

```

public Partial 2 {
    public ListaGenerica<String> resolver (Grafo<Ciudad> ciudades,
        String origen, String destino, ListaGenerica<String> pasadoPor) {
        boolean[] marca = new boolean[ciudades.listaDeVertices().tamaño()];
        ListaGenerica<String> camino = new ListaGenerica<String>();
        ListaGenerica<Vertice<Ciudad>> aux = ciudades.listaDeVertices();
        boolean encuentre = false;
        aux.comenzar();
        Vertice<Ciudad> v = null;
        while (!aux.fin() && !encuentre) {
            v = aux.proximo();
            if (v.datallnombre().equals(origen)) {
                encuentre = true;
            }
        }
        if (encuentre) {
            encuentre = resolver(ciudades, destino, pasadoPor, marca, camino,
                                v.posicion());
        }
        if (!encuentre) {
            camino = new ListaGenerica<String>();
        }
        return camino;
    }
}

```



```

private boolean resolver (Grafo<Ciudad> ciudades, String destino,
    Lista Generica<String> pasadoPor, boolean[] marca, Lista Generica<String> caminos,
    int i) {
    boolean encuentre = false;
    marca[i] = true;
    Vertice<Ciudad> v = ciudades.listaDeVertices().elemento(i);
    caminos.agregarFinal(v.data().nombre());
    if (v.data().nombre().equals(destino)) {
        encuentre = true;
        pasadoPor.comenzar();
        while (!pasadoPor.fin() && encuentre) {
            encuentre = caminos.incluye(pasadoPor.proximo());
        }
    }
    if (!encuentre) {
        Lista Generica<Arista<Ciudad>> aristas = ciudades.listaDeAristas();
        aristas.comenzar();
        while (!aristas.fin() && !encuentre) {
            Arista<Ciudad> arista = aristas.proximo();
            int j = arista.verticeDestino().posicion();
            if (!marca[j]) {
                encuentre = resolver(ciudades, destino, pasadoPor,
                    marca, caminos, j);
            }
            if (!encuentre) {
                marca[j] = false;
                caminos.eliminarFinal(caminos.tamano()-1);
            }
        }
    }
    return encuentre;
}

```