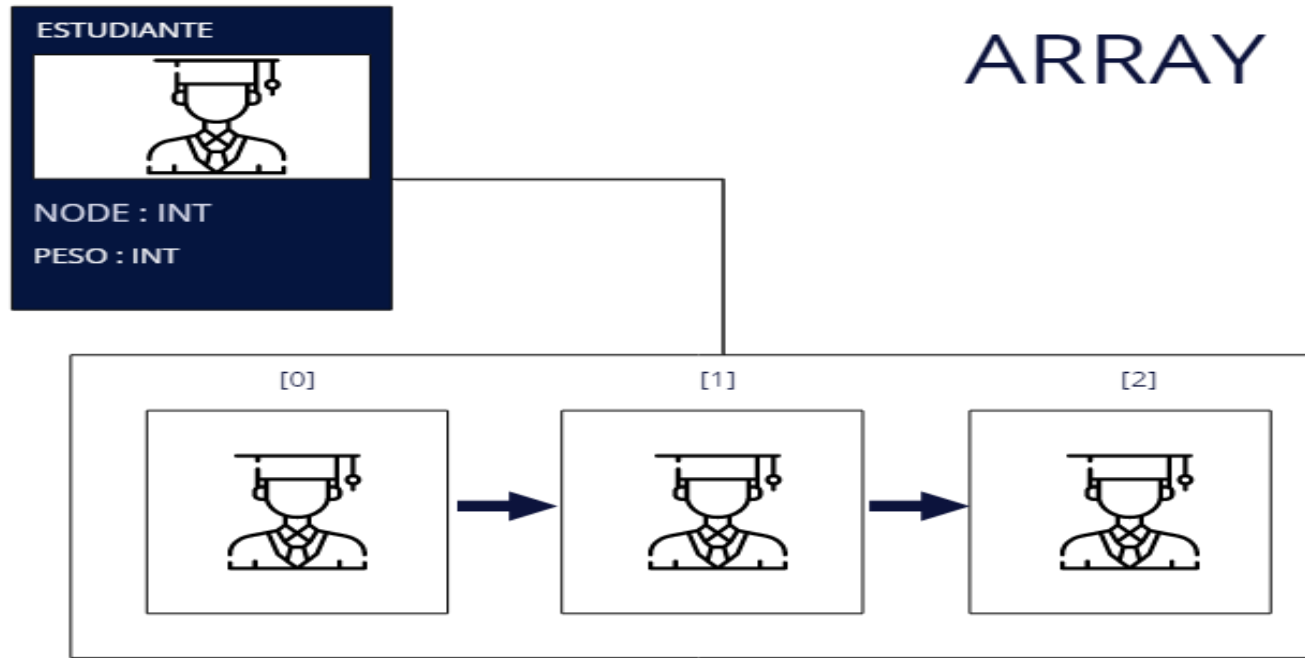
Estructuras de Datos Diseñada



MATRIZ

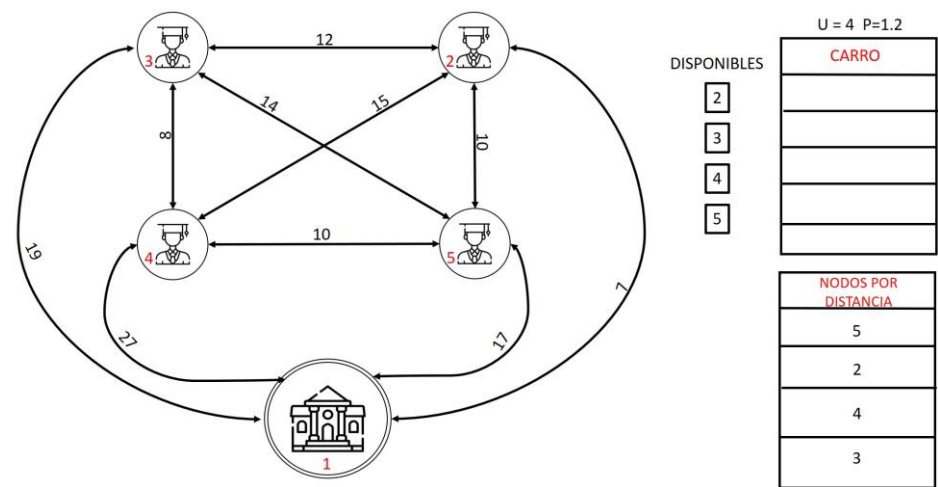


Grafica 1: Implementación de la matriz para guardar el grafo



Grafica 1.1: Implementación del array como estructura de datos

Explicación del algoritmo y su complejidad



Método	Complejidad
Adición en matriz	O(1)
Adición en arreglo	O(1)
Obtención de matriz	O(1)
Obtención de arreglo	O(1)
