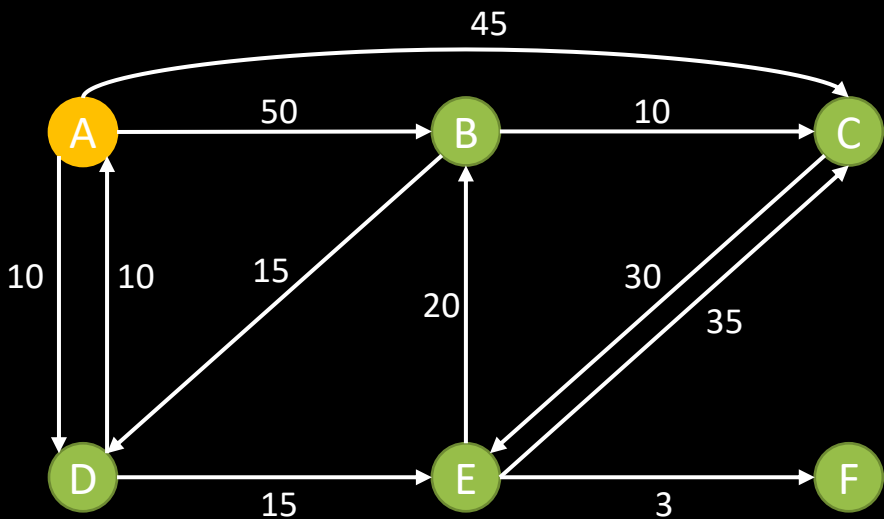


El camino más corto desde un punto específico

- Algoritmo Dijkstra
 - Calcula la distancia más corta desde un nodo inicial hacia todos los nodos del grafo.

Algoritmo Dijkstra

la representacion del grafo ya esta



faltata el analisis del algoritmo

que esta aca, igual el mostrar la tabala igual ya esta

Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | ∞ | |
| C | ∞ | |
| D | ∞ | |
| E | ∞ | |
| F | ∞ | |

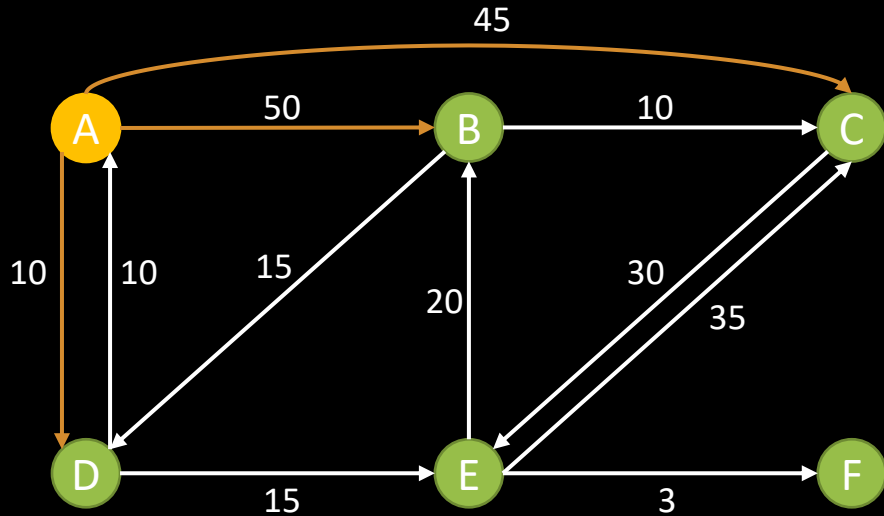
representacion de estas colas igual

Nodos visitados = []

Nodos por visitar = [A,B,C,D,E,F]

Algoritmo Dijkstra

A con quien se conecta para poder elegir al primre nodo, medinate la lista de adyacencia



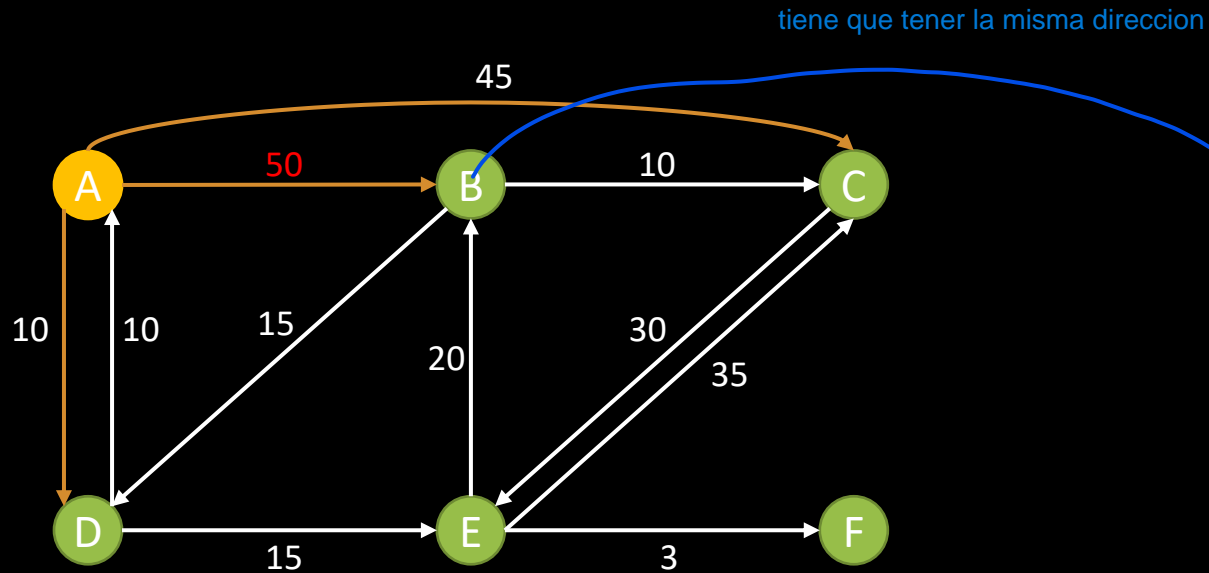
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | ∞ | |
| C | ∞ | |
| D | ∞ | |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = []

Nodos por visitar = [A,B,C,D,E,F]

Algoritmo Dijkstra



$$d(A,B) = 50 < \infty \quad \checkmark$$

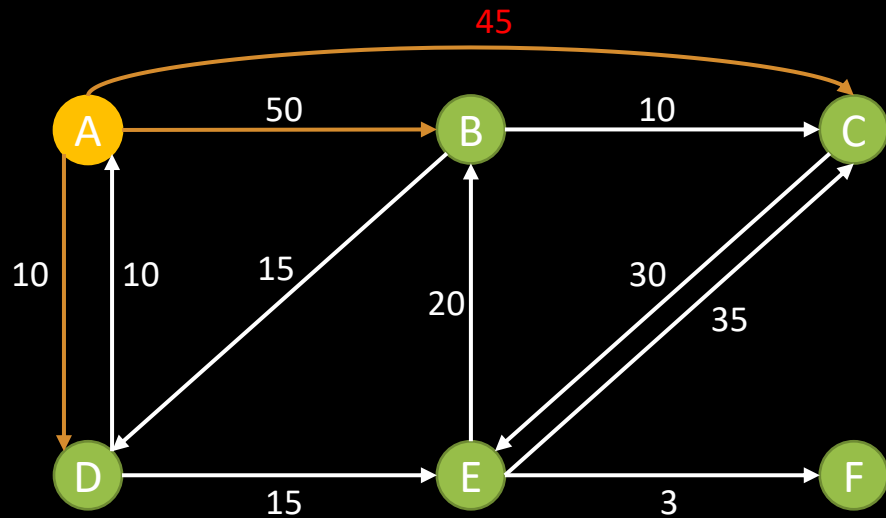
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | ∞ | |
| D | ∞ | |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = []

