

~~DATA y los datos / Nombres / Pasados / Argumentos~~
~~Te Devela la cantidad de ciudades donde aparece un nodo~~
~~Sumados~~

Public class Parcial {

```
-- Public. ListaGenerica<ListaGenerica<String>> Resolver (Grafo<String> Ciudades,
String Origen, String Destino, String Pasando Por) {
    ListaGenerica<ListaGenerica<String>> Caminos = new ListaEnlazadaGenerica();
    IF (!Ciudades.esVacio()) {
        int pos = Buscar (Ciudades, ListaDeVertices(), Origen, Destino);
        IF (pos != -1) {
            Boolean[] Marca = new Boolean [Ciudades.ListaDeVertices().Tamanio()-1];
            ListaGenerica<String> CaminoAct = new ListaEnlazadaGenerica();
            DFS (pos, Caminos, Destino, Marca, Ciudades, CaminoAct, Falso);
        }
    }
}
```

Return Caminos;

```
Private int Buscar (ListaGenerica<String> L, (String Origen, String Destino) {
    int pos = -1; Boolean OK = False; Boolean OK-Des = False; int pos-0 = 0;
    L.comenzar();
    While ((!OK || !OK-Des) && (!L.Fin())) {
        -- Verace<String> v = L.Proximo();
        -- IF ( v.Dato().Equals(Origen) ) {
            -- pos-0 = v.Posicion();
            -- OK-or = True;
        } else IF ( v.Dato().Equals(Destino) ) {
            -- OK-Des = True;
        }
    }
    IF (OK || OK-Des) {
        pos = pos-0;
    }
    Return pos;
}
```



```

PRIVATE VOID DFS (INT POS, LISTA_GENERICA<LISTA_GENERICA<STRING>> CAMINOS, STRING DESTINO,
    BOOLEAN[] MARCA, GRAFO<STRING> G, LISTA_GENERICA<STRING> CAM_ACTO, Boolean OK,
    STRING PASAR POR) {
    - MARCA[POS] = TRUE;
    VERTEX<STRING> V = G.VERTECE (POS);
    - CAM_ACTO.AGREGAR_FINAL (V.DATO());
    IF ((V.DATO().EQUALS (DESTINO)) && (OK)) {
        CAMINOS.AGREGAR_FINAL (CAM_ACTO.CLONAR());
    }
    } ELSE IF (V.DATO().EQUALS (PASAR POR)) {
        OK = TRUE;
    }
    } ELSE {
        LISTA_GENERICA<ARISTA<STRING>> ADY = V.LISTA_DE_ADYACENTES ();
        ADY.COMENZAR ();
        WHILE (!ADY.FIN()) {
            ARISTA<STRING> A = ADY.PROXIMO ();
            IF (A.PESO().EQUALS ("H")) {
                INT J = A.VERTECE_DESTINO().POSICION ();
                IF (!MARCA[J]) {
                    DFS (J, CAMINOS, DESTINO, MARCA, G, CAM_ACTO, OK, PASAR POR);
                }
            }
        }
    }
}

MARCA[POS] = FALSE;
CAM_ACTO.ELIMINAR_EN (CAM_ACTO.TAMANIO ());
}
}

```