Ordenamiento de arreglo	O(nlogn)
Asignación de vehículos	O(n^2)

Gráfico 2: Inserción de los nodos en un array llamado Nodos por Distancia

Tabla 1: Complejidad del algoritmo para el peor de los casos, el mejor y el caso promedio

Explicación del algoritmo

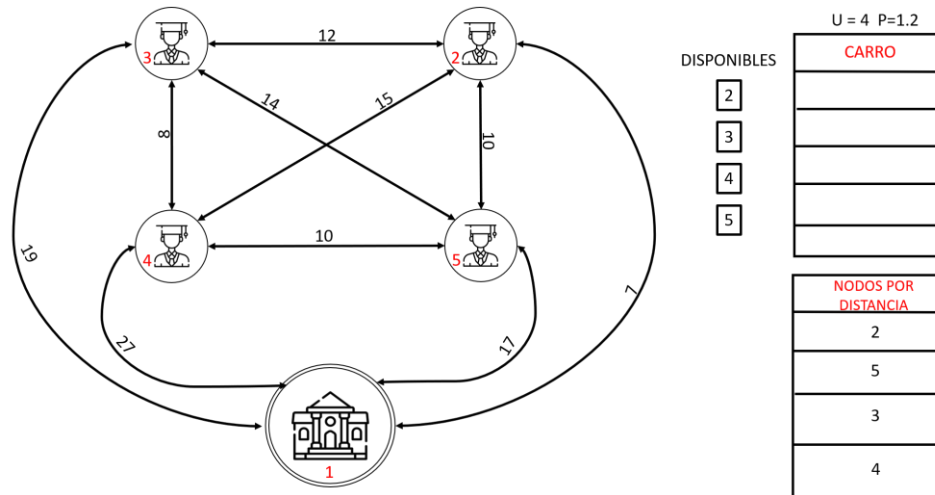


Gráfico 3: Ordenamiento descendente de los nodos en el array con respecto a su distancia a la Universidad

