

Un Camino Excepcional En La Ciudad

```
Public class Paredal () {
```

```
    Public class ciudad () {
```

```
        Private String nombre;
```

```
        Private int Fase;
```

```
        public Ciudad (String unNombre, int unaFase) {
```

```
            this.nombre = unNombre;
```

```
            this.Fase = unaFase;
```

```
        }
```

```
        Public String getNombre () { return this.nombre; }
```

```
        Public int getFase () { return this.Fase; }
```

```
        Public String setNombre (String nombre) { this.nombre = nombre; }
```

```
        Public int setFase (int Fase) { this.Fase = Fase; }
```

```
    }
```

```
    Public <G<String> resolver (Grafo<ciudad> ciudades, String Origen, String Destino)
```

```
    { <G<String> camino = new ListaEnlazadaGenerica ();
```

```
      IF (!ciudades.esVacio ()) {
```

```
        int Pos = Buscar (ciudades, origen, destino);
```

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

```
            Boolean C] mrcos = new Boolean [ciudades.listaDeVertices (). tamaño () + 1]
```

```
            dfs (Pos, ciudades, destino, camino, mrcos)
```

```
        }
```

```
    }
```

```
    return camino;
```

```
}
```

```
    Public int Buscar (Grafo<ciudad> g, String Origen, String Destino) {
```

```
        int PosOrigen = -1, PosDestino = -1;
```

```
        ListaGenerica<Vertice<ciudad>> vertices = g.listaDeVertices ();
```

```
        Vertices.convergir ();
```

```
        While ((PosOrigen != -1) || (PosDestino != -1)) && (!vertices.Fin ()) {
```

```
            Vertice<ciudad> vActual = vertices.Proximo ();
```

```
            IF (PosOrigen != -1) && (vActual.dato ().getNombre ().equals (origen))
```

```
                PosOrigen = vActual.Posicion ();
```

```
            }
```

```
            else IF (PosDestino != -1) && (vActual.dato ().getNombre ().equals (destino)) {
```

```
                PosDestino = vActual.Posicion ();
```

```
            }
```

```
        }
```

```
        IF (PosOrigen != -1) || (PosDestino != -1) {
```

```
            PosOrigen = -1
```

```
        }
```

```
        return PosOrigen;
```

```
}
```



```

public boolean dfs (int Pos, Grafo<Ciudad> g, String Destino, LGesmapa
boolean C] m
boolean Llegue = false
marco [Pos] = true
verice <Ciudad> vActual = g.verice(Pos);
camino.agregarFinal(vActual.dato().getNombre());
if (vActual.dato().getNombre().equals(Destino)) {
    Llegue = true
}
else {
    LG < Arista<String>> adys = g.listaDeAristas(vActual);
    adys.comezar();
    while (!Llegue) && (!adys.Fin()) {
        Arista < Ciudad> arista = adys.Proximo();
        verice < Ciudad> vSiguiente = arista.vericeDestino();
        if (vSiguiente.dato().getNombre() != "-") {
            int j = vSiguiente.Posicion();
            if (!marco[j]) {
                Llegue = dfs(j, g, Destino, camino, marco)
            }
        }
    }
}
if (!Llegue) {
    camino.eliminarEn(camino.Tam()-1);
}
return Llegue;
}

```