Nodos por visitar = [A,B,C,D,E,F]

Algoritmo Dijkstra



$$d(A,C) = 45 < \infty \quad \checkmark$$

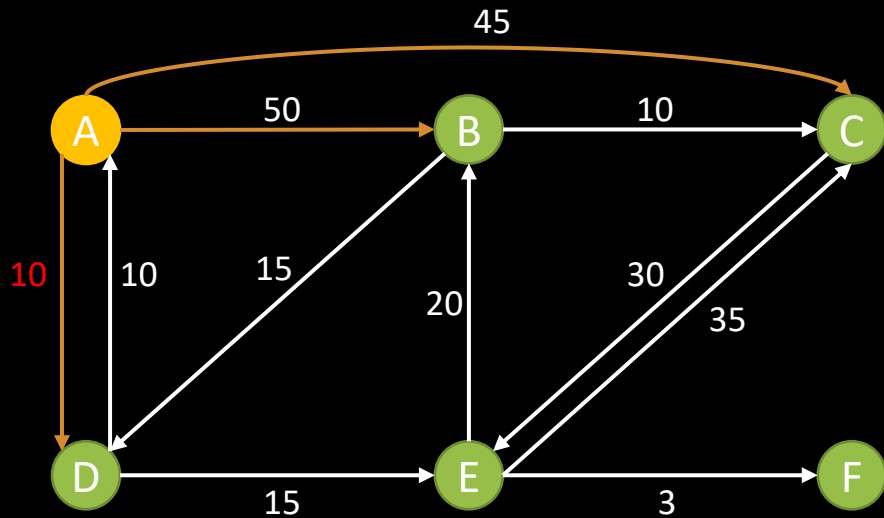
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | ∞ | |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = []

Nodos por visitar = [A,B,C,D,E,F]

Algoritmo Dijkstra



$$d(A,D) = 10 < \infty \quad \checkmark$$

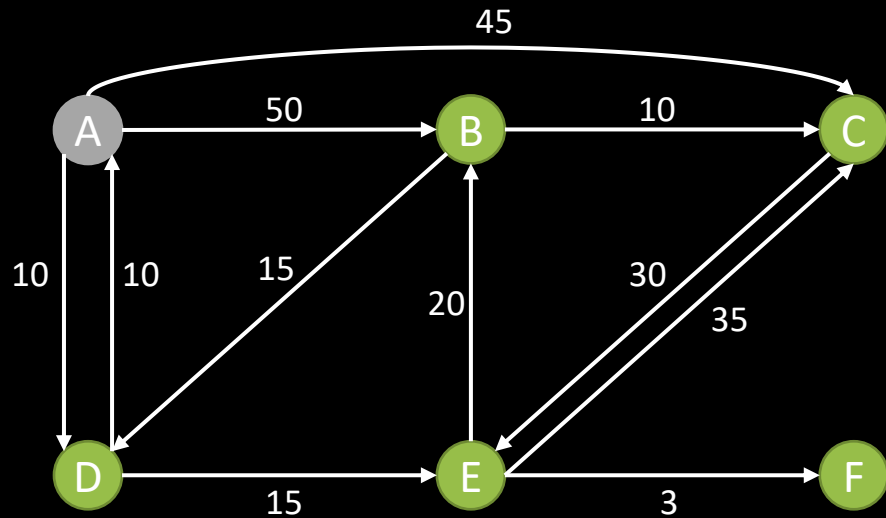
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = []

Nodos por visitar = [A,B,C,D,E,F]

Algoritmo Dijkstra



Para continuar vamos a seleccionar al nodo con la distancia más corta de los nodos por visitar.

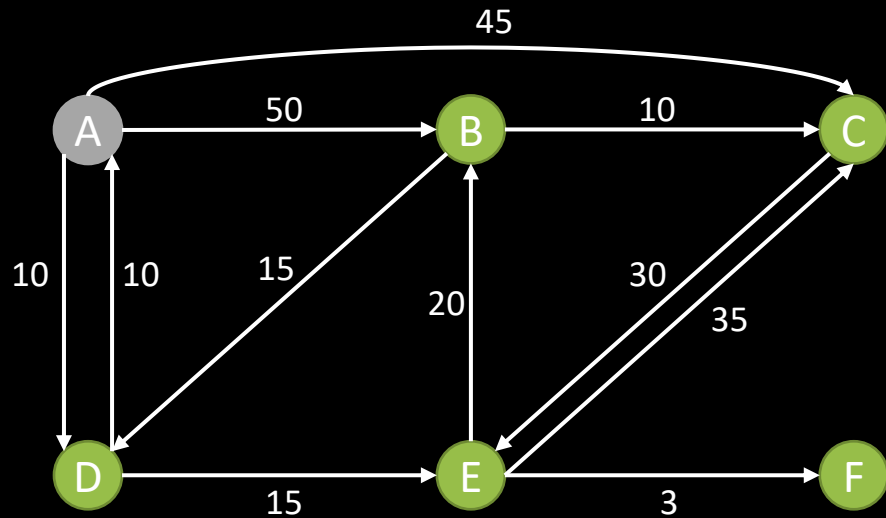
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------|-------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = [A]

Nodos por visitar = [B,C,D,E,F]

Algoritmo Dijkstra



Para continuar vamos a seleccionar al nodo con la distancia más corta de los nodos por visitar.

Nodo D

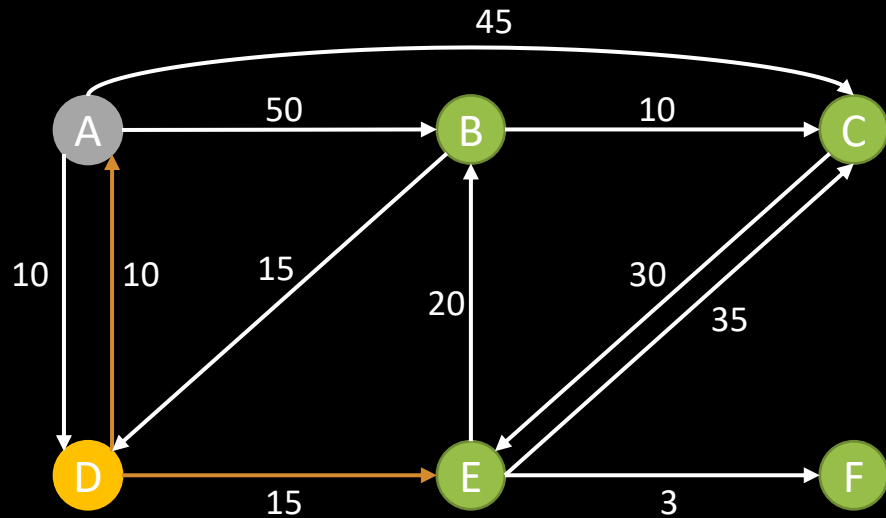
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------|-------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = [A]

