# Punto 2. Ruta más Corta (25 puntos)

## Nota

Todos los algoritmos fueron implementados en Java y su código con datos se encuentra en el siguiente repositorio: <https://github.com/Otros-Codigos/Algoritmos-Log-stica>

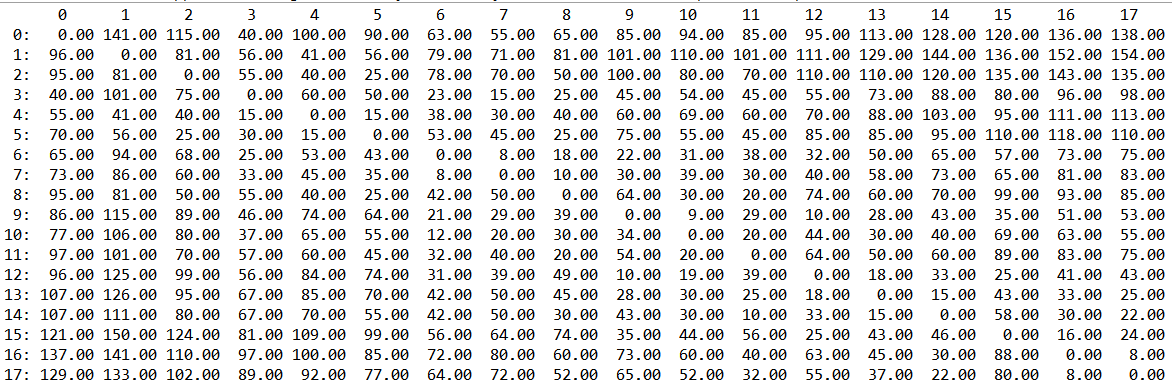
Para su implementación cada vértice notado alfabéticamente tuvo su respectiva referencia numérica empezando desde el 0.

## Literal A

Luego de realizar el Algoritmo Floyd-Warshall por medio de una implementación en Java se puede concluir las siguientes rutas saliendo de A, B, y C para llegar a Q.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Rutas** | | | | | |
| **Desde A** | | **Desde B** | | **Desde C** | |
| **Id** | **Vértice** | **Id** | **Vértice** | **Id** | **Vértice** |
| 0 | A | 1 | B | 2 | C |
| 3 | D | 4 | E | 5 | F |
| 7 | H | 3 | D | 8 | I |
| 6 | G | 7 | H | 10 | K |
| 9 | J | 6 | G | 13 | N |
| 12 | M | 9 | J | 17 | R |
| 15 | P | 12 | M | 16 | Q |
| 16 | Q | 15 | P |  |  |
|  |  | 16 | Q |  |  |
|  |  |  |  |  |  |
| Costo | 136 | Costo | 152 | Costo | 143 |

Esto se reafirma con la siguiente matriz de ruteo más corto y las rutas ya especificadas arrojadas por consola una vez se ejecuto el algoritmo.

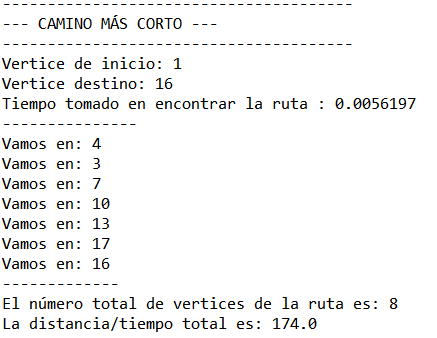


## Literal B

Luego de realizar los cambios necesarios para los nodos J y L y el uso de Dijkstra se obtuvo la siguiente ruta más corta:

|  |  |
| --- | --- |
| **Ruta** | |
| **Id** | **Vértice** |
| 1 | B |
| 4 | E |
| 3 | D |
| 7 | H |
| 10 | K |
| 13 | N |
| 17 | R |
| 16 | Q |
| Costo | 174 |

Esto se reafirma con el resultado por consola arrojado por la ejecución del algoritmo.



## Literal C

Con el fin de determinar el costo máximo que igualmente minimice la ruta se hizo uso de dos escenarios. Primero, el escenario donde la ruta no existe. Segundo, el escenario donde si existe, pero su costo es cero. La diferencia entre el costo total de las dos rutas es el máximo costo del arco NP que da el mismo resultado; no obstante, como se busca minimizar el peso máximo es la diferencia menos uno.

Estos tres escenarios se realizaron por medio del algoritmo Dijkstra y su respectiva implementación. El resultado fue que el peso (tiempo para este caso) del arco NP es de 16.

Los resultados de los escenarios y rutas fueron:

|  |  |
| --- | --- |
| **Ruta Original** | |
| **Id** | **Vértice** |
| 1 | B |
| 4 | E |
| 3 | D |
| 7 | H |
| 10 | K |
| 13 | N |
| 17 | R |
| 16 | Q |
| Costo | 174 |

|  |  |  |
| --- | --- | --- |
| **Ruta Reducida Ideal con arco NP** | | **Costo NP** |
| **Id** | **Vértice** | 0 |
| 1 | B |  |
| 4 | E |  |
| 3 | D |  |
| 7 | H |  |
| 10 | K |  |
| 13 | N |  |
| 15 | P |  |
| 16 | Q |  |
| Costo | 157 |  |

|  |  |  |
| --- | --- | --- |
| **Ruta Reducida con arco NP** | | **Costo Máx. NP** |
| **Id** | **Vértice** | 16 |
| 1 | B |  |
| 4 | E |  |
| 3 | D |  |
| 7 | H |  |
| 10 | K |  |
| 13 | N |  |
| 15 | P |  |
| 16 | Q |  |
| Costo | 173 |  |

Esto se reafirma con los resultados obtenidos por consola al correr el algoritmo en cada escenario:

