ALVAREZ Julián 14173 23/06/23

```
# define t_graph_elem char*
# define t_matrix_elem int
(2) # define t_queue_elem char*

typedef struct {
    int Min;
    queue* Route;
} MAP;

MAP* get_PATH (graph* g, char* origen, char* destino, matrix* distancias
               matrix* precedencias)
{
  if (!g) return;

  Map* Trip = (Map*) malloc (sizeof(MAP));
int k=0;
int i = graph_vertex_index (g, origen, STRCMP);
int j = graph_vertex_index (g, destino, STRCMP);
 Trip -> Min = matrix_get (distancias, i, j);
 Trip -> Route = queue_new ();
 enqueue (Trip-> route, graph_vertex_get (g, i));
 while ( j != matrix_get (precedencias, i, j))
 {
    k = matrix_get ( precedencias, i, j);
    enqueue (Trip -> route, graph_vertex_get (g,k));
    i=k;
 }
 enqueue (Trip-> route, graph_vertex_get (g,j));
 return Trip;
}
```

Bien definida la estructura

Reserva memoria

Levanta perfectamente los índices

Mete orígen en la queue

Esta está de mas