

# Pseudocódigo del Algoritmo

```
indice := integer = x;
conjunto := lista de enteros = rellena;
M := matrizAdyacencia = rellena;
camino, camino_siguiete, aux := lista de enteros = vacia;
indice_siguiete := integer;
min, min_aux := pareja de un double y una lista de enteros;

function distanciaConjunto (indice, conjunto) is
    aux := conjunto

    pair<double, list<int>> min, min_aux;
    camino.push_back(indice);

    if conjunto.size == 0 then
        hacer pareja M[indice-1][0] y camino
    else
        camino_siguiete := camino
        indice_siguiete := aux[0]
        remover aux[0]
        resultadoHijo := distanciaConjunto(indice_siguiete, aux)
        transferir al final de camino_siguiete resultadoHijo[1]
        min := pareja de M[indice-1][indice_siguiete-1] + resultadoHijo[0] y camino_siguiete)
        aux[0] insertar indice_siguiete

    for i in cada elemento de aux despues del primero do begin
        indice_siguiete := aux[i]
        borrar aux[i]

        camino_aux := camino

        hacer pareja resultadoHijo := distanciaConjunto(indice_siguiete, aux)
        transferir al final de camino_siguiete resultadoHijo[1]
        min_aux := pareja de M[indice-1][indice_siguiete-1] + resultadoHijo[0] y camino_siguiete)
        aux[i] insertar indice_siguiete

        if min_aux.first < min.first then
            min := min_aux;
        end
    end
    devolver min
end
end
```