```
Public CLASS PARCial {

--Public LISTAGeneRICA<STRING> CAMINODISTANCIAMAXIMA (GRAFO<STRING> CIUDADES,
      STRING ORIGEN, STRING DESTINO, INT DISTANCIAMAXIMA){
  LISTAgeneRICA<STRING> CAMINO = NEW LISTAENLAZADAGeneRICA();
  IF(!CIUDADES.esVACIO()){
     INT POS = BUSCAR (CIUDADES.LISTAde VERTICES(), ORIGEN, DESTINO);
     IF(POS != -1){
        BOOLEAN[] MARCA = NEW BOOLEAN[CIUDADES.LISTAdeVertices().TAMAÑO
                                      +1]();
        LISTAGeneRICA<STRING> CAMACD = NEW LISTAENLAZADA genera();
        DFS(POS, CIUDADES, DESTINO, DISTANCIAMAXIMA, CAMINO, CAMACT, MARCA);
     }
  }
  RETURN CAMINO;

PRIVATE INT BUSCAR (LISTAGENERICA<STRING> LISTA, STRING ORIGEN, STRING DESTINO){
  INT POS=-1, DES=-1;
  LISTA.COMENZAR();
  WHILE(!LISTA.FIN){
     VERTICE<STRING> V = LISTA.PROXIMO();
     IF((POS==-1) && (V.DATO().EQUALS(ORIGEN))){
        POS = V.POSICIÓN();
     }ELSE IF((DES==-1) && (V.DATO().EQUALS(DESTINO))){
        DES = V.POSICIÓN();
     }
  }
  IF(DES==-1){ POS=-1;}
  RETURN POS;
```

```
Private Boolean DFS(int i, grafo<String> g, String destino, int dmax,
                    , ListaGenerica<String> camAct, Boolean[] marca);
marca[i] = true;
Vertice<String> v = g.vertice(i);
camAct.agregarFinal(v.dato());
Boolean cumple = v.dato().equals(destino);

IF (!cumple){
  ListaGenerica<Arista<String>> ady = v.listaDeAdyacentes()
  ady.comenzar();
  while (!ady.fin()){
     Arista<String> a = ady.proximo();
     IF (a.peso() <= dmax){
        int j = a.verticeDestino().posicion();
        IF (marca[j]){
           cumple = DFS(j, g, dmax, camAct, marca)
        }
     }
  }
}
IF (!cumple){
   camAct.eliminarEn(camAct.tamanio());
}
return cumple;
```