HOJA Nº

FECHA

```
public class Parcial {

  public ListaGenerica<ListaGenerica<String>> resolver (Grafo<String> ciudades, String origen,
                                                         String destino, String pasandoPor){
                                       resultado
    ListaGenerica <ListaGenerica<String>> = new ListaGenericaEnlazada<ListaGenericaEnlazada<String>> ();
    IF ((ciudades != null) && (! ciudades.esVacio())){
      Vertice<String> vIni = null;
      Vertice<String> vFin = null;
      Vertice<String> vCond = null;
      boolean [] visitados = new boolean [ciudades.listaDeVertices().tamanio()];
      ListaGenerica <String> caminoAct = new ListaGenericaEnlazada <String> ();
      ListaGenerica <String> vertices = ciudades.listaDeVertices ();
      vertices.comenzar ();
      while (!vertices.fin()){
        Vertice<String> vAux = vertice.proximo ();
        IF (vAux.dato().equals (origen)
          vIni = vAux;
        IF (vAux.dato().equals (destino)
          vFin = vAux;
        IF (vAux.dato().equals (pasandoPor)
          vCond = vAux;
      }
      IF ((vIni != null) && (vDestino != null) && (vCond != null))
        DFS (vIni, vFin, vCond, caminoAct, visitados, resultado);
    }
  return resultado;

  private void DFS( Vertice<String> vIni, Vertice<String> vFin, Vertice<String> vCond, ListaGenerica<String> camA,
                    boolean[] visitados, ListaGen<ListaGen<String>> resultado){

    camA.agregarAlFinal (vIni);
    visitados[vIni.posicion]= True;
    IF (vIni == vFin) && ( camA.incluye (vCond)) {
      ListaGenerica <String> lis = new ListaGenericaEnlazada <String> ();
      copiarLista ( lis, camA);
      AgregarAlFinal (lis);
    }
    ListaGenerica <String> adyacentes = ciudades.listaDeAdyacentes (vIni);
    adyacentes.comenzar();
    while (!adyacentes.fin())
      Arista<String> arista = adyacentes.proximo ();
      Vertice<String> vAux = arista.verticeDestino();
      IF (!visitados[vAux.posicion()] && (arista.peso() == True))
        DFS (vAux, vFin, vCond, camA, visitados, resultado);
    }

}   visitados [vIni.posicion()] = False;
    camA.eliminarEn ( camA.tamano () - 1);
```

NOTA // falta hacer método de copiar lista