Nodos por visitar = [B,C,D,E,F]

Algoritmo Dijkstra



Salimos del nodo A

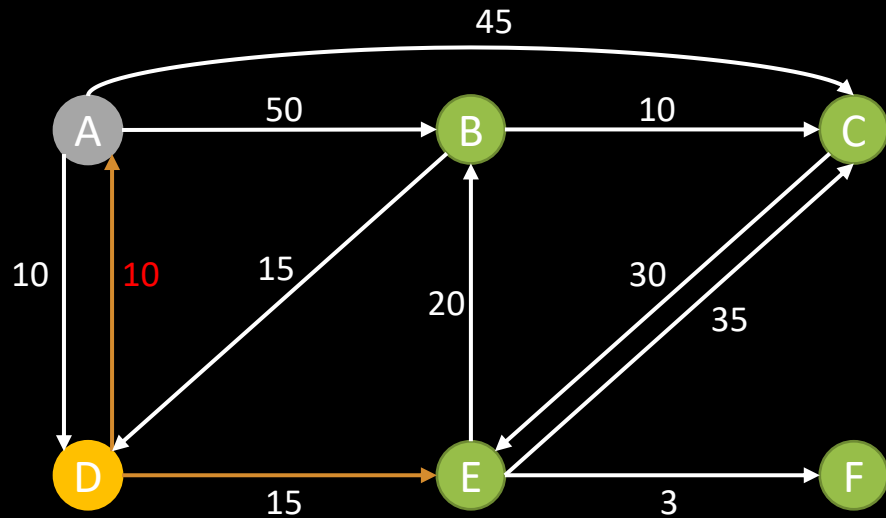
| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | ∞ | |
| F | ∞ | |

Nodos visitados = [A]

Nodos por visitar = [B,C,D,E,F]



Algoritmo Dijkstra



Salimos del nodo A

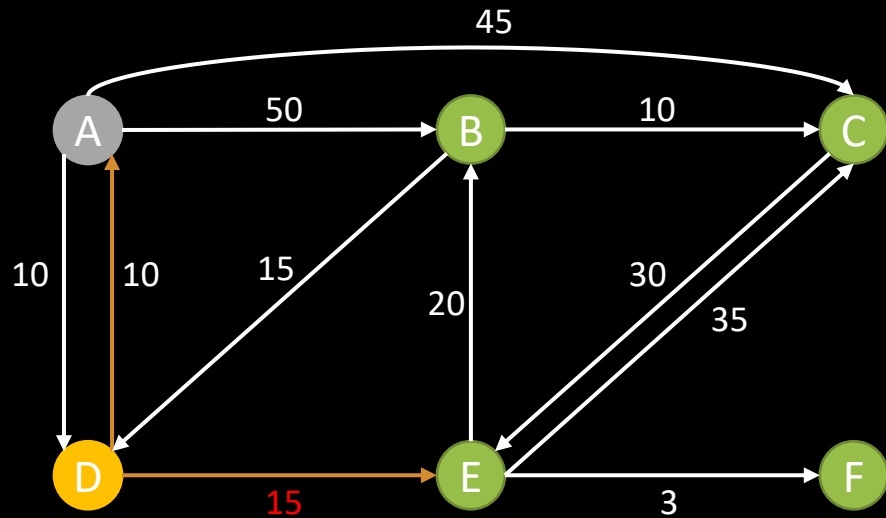
| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | ∞ | |
| F | ∞ | |

Descartamos la arista porque se dirige hacia un nodo visitado.

Nodos visitados = [A]

Nodos por visitar = [B,C,D,E,F]

Algoritmo Dijkstra



$$d(A,E) = d(A,D) + d(D,E) = 10 + 15 = 25 < \infty \quad \checkmark$$

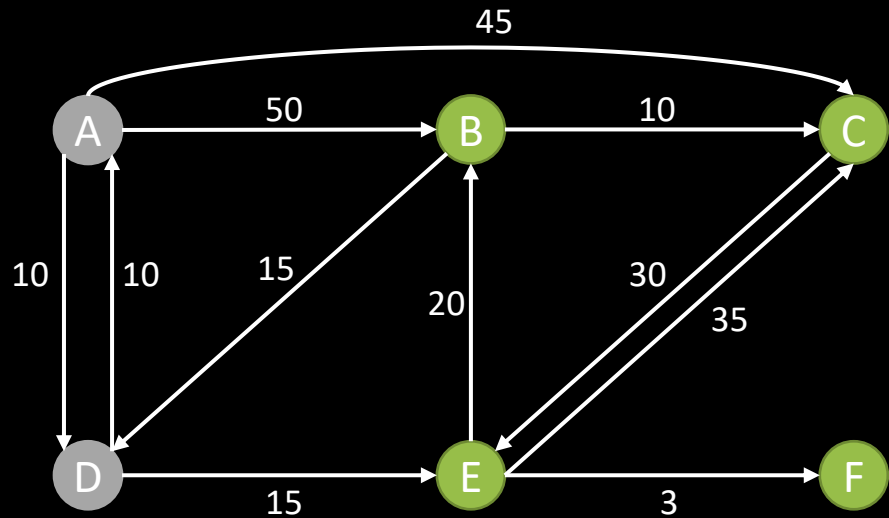
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | ∞ | |

Nodos visitados = [A]

Nodos por visitar = [B,C,D,E,F]

Algoritmo Dijkstra



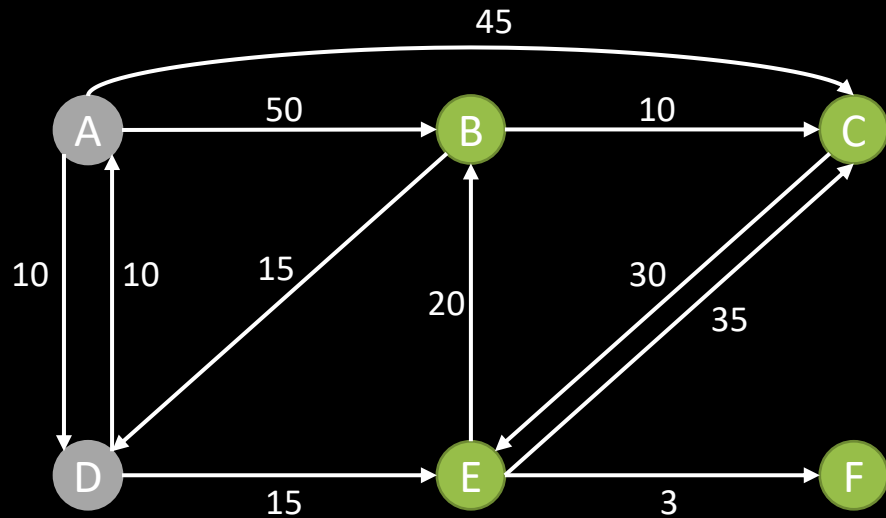
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | ∞ | |

Nodos visitados = [A,D]

Nodos por visitar = [B,C,E,F]

Algoritmo Dijkstra



Para continuar vamos a seleccionar al nodo con la distancia más corta de los nodos por visitar.

