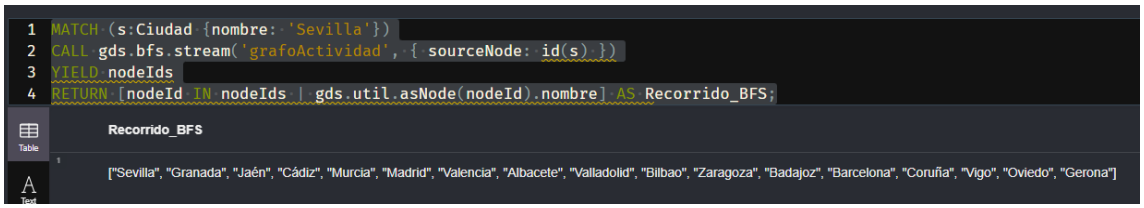
Actividad 05. Recorrido de grafos

Recorridos y caminos mínimos

(Sobre el grafo de la actividad anterior)

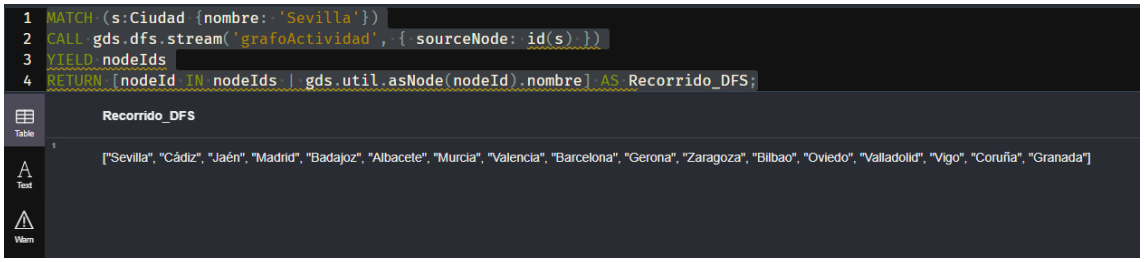
- Recorrido BFS desde Sevilla

```
MATCH (s:Ciudad {nombre: 'Sevilla'})
CALL gds.bfs.stream('grafoActividad', { sourceNode: id(s) })
YIELD nodeIds
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS Recorrido_BFS;
```



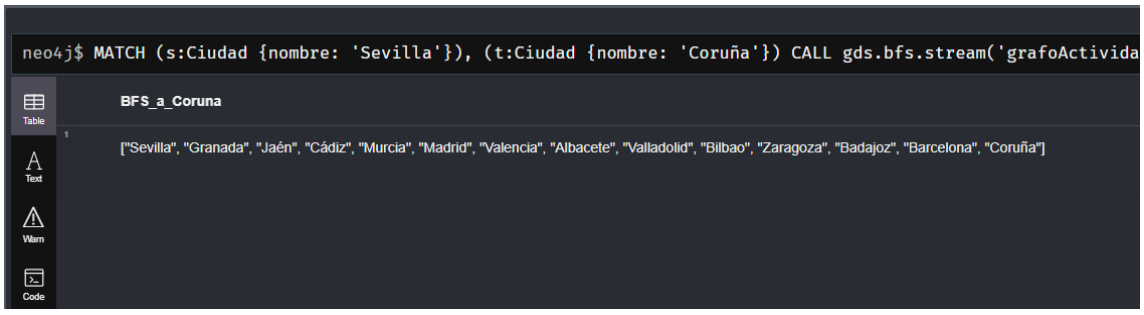
- Recorrido DFS desde Sevilla

```
MATCH (s:Ciudad {nombre: 'Sevilla'})
CALL gds.dfs.stream('grafoActividad', { sourceNode: id(s) })
YIELD nodeIds
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS Recorrido_DFS;
```



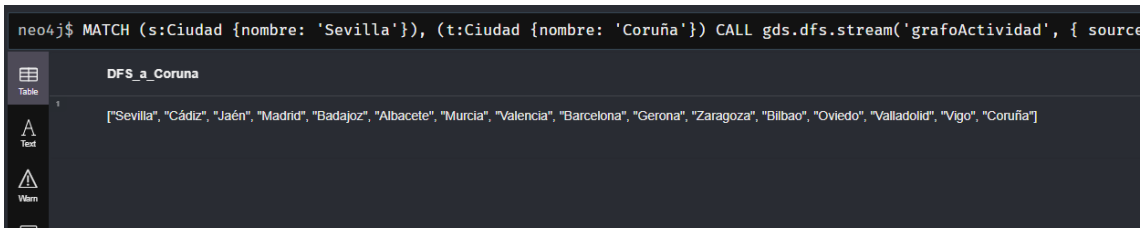
- Recorrido BFS especificando nodos destino Coruña

```
MATCH (s:Ciudad {nombre: 'Sevilla'}), (t:Ciudad {nombre: 'Coruña'})
CALL gds.bfs.stream('grafoActividad', { sourceNode: id(s), targetNodes: [id(t)] })
YIELD nodeIds
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS BFS_a_Coruna;
```



