

```
PUBLIC CLASS PERSONA {
```

```
    PRIVATE BOOLEAN EMPLEADO;
```

```
    PRIVATE STRING NOMBRE;
```

```
    PRIVATE STRING DOMICILIO;
```

```
    PUBLIC PERSONA () {
```

```
    }
```

```
    PUBLIC GETNOMBRE () {
```

```
        RETURN THIS.NOMBRE;
```

```
    }
```

```
    PUBLIC ISEMPLEADO () { RETURN THIS.EMPLEADO }
```

```
}
```

PARTIAL 3 - DFS

```
PUBLIC LISTA GENÉRICA <STRING> RESOLVER ( GRAFO <STRING> CIUDADES, STRING ORIGEN,  
    STRING DESTINO, STRING PASANDO POR );
```

```
    LISTA GENÉRICA <LISTA GENÉRICA <STRING>> RES = NEW ... ();
```

```
    IF ( !CIUDADES.ESVACÍO () && CIUDADES != NULL ) {
```

```
        BOOLEAN [] VISITADOS = NEW BOOLEAN [GRAFO.CIUDADES.LISTADEVENTICES().  
    TAMANIO ()];
```

```
        LISTA GENÉRICA <STRING> ACTUAL = NEW ... ();
```

```
        ACTUAL = CIUDADES.LISTADEVENTICES ();
```

```
        VENTICE <STRING> INICIO = BUSCAR VENTICE (CIUDADES, ORIGEN);
```

```
        IF ( INICIO != NULL ) {
```

```
            DFS ( CIUDADES, INICIO, POSICION ( ), DESTINO, PASANDO POR, VISITADOS, ACTUAL,
```

```
            RES );
```

```
        }  
    }  
    RETURN RES;
```

```
}  
NOTA
```


GNAFO

```

PRIVATE VOID DFS ( GNAFO < STRING > CIUDADES, VERIFICAR INT POS, STRING DESTINO,
STRING INTEN, BOOLEAN EJ VISITADOS, LISTA GENÉRICA < STRING > ACTUAL, LISTA GENÉRICA
< LISTA GENÉRICA < STRING > > RES ) {

```

```

    VISITADOS [POS] = TRUE;

```

```

    ACTUAL . AGREGAR FINAL ( GNAFO . LISTA DE

```

```

    VENTILAS < STRING > VACT = GNAFO . LISTA DE VENTILAS . ELEMENTO (POS);

```

```

    ACTUAL . AGREGAR FINAL ( VACT . DATO (1) );

```

```

    IF ( ( VACT . DATO (1) . EQUALS (DESTINO) ) && ACTUAL . INCLUDE (INTEN) ) {

```

```

        LISTA GENÉRICA < STRING > AUX ;

```

```

        CLONAR (ACTUAL, AUX);

```

```

        RES . AGREGAR FINAL (AUX);

```

```

    }

```

```

    LISTA GENÉRICA < ANISTA < STRING > > ADY = GNAFO . LISTA DE ADY (VACT);

```

```

    ADY . COMENZAR (1);

```

```

    WHILE ( !ADY . FIN (1) )

```

```

        ANISTA < STRING > A = ADY . PROXIMO (1);

```

```

        INT J = A . VENTILAS DESTINO (1) . POSICION (1);

```

```

        IF ( !VISITADOS [J] && A . PESO (1) . EQUALS ("H") )

```

```

            DFS ( GNAFO, J, DESTINO, INTEN, VISITADOS, ACTUAL, RES );

```

```

        }

```

```

    }

```

```

    VISITADOS [POS] = FALSE;

```

```

    ACTUAL . ELIMINAR EN (ACTUAL . TAMAÑO (1) - 1);

```

```

}

```