Nodo E

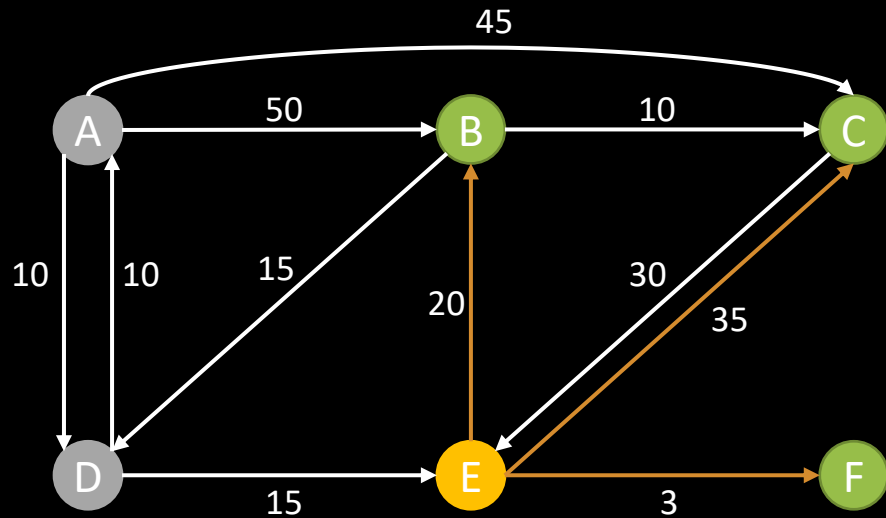
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------|-------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | ∞ | |

Nodos visitados = [A,D]

Nodos por visitar = [B,C,E,F]

Algoritmo Dijkstra



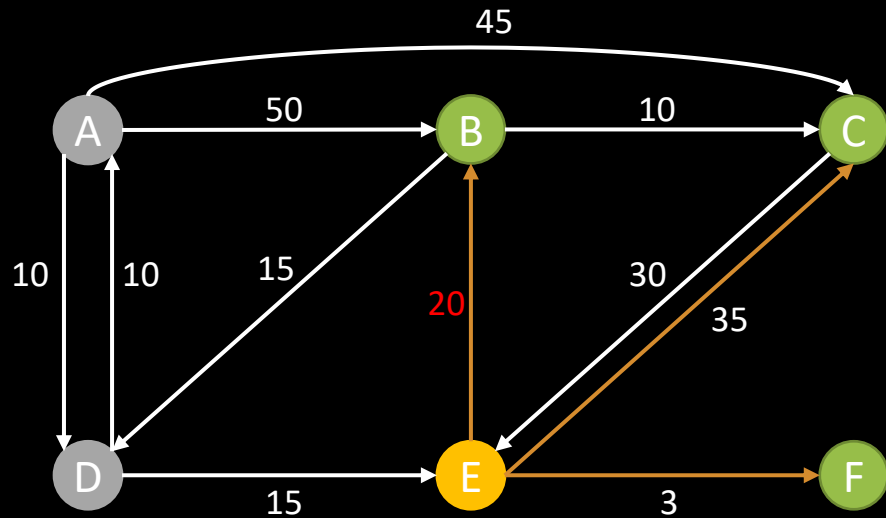
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 50 | A |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | ∞ | |

Nodos visitados = [A,D]

Nodos por visitar = [B,C,E,F]

Algoritmo Dijkstra



$$d(A,B) = d(A,E) + d(E,B) = 25 + 20 = 45 < 50 \quad \checkmark$$

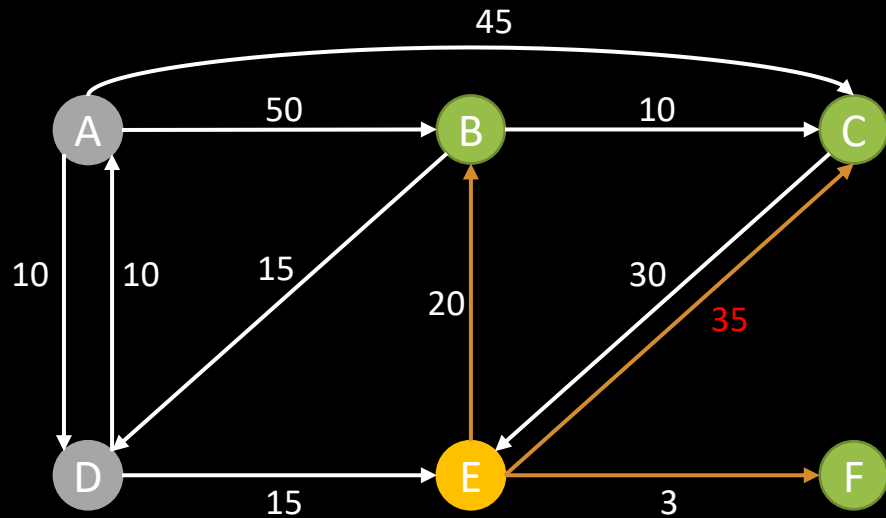
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | ∞ | |

Nodos visitados = [A,D]

Nodos por visitar = [B,C,E,F]

Algoritmo Dijkstra



$$d(A,C) = d(A,E) + d(E,C) = 25 + 35 = 60 < 45 \quad \text{✗}$$

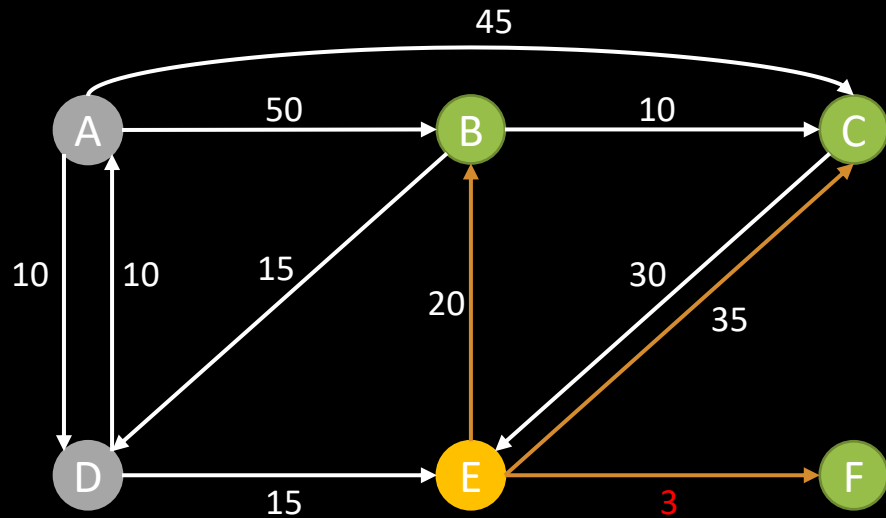
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | ∞ | |

