```
Public class BuscadorDeCaminos (){

    Private Grafo<string> busqueda;

    Public ListaGenerica < ListaGenerica <String>> recorridosMasSeguros() {
        ListaGenerica < ListaGenerica <string>> caminos = new ListaEnlazadaGenerica
        if (! this.busqueda. esVacio()){
            int pos = buscar (this.busqueda. listaDeVertices(), "casa Gabriela", "casa ab
            if (pos != -1) {
                boolean [] marca = new boolean [this.busqueda. listaDeVertices(). tamanio()
                dFS (pos, caminos, new ListaEnlazadaGenerica(), marca, "casa abuelo"
            }
        }
        return caminos;
    }

    Private void dFS (int pos, LG<LG<string>> caminos, LG<string> camActual,
                      boolean [] marca, string destino) {

        marca [pos] = true;
        Vertice <String> vActual = this.busqueda. vertice(pos);
        camActual. agregarFinal(vActual. dato());
        if(vActual. dato(). equals(destino)){
            caminos. agregarFinal(camActual. clonar());
        }
        else {
            ListaGenerica <Arista <string>> adys = this. busqueda. listaDeAdyacentes(vAc
            adys. comenzar();
            while(!adys. Fin()){
                Arista <string> arista = adys. proximo();
                int j = arista. verticeDestino(). posicion();
                if(! marca [j]) && (arista. Peso() < 5){
                    dfs(j, caminos, camActual, marca, destino)
                }
            }
        }
        marca [pos] = false;
        camActual. eliminarEn(camActual. tamanio());
    }
```

```
Public class BuscadorDeCaminos () {

    Private Grafo<String> bosque;

    Public ListaGenerica<ListaGenerica<String>> recorridosMasSeguros() {
        ListaGenerica<ListaGenerica<String>> caminos = new ListaEnlazadaGenerica
        IF (!this.bosque.esVacio()){
            int pos = buscar(this.bosque.listaDeVertices(), "casa Caperucita", "casa ab
            IF (pos != -1) {
                boolean [] marca = new boolean [this.bosque.listaDeVertices().tamanio()
                dFS(pos, caminos, new ListaEnlazadaGenerica(), marca, "casa abuelita"
            }
        }
        return caminos;
    }

    Private void dFS (int pos, LG<LG<String>> caminos, LG<String> camActual,
                      boolean [] marca, String destino) {

        marca [pos] = true;
        Vertice<String> vActual = this.bosque.vertice(pos);
        camActual.agregarFinal(vActual.dato());
        iF(vActual.dato().equals(destino)){
            caminos.agregarFinal(camActual.clonar());
        }
        else {
            ListaGenerica<Arista<String>> adys = this.bosque.listaDeAdyacentes(vAc
            adys.comenzar();
            while(!adys.Fin()) {
                Arista<String> arista = adys.Proximo();
                int j = arista.verticeDestino().Posicion();
                if(!marca[j]) && (arista.Peso() < 5){
                    dfs(j, caminos, camActual, marca, destino)
                }
            }
        }
        marca[pos] = false;
        camActual.eliminarEn(camActual.tamanio());
    }
```