

```
Public CLASS PARCIAL {
```

```
Public CLASS CIUDAD {
```

```
Private String Nombre;
```

```
Private int Fase;
```

```
Public CIUDAD (String N, int F){
```

```
    This.Nombre = N;
```

```
    This.Fase = F;
```

```
}
```

```
Public String getNombre(){
```

```
    Return This.Nombre;
```

```
}
```

```
Public int getFase()
```

```
    Return This.Fase;
```

```
}
```

```
} // FIN OBJETO
```

```
Public LISTA Generica <String> resolver (Grafo <CIUDAD> CIUDADES, String Origen,  
String Destino) {
```

```
    LISTA generica <String> CAMINO = new LISTA enlazada generica();
```

```
    IF (CIUDADES.esVacio()) {
```

```
        int POS = BUSCAR (CIUDADES.listaDeVertices(), Origen, Destino);
```

```
        IF (POS != -1) {
```

```
            Boolean[] MARCA = new Boolean [CIUDADES.listaDeVertices().TAMANO];
```

```
            Ejecutar CIUDADES (CIUDADES.listaDeVertices(), MARCA)
```

```
            DFS (POS, CIUDADES, Destino, CAMINO, MARCA);
```

```
        }
```

```
    }
```

```
    Return CAMINO;
```

```
Private BUSCAR (LISTA generica <VERTICES <CIUDAD>> L, String Origen, String  
Destino) {
```

```
    int POS = -1, DEST = -1;
```

```
    L.comenzar();
```

```
    while ((POS == -1 || DEST == -1) && (!L.FIN())) {
```

```
        Vertice <CIUDAD> V = L.Proximo();
```

```
        String NAME = V.DATO().getNombre();
```

```
        IF ((POS == -1) && (NAME.equals(Origen))) {
```

```
            POS = V.Posicion();
```

```
        } else IF (DEST == -1) && (NAME.equals(Destino)) {
```

```
            DEST = V.Posicion();
```

```
        }
```

```
    } IF (DEST == -1) { POS = -1;
```

```
    Return POS;
```

```
}
```

```

PRIVATE VOID EXCEPTAR CIUDADES (LISTA_GENERICA<VERTICES<CIUDAD>> L, BOOLEAN[]
MARCA) {

```

```

    L.COMENZAR();
    WHILE (!L.FIN()) {
        VERTICE<CIUDAD> V = L.PROXIMO();
        IF (V.DATO().GETFASE() == 1) {
            MARCA[V.POSICION] = TRUE;
        }
    }
}

```

```

PRIVATE BOOLEAN DFS (INT i, GRAFO<CIUDAD> g, STRING DESTINO, LISTA_GENERICA<STRING>
CAMINO, BOOLEAN[] MARCA) {
    MARCA[i] = TRUE;
    VERTICE<CIUDAD> V = g.VERTECE(i);
    CAMINO.AGREGAR_FINAL(V.DATO().GETNOMBRE());
    BOOLEAN encuentre = V.DATO().GETNOMBRE().EQUALS(DESTINO);
    IF (!encuentre) {
        LISTA_GENERICA<ARISTAS<CIUDAD>> ADY = V.LISTADEADYACENTES();
        ADY.COMENZAR();
        WHILE (!encuentre) && (!ADY.FIN()) {
            -- ARISTA<CIUDAD> A = ADY.PROXIMO();
            -- INT j = A.VERTECE_DESTINO().POSICION;
            IF (!MARCA[j]) {
                -- ENCONTRE = DFS(j, g, DESTINO, CAMINO, MARCA);
            }
        }
    }
    IF (!encuentre) {
        CAMINO.ELIMINAR(CAMINO.TAMANIO-1);
    }
    RETURN ENCONTRE;
}
}

```