Nodos visitados = [A,D]

Nodos por visitar = [B,C,E,F]

Algoritmo Dijkstra



$$d(A,F) = d(A,E) + d(E,F) = 25 + 3 = 28 < \infty \quad \checkmark$$

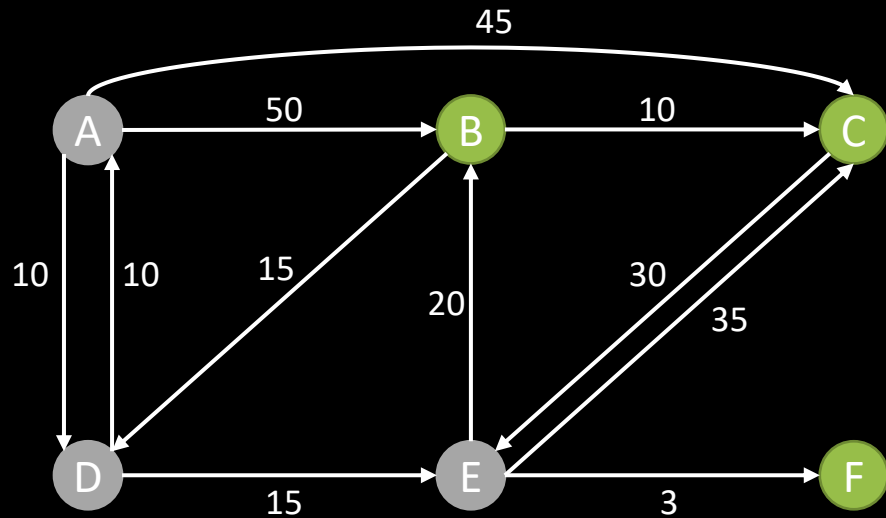
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D]

Nodos por visitar = [B,C,E,F]

Algoritmo Dijkstra



Para continuar vamos a seleccionar al nodo con la distancia más corta de los nodos por visitar.

Nodo F

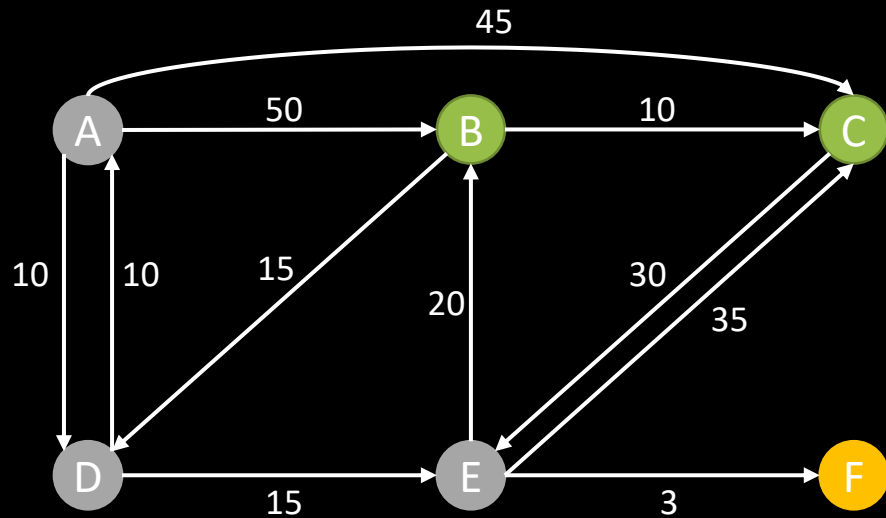
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------|-------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E]

Nodos por visitar = [B,C,F]

Algoritmo Dijkstra



Desde el nodo F no se puede llegar a ningún otro nodo.

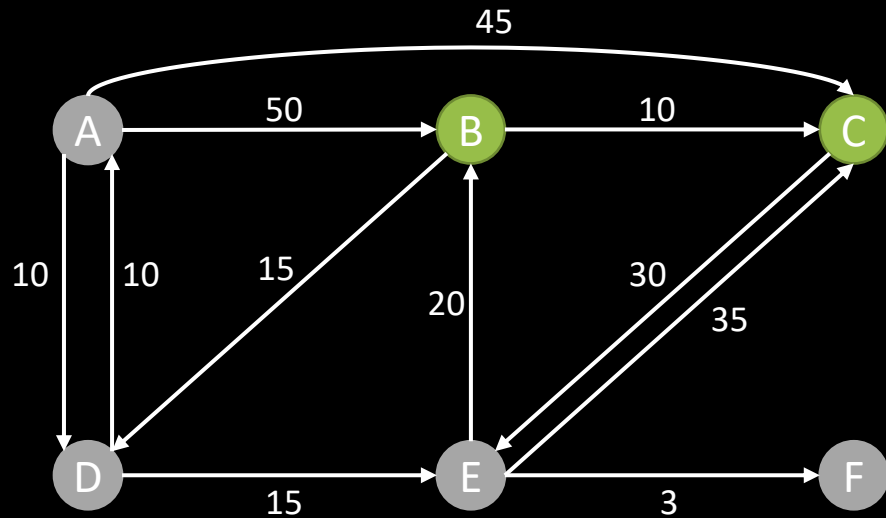
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E]

Nodos por visitar = [B,C,F]

Algoritmo Dijkstra



Para continuar vamos a seleccionar al nodo con la distancia más corta de los nodos por visitar.

B y C tienen la misma distancia por lo que se puede continuar con cualquiera de los dos.

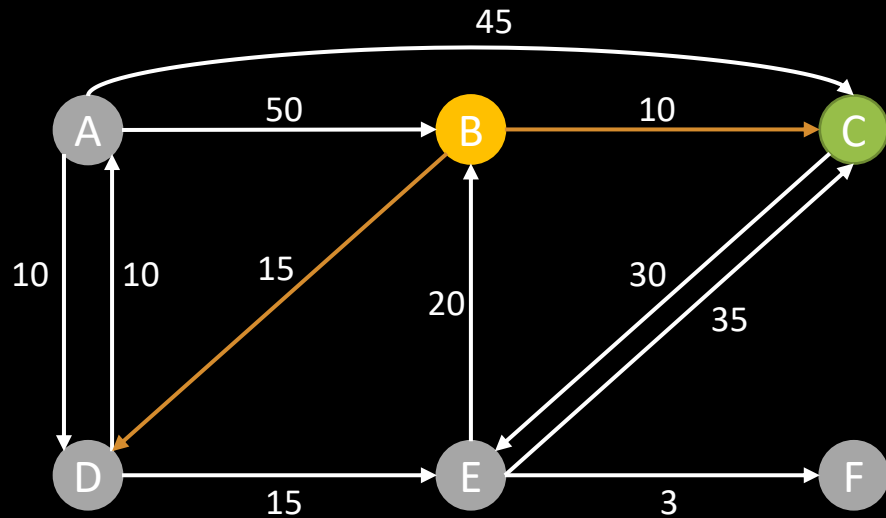
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------|-------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F]

Nodos por visitar = [B,C]

