

```
public ListaGenerica<Ciudad> resolver (Grafo<Ciudad> ciudades, String origen,
                                         String destino) {
```

```
    ListaGenerica<Ciudad> resultado = new ListaGenericaEnlazada<Ciudad> ();
```

```
    if (ciudades != null && !ciudades.esVacio()) {
```

```
        - Verice<Ciudad> vIni = null;
```

```
        - Verice<Ciudad> vFin = null;
```

```
        - boolean[] visitados = new boolean[ciudades.listaDeVerices().tamaño()];
```

```
        - ListaGenerica<Ciudad> caminoAct = new ListaGenericaEnlazada<Ciudad> ();
```

```
        - ListaGenerica<Ciudad> verices = ciudades.listaDeVerices();
```

```
        - verices.comenzar();
```

```
        - while (!verices.fin()) {
```

```
            - Verice<Ciudad> vAux = verices.proximo();
```

```
            - if (vAux.dato() == origen)
```

```
                - vIni = vAux;
```

```
            - if (vAux.dato() == destino)
```

```
                - vFin = vAux;
```

```
        - if ((vIni != null) && (vFin != null)) {
```

```
            - dfs(vIni, vFin, visitados, caminoAct, resultado);
```

```
        } }
```

```
    return resultado;
```

```
private void dfs (Verice<Ciudad> vIni, Verice<Ciudad> vFin, boolean[] visitados, ListaGenerica<Ciudad> caminoAct,
                  ListaGenerica<Ciudad> resultado) {
```

```
    caminoActual.agregarAlFinal(vIni.dato());
```

```
    visitados[vIni.posicion()];
```

```
    if (vIni == vFin)
```

```
        - copiarLista (caminoActual, resultado);
```

```
    else {
```

```
        - ListaGenerica<Arista<Ciudad>> adyacentes = ciudades.listaDeAdyacentes();
```

```
        - adyacentes.comenzar();
```

```
        - while (!adyacentes.fin()) {
```

```
            - Verice<Ciudad> vAux = adyacentes.proximo().getVericeDestino();
```

```
            - if (!visitados[vAux.posicion()] && (vAux.getFase() != 1))
```

```
                - dfs(vAux, vFin, visitados, caminoAct, resultado);
```

```
            }
```

```
        }
```

```
    caminoActual.eliminarEn (caminoActual.tamaño() - 1);
```

```
    visitados[vIni.posicion()] = false;
```

```
}
```

(*)

NOTA


```
public class Ciudad {
```

```
- private String nombre;
```

```
- private int fare;
```

```
public Ciudad {
```

```
}
```

```
public Ciudad (String nombre, int fare) {
```

```
- this.nombre = nombre;
```

```
- this.fare = fare;
```

```
}
```

```
public String getNombre () {
```

```
- return nombre;
```

```
}
```

```
public void setNombre (String nom) {
```

```
- this.nombre = nom;
```

```
}
```

```
public void getFare () {
```

```
- return fare;
```

```
}
```

```
public void setFare (int fare) {
```

```
- this.fare = fare;
```

```
}
```

```
}
```

```
* private void copiaLista (ListaGenerica<Ciudad> caminoA, ListaGenerica<Ciudad> resultado) {
```

```
while (!resultado.empty())
```

```
- resultado.eliminar(0);
```

```
caminoA.comenzar(0);
```

```
while (!caminoA.fin())
```

```
- resultado.agregarAlFinal (caminoA.proximo());
```

```
}
```

```
}
```