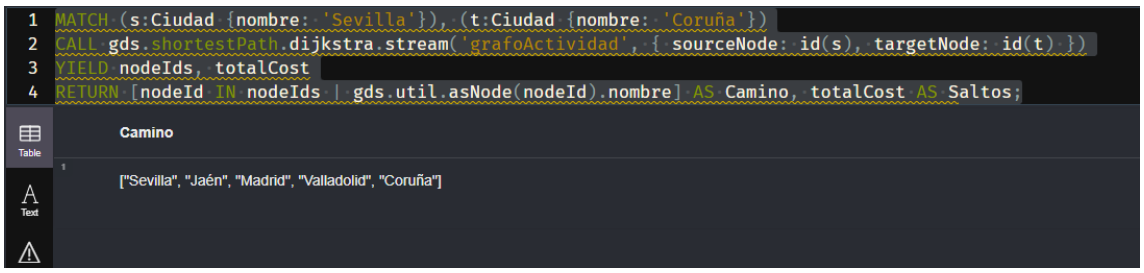
- Recorrido DFS especificando nodos destino Coruña

```
MATCH (s:Ciudad {nombre: 'Sevilla'}), (t:Ciudad {nombre: 'Coruña'})
CALL gds.dfs.stream('grafoActividad', { sourceNode: id(s), targetNodes: [id(t)] })
YIELD nodeIds
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS DFS_a_Coruna;
```



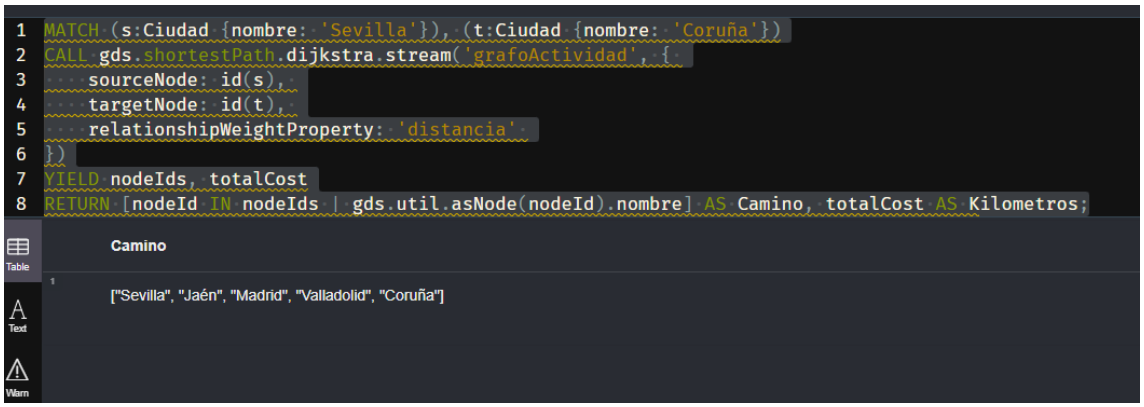
- Camino mínimo (Dijkstra) entre Sevilla y Coruña (no pesado)

```
MATCH (s:Ciudad {nombre: 'Sevilla'}), (t:Ciudad {nombre: 'Coruña'})
CALL gds.shortestPath.dijkstra.stream('grafoActividad', { sourceNode: id(s), targetNod
e: id(t) })
YIELD nodeIds, totalCost
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS Camino, totalCost
AS Saltos;
```



- Camino mínimo (Dijkstra) entre Sevilla y Coruña (pesado)

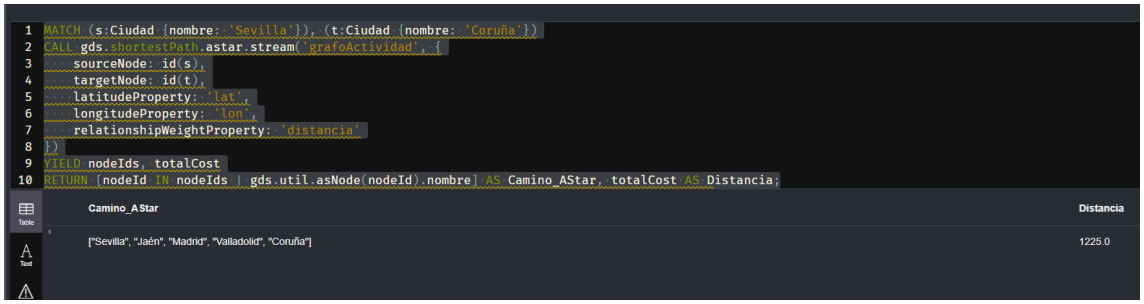
```
MATCH (s:Ciudad {nombre: 'Sevilla'}), (t:Ciudad {nombre: 'Coruña'})
CALL gds.shortestPath.dijkstra.stream('grafoActividad', {
    sourceNode: id(s),
    targetNode: id(t),
    relationshipWeightProperty: 'distancia'
})
YIELD nodeIds, totalCost
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS Camino, totalCost
AS Kilometros;
```



- Camino mínimo (A*) entre Sevilla y Coruña. Comenta el resultado. ¿Qué tendrías que modificar en el grafo actual para obtener un resultado diferente?. Verifícalo.

```
MATCH (s:Ciudad {nombre: 'Sevilla'}), (t:Ciudad {nombre: 'Coruña'})
CALL gds.shortestPath.astar.stream('grafoActividad', {
    sourceNode: id(s),
    targetNode: id(t),
```

```
    latitudeProperty: 'lat',
    longitudeProperty: 'lon',
    relationshipWeightProperty: 'distancia'
})
YIELD nodeIds, totalCost
RETURN [nodeId IN nodeIds | gds.util.asNode(nodeId).nombre] AS Camino_AStar, tot
alCost AS Distancia;
```



Para obtener un resultado diferente, tendrías que modificar la distancia de las aristas del camino actual. Por ejemplo, si aumentas drásticamente la distancia del tramo Madrid-Valladolid, el algoritmo descartará esa ruta y buscará una alternativa con menor coste total. En el caso de A*, también podrías alterar las coordenadas (`lat`/`lon`) para "engañar" a la heurística y cambiar la prioridad de exploración.