Algoritmo Dijkstra



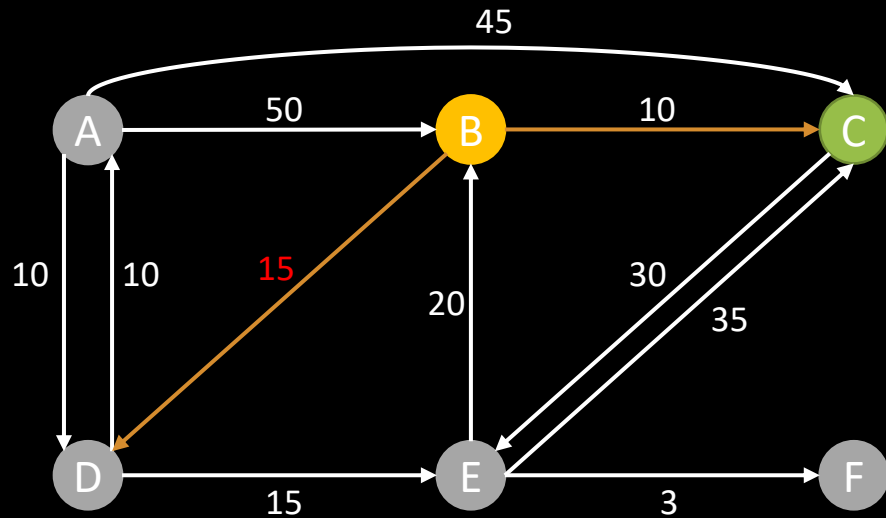
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F]

Nodos por visitar = [B,C]

Algoritmo Dijkstra



Esta arista se ignora porque va hacia un nodo ya visitado.

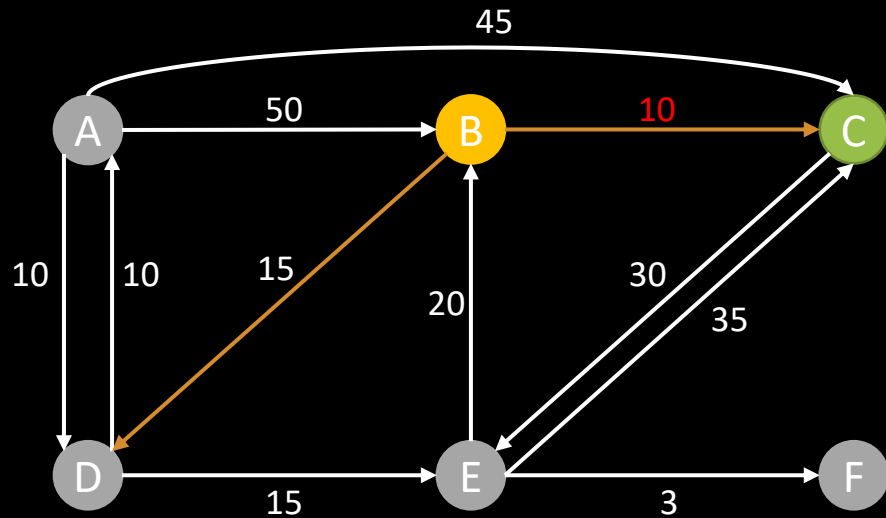
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F]

Nodos por visitar = [B,C]

Algoritmo Dijkstra



$$d(A,C) = d(A,B) + d(B,C) = 45 + 10 = 55 < 45 \quad \text{✗}$$

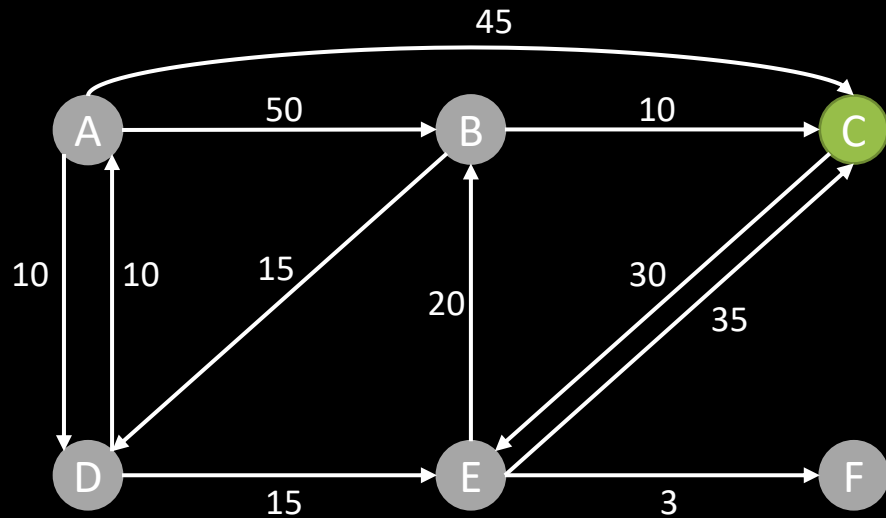
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F]

Nodos por visitar = [B,C]

Algoritmo Dijkstra



Para continuar vamos a seleccionar al nodo con la distancia más corta de los nodos por visitar.

El único nodo que queda es C

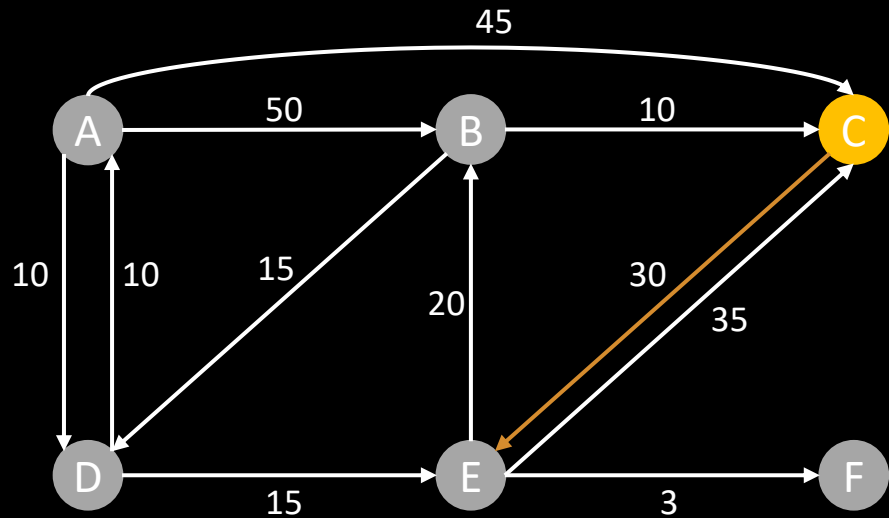
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B]

Nodos por visitar = [C]

