

```
public class Parcial {
```

```
    public int resolver( Grafo<Ciudad> ciudades, String origen, String destino, int maxControler) {
```

```
        int MaxTiempo = 0;
```

```
        if ((ciudades != null && (!ciudades.esVacio()))) {
```

```
            Vertice<Ciudad> vIni = null;
            Vertice<Ciudad> vFin = null;
            boolean[] visitados = new boolean[ciudades.listaDeVertices().tamaño()];
            Tiempo max = new Tiempo();
            int TiempoParcial = 0;
            ListoGenico<Ciudad> vertices = ciudades.listaDeVertices();
            vertices.comenzar();
            while (!vertices.fin()) {
                Vertice<Ciudad> vAux = vertices.proximo();
                if (vAux.dato().nombre().equals(origen))
                    vIni = vAux;
                if (vAux.dato().nombre().equals(destino))
                    vFin = vAux;
            }
            if (vIni != null & vFin != null) {
                dfs(vIni, vFin, visitados, max, ciudades, TiempoParcial, maxControler);
            }
        }
```

```
        MaxTiempo = max.getTiempo();
```

```
    return MaxTiempo;
```

```
    private void dfs(Vertice<Ciudad> vIni, Vertice<Ciudad> vFin, boolean[] visitados, Tiempo max, Grafo<Ciudad> ciudades) {
```

```
        int TiempoParcial = 0;
        TiempoParcial += vIni.dato().getTransito();
        visitados[vIni.posicion()] = true;
        if ((vIni == vFin) && (TiempoParcial > max.getTiempo())) {
            max.setTiempo(TiempoParcial);
        } else {
            ListoGenico<Arista<Ciudad>> adyacentes = ciudades.listaDeAdyacentes(vIni);
            adyacentes.comenzar();
            while (!adyacentes.fin()) {
                Arista<Ciudad> arista = adyacentes.proximo();
                int Peso = arista.peso();
                Vertice<Ciudad> vAux = arista.verteiceDestino();
                if (!visitados[vAux.posicion()] && (Peso < maxControler)) {
                    dfs(vAux, vFin, visitados, max, ciudades, TiempoParcial, maxControler);
                }
            }
            TiempoParcial -= vIni.dato().getTransito();
            visitados[vIni.posicion()] = false;
        }
    }
```

NOTA

// clase donde manejo la referencia

```
public class Tiempo {
```

```
    private int tiempo;
```

```
    public Tiempo() // constructor vacío
```

```
}
```

```
    public Tiempo(int time) { // constructor con parámetros
```

```
        this.tiempo = time;
```

```
}
```

```
    public void setTiempo(int tiempoNue) {
```

```
        this.tiempo = tiempoNue;
```

```
}
```

```
    public void getTiempo() {
```

```
        return tiempo;
```

```
}
```

```
}
```

// clase Ciudad (dato del Vertice)

```
public class Ciudad {
```

```
    private int transito;
```

```
    private String nombre;
```

```
    public Ciudad {
```

```
}
```

```
    public Ciudad(int trans, String nom) {
```

```
        this.transito = trans;
```

```
        this.nombre = nom;
```

```
}
```

```
    public getNombre() {
```

```
        return nombre;
```

```
}
```

```
    public setNombre(String nom) {
```

```
        this.nombre = nom;
```

```
}
```

```
}
```

```
    public getTransito() {
```

```
        return transito;
```

```
}
```

```
    public setTransito(int trans) {
```

```
        this.transito = trans;
```

```
}
```

NOTA