

Parcial 1 Grafos

Todo adentro de un
if (!ciudades.esVacio()) 4

HOJA N°

FECHA

```
Lista Generica<Ciudad> resolver (Grafo<Ciudad> ciudades, String  
    origen, String destino) {  
    Vertice<Ciudad> inicio = obtener Vertice(origen, ciudades);  
    Vertice<Ciudad> fin = obtener Vertice(destino, ciudades);  
    if (inicio == null || fin == null) return null;  
    int dim = ciudades.listaDe Vertices.tamano();  
    boolean[] marca = new boolean[dim];  
    for (boolean e : marca)  
        e = false;  
    Lista Generica<Ciudad> resultado = new Lista Generica Enlazada<Ciudad>();  
    resultado.comenzar();  
    resolver (ciudades, inicio, fin, marca, resultado);  
    return resultado;  
}
```

```
private Vertice<Ciudad> obtener Vertice(String dato, Grafo<Ciudades> grafo) {  
    Lista Generica<Vertice<Ciudad>> vertices = grafo.listaDe Vertices();  
    Vertice<Ciudades> aux;  
    vertices.comenzar();  
    while (!vertices.fin()) {  
        aux = vertices.proximo();  
        if (aux.dato().ciudad().equals(dato))  
            return aux;  
    }  
    return null;  
}
```



```

Private void resolver(Grafo<Ciudades> grafo, Vertice<Ciudad> actual,
    Vertice<Ciudad> destino, boolean[] marca, ListaGenerica<Ciudad> resultado) {
    int Pos = actual.Posicion();
    marca[Pos] = true;
    resultado.agregarFinal(actual.dato());
    if (actual.equals(destino))
        return;
    else if (actual.dato().fase() == 1) {
        resultado.eliminarEn(resultado.tamano() - 1);
        return;
    }
    else {
        ListaGenerica<Vertice<Ciudad>> ady = grafo.listaDeAdyacentes(actual);
        Vertice<Ciudad> aux;
        ady.comenzar();
        while (ady.fin()) {
            aux = ady.proximo();
            if (!marca[aux.Posicion])
                resolver(grafo, aux, destino, marca, resultado);
        }
    }
}
}

```

```

Public class Ciudad {
    Private String ciudad;
    Private int fase;
    Public String ciudad() {
        return this.ciudad;
    }
    Public int fase() {
        return this.fase;
    }
}

```