Algoritmo Dijkstra



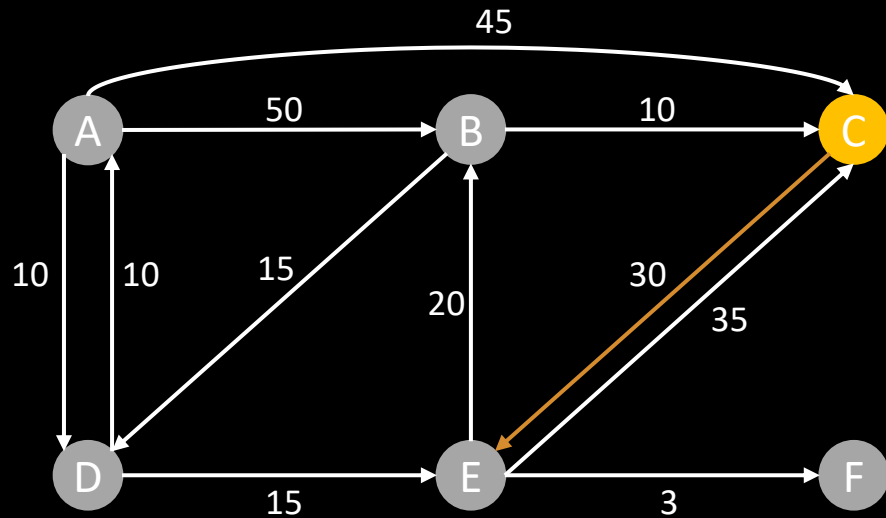
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B]

Nodos por visitar = [C]

Algoritmo Dijkstra



La única arista que sale de C se dirige hacia un nodo ya visitado, por lo tanto se ignora.

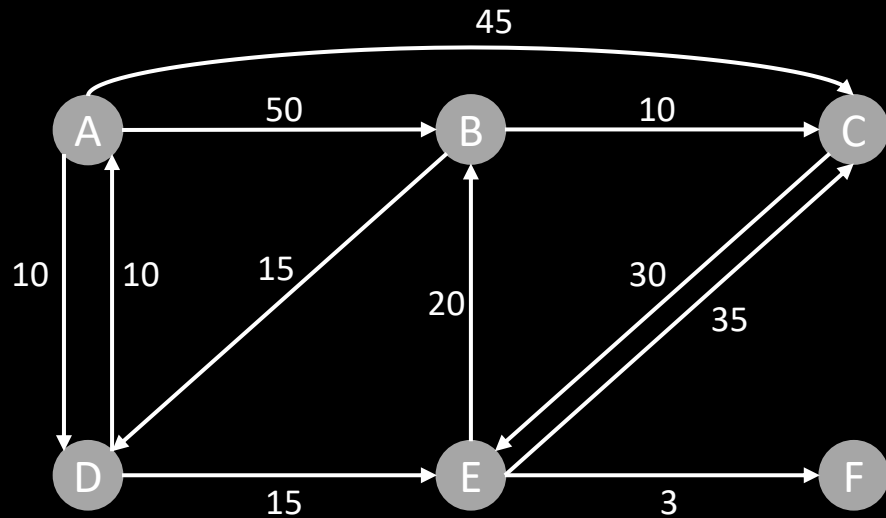
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B]

Nodos por visitar = [C]

Algoritmo Dijkstra



Ya hemos calculado la distancia más corta desde el nodo A hacia el resto de los nodos del grafo.

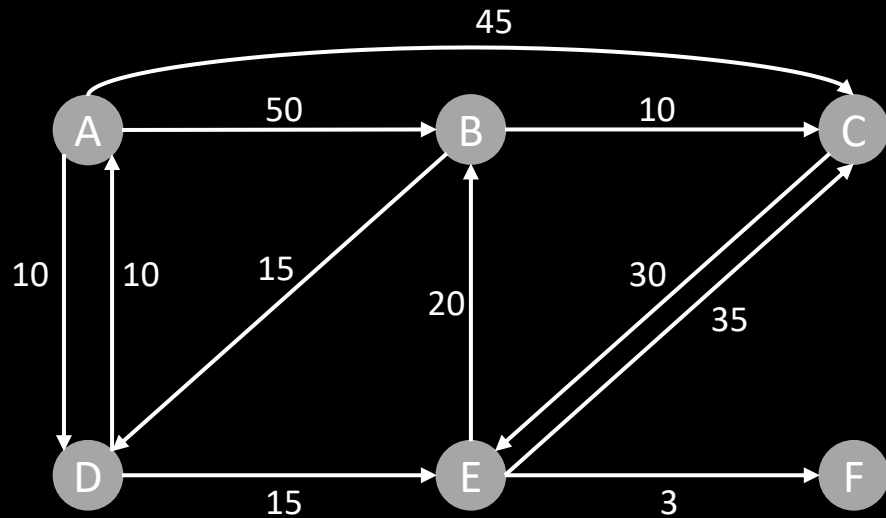
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B,C]

Nodos por visitar = []

Algoritmo Dijkstra



Y si ahora quiero saber el camino más corto desde A hasta F?

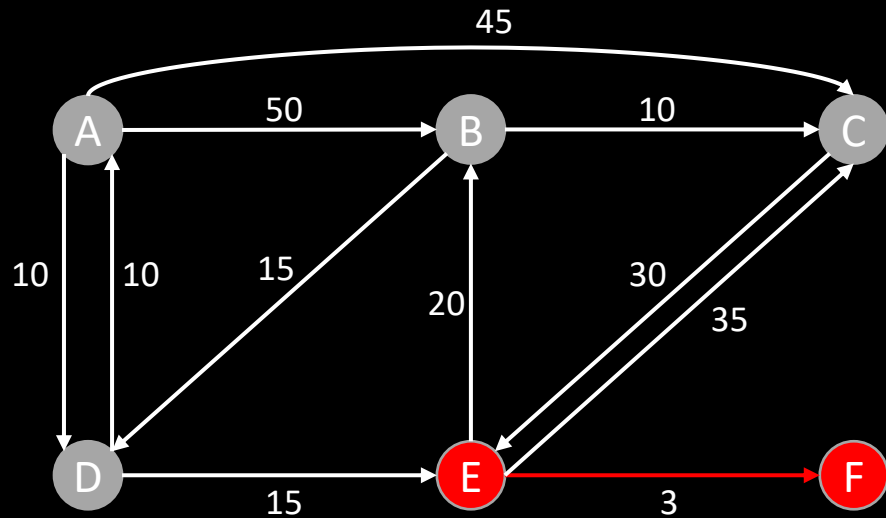
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B,C]

Nodos por visitar = []

Algoritmo Dijkstra



Y si ahora quiero saber el camino más corto desde A hasta F?

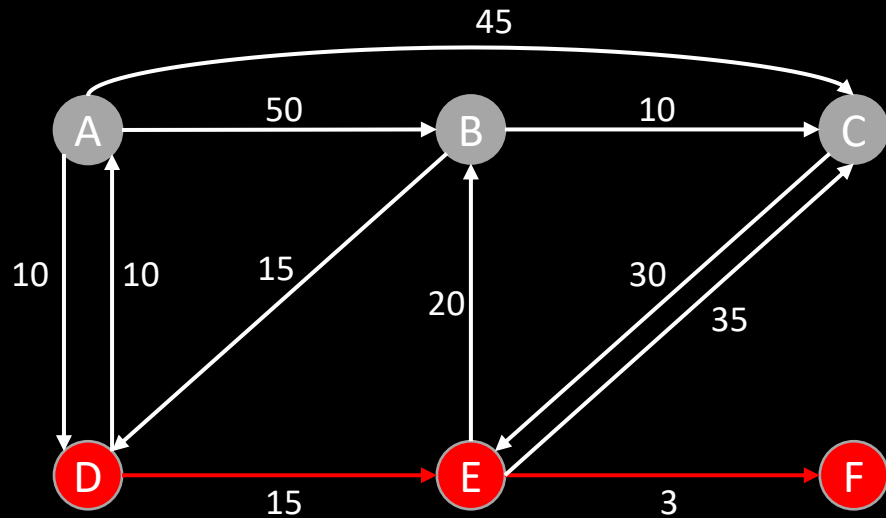
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B,C]

Nodos por visitar = []

Algoritmo Dijkstra



Y si ahora quiero saber el camino más corto desde A hasta F?