Explicación del algoritmo

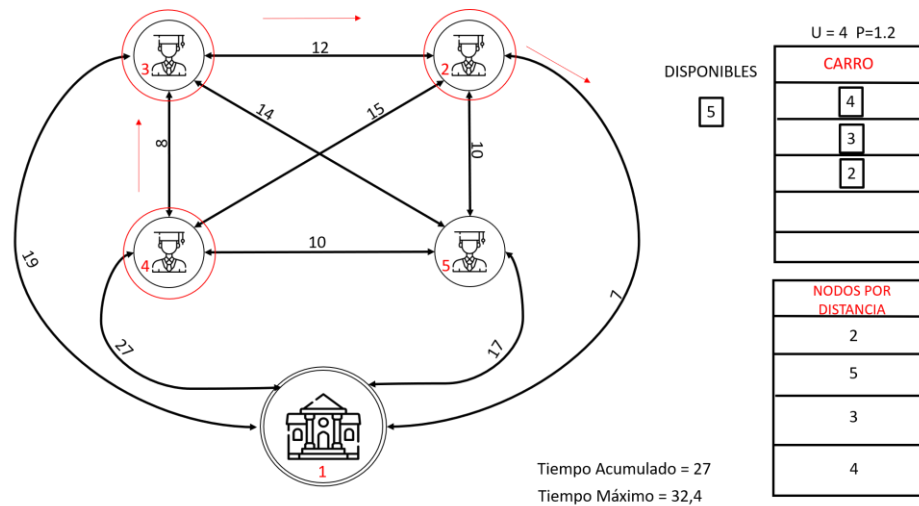


Gráfico 4: *Escogencia de la ruta más corta y asignación a un carro*

Explicación del algoritmo

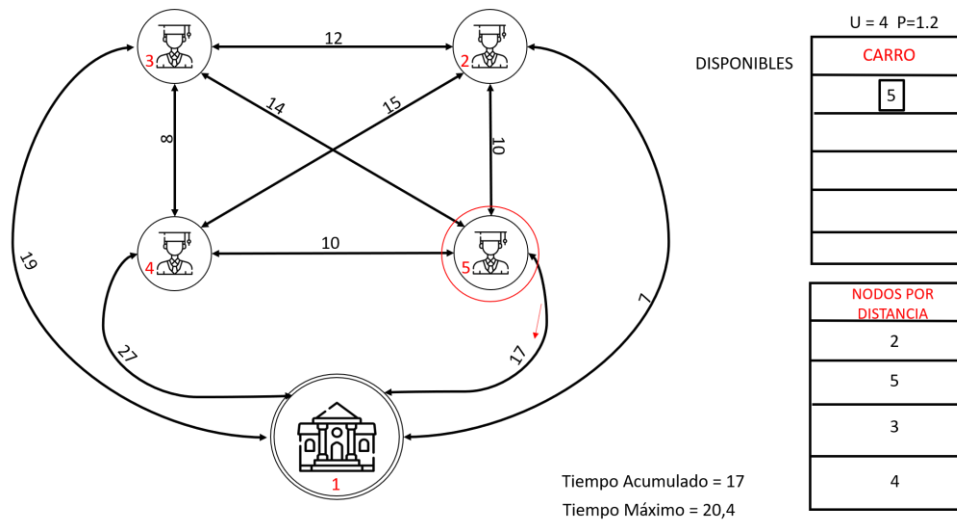


Gráfico 5: Escogencia de la ruta más corta con el nodo disponible y asignación a un carro

Complejidad

Sub problema	Complejidad
Leer el dataset y crear la estructura de datos	$O(n)$
Ordenar el arreglo de sucesores	$O(n \log n)$
Llenar los carros	$O(n^2)$
Complejidad Total	$O(n^2)$

Tabla 2: Complejidad del algoritmo para el peor de los casos, el mejor y el caso promedio

Criterios de Diseño del Algoritmo

El algoritmo se diseñó de esta manera debido a que almacenar el grafo mediante una matriz de adyacencia le brinda alta eficiencia en tiempo y memoria, además el hecho de que el algoritmo escoja a los nodos más distantes de la universidad como posibles candidatos a ser conductores en donde posteriormente centrado en el nodo conductor buscara cual es el más cercano a él, revisando en todos los casos que no exceda el tiempo máximo permitido para ese nodo y que cada vez se esté acercando más a la universidad nos asegura una alta precisión en el número optimo total de vehículos y en el tiempo de ejecución del algoritmo.

Consumo de Tiempo

	Constante (p) = 1,1	Constante (p) = 1,2	Constante (p) = 1,3
Mejor caso	0,76 ms	0,77 ms	1,23 ms
Caso promedio	2,43 ms	2,74 ms	2,74 ms
Peor caso	23,2 ms	32,1 ms	38,7 ms

Tabla 3: consumo de tiempo del algoritmo con 205 usuarios.