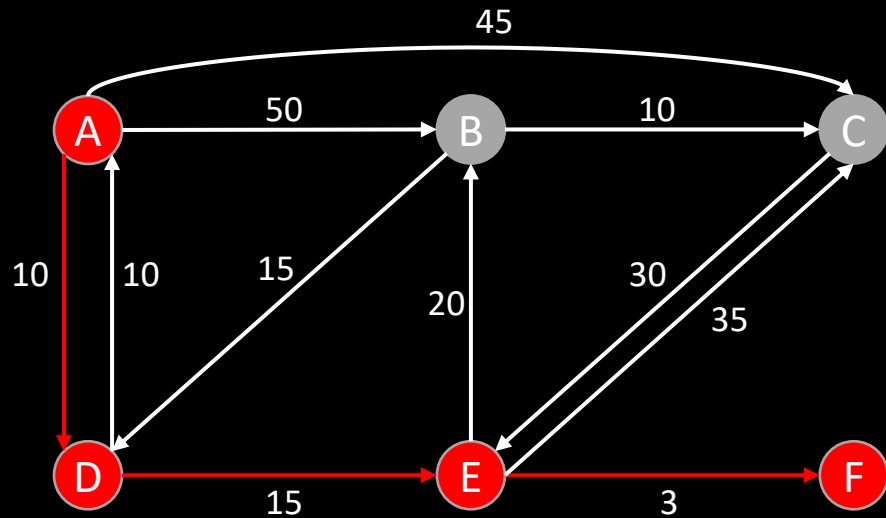
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B,C]

Nodos por visitar = []

Algoritmo Dijkstra



Y si ahora quiero saber el camino más corto desde A hasta F?

Camino: $A \rightarrow D \rightarrow E \rightarrow F$
Costo = 28

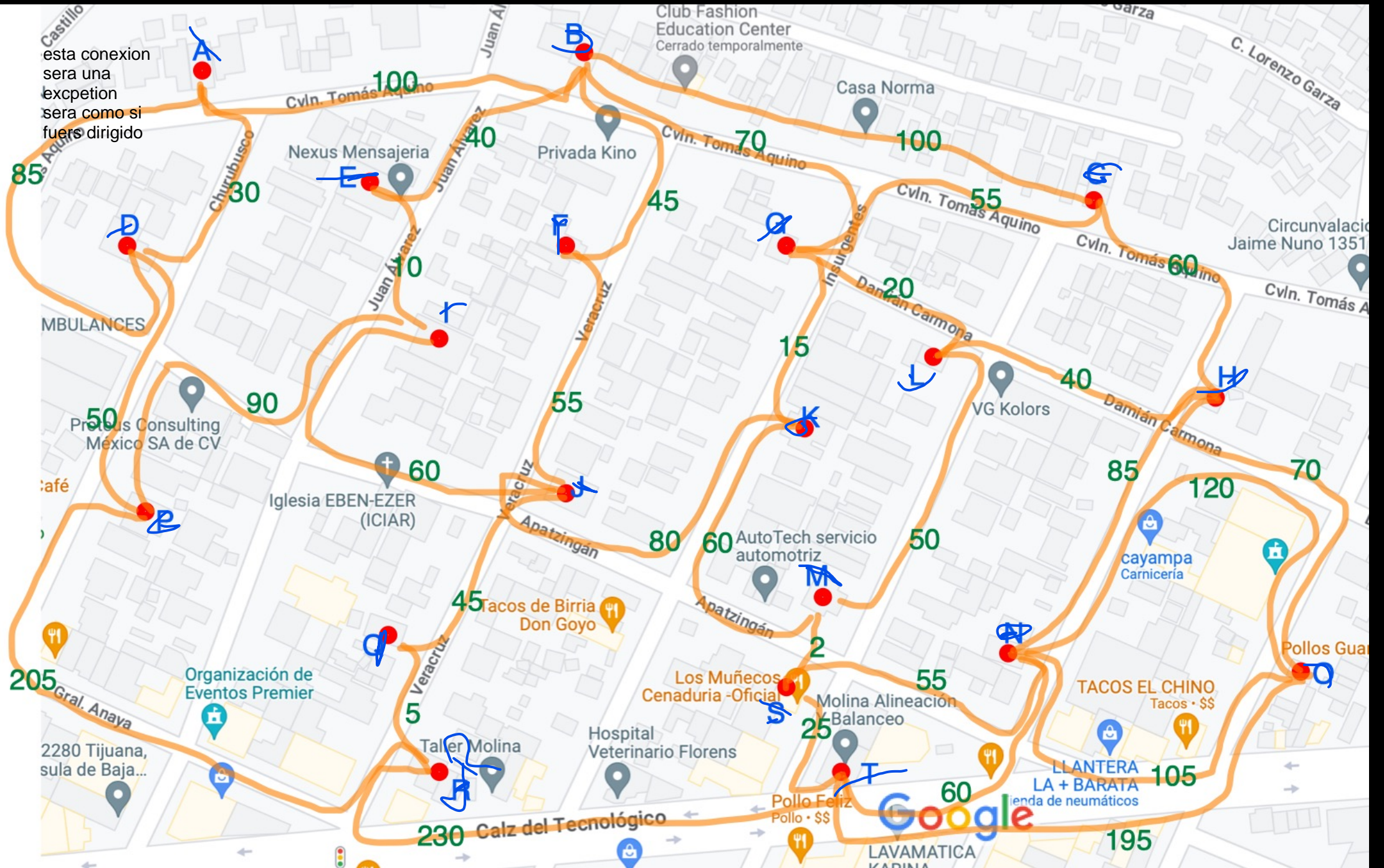
Salimos del nodo A

| | Distancia más corta | Nodo previo |
|---|---------------------------|----------------|
| A | 0 | |
| B | 45 | E |
| C | 45 | A |
| D | 10 | A |
| E | 25 | D |
| F | 28 | E |

Nodos visitados = [A,D,E,F,B,C]

Nodos por visitar = []

esta conexion
sera una
excpetion
sera como si
fuers dirigido



Problemas

- Un cartero necesita entregar cartas a 20 casas.Cuál es el camino que debe seguir para recorrer todas las casas haciendo la menor distancia posible?
- Pablo vive en la casa A y quiere ir a visitar a Carlos que vive en la casa O.Cuál es el camino que debe seguir para recorrer la menor distancia posible?