Consumo de Memoria

	Constante (p) = 1,1	Constante (p) = 1,2	Constante (p) = 1,3
Consumo de memoria	16,7 MB	16,6 MB	16,1 ms

Tabla 4: consumo de tiempo del algoritmo con 205 usuarios.

Resultados

Dataset	Resultados
$U = 5$ y $P = 1,2$	3 carros
$U = 5$ y $P = 1,7$	2 carros
$U = 11$ y $P = 1,1$	6 carros
$U = 11$ y $P = 1,2$	6 carros
$U = 11$ y $P = 1,3$	5 carros
$U = 205$ y $P = 1,1$	56 carros
$U = 205$ y $P = 1,2$	52 carros
$U = 520$ y $P = 1,3$	51 carros

Tabla 5: Resultados obtenidos.

Conclusiones

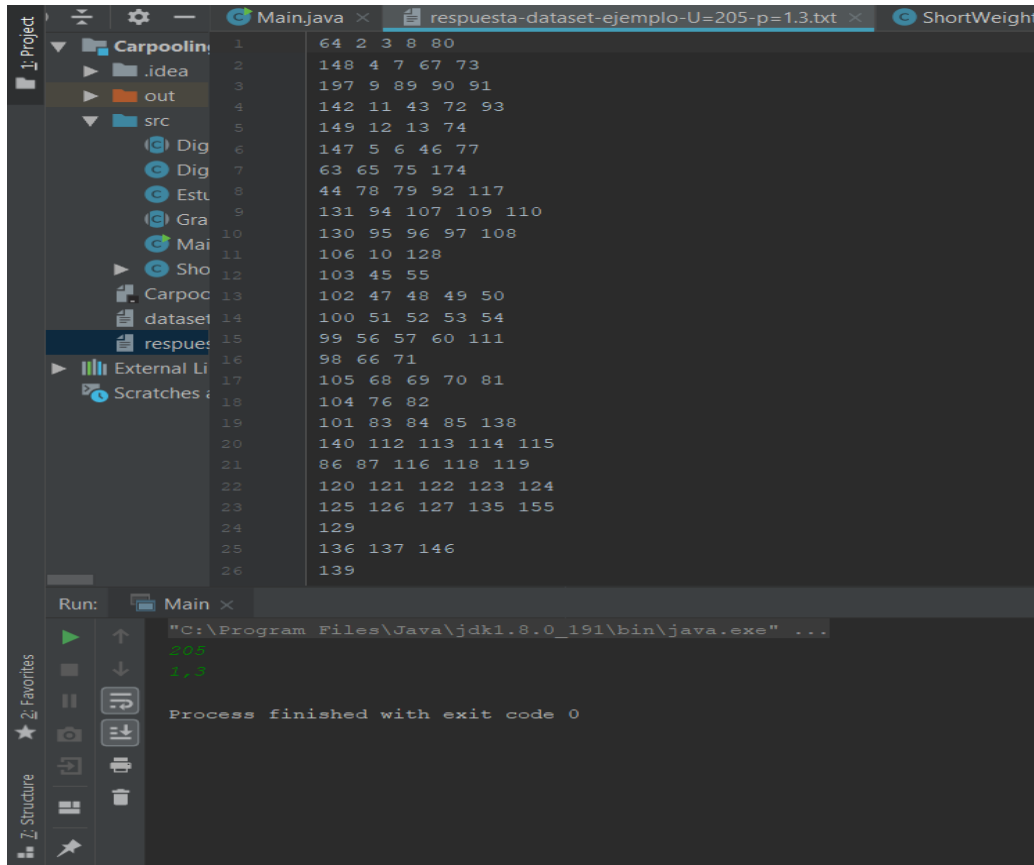


Mediante la implementación del algoritmo se dejaron de emitir mas del 50% de gases contaminantes.



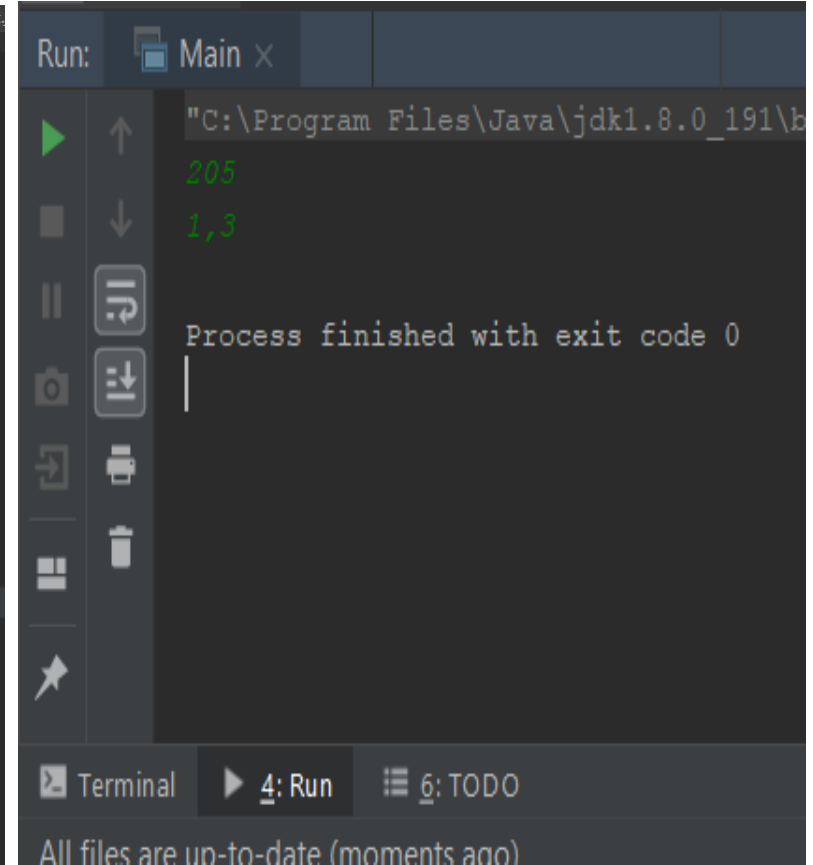
Mediante la implementación del algoritmo se redujo el trafico en mas del 50%.

Software en funcionamiento



The screenshot shows an IDE with a project named 'Carpooling'. The 'src' folder contains a file named 'respuesta-dataset-ejemplo-U=205-p=1.3.txt'. The file's contents are displayed in the editor, showing a list of numbers arranged in columns. The 'Run' tab at the bottom shows the command: `"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...` and the output: `205`, `1,3`, and `Process finished with exit code 0`.

```
1 64 2 3 8 80
2 148 4 7 67 73
3 197 9 89 90 91
4 142 11 43 72 93
5 149 12 13 74
6 147 5 6 46 77
7 63 65 75 174
8 44 78 79 92 117
9 131 94 107 109 110
10 130 95 96 97 108
11 106 10 128
12 103 45 55
13 102 47 48 49 50
14 100 51 52 53 54
15 99 56 57 60 111
16 98 66 71
17 105 68 69 70 81
18 104 76 82
19 101 83 84 85 138
20 140 112 113 114 115
21 86 87 116 118 119
22 120 121 122 123 124
23 125 126 127 135 155
24 129
25 136 137 146
26 139
```



The screenshot shows the 'Run' tab of an IDE. The command is: `"C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...`. The output is: `205`, `1,3`, and `Process finished with exit code 0`. The status bar at the bottom indicates 'All files are up-to-date (moments ago)'.

```
Run: "C:\Program Files\Java\jdk1.8.0_191\bin\java.exe" ...
205
1,3
Process finished with exit code 0
```

Gráfico 5: Software implementado en funcionamiento