

1. ¿Qué variable representa la lista ABIERTA?

```
final List<Graph.Vertex<T>> openSet = new  
ArrayList<Graph.Vertex<T>>(size); // The set of tentative  
nodes to be evaluated, initially containing the start node
```

openSet es una variable que guarda el conjunto de nodos que te faltan por expandir

2. ¿Qué variable representa la función g?

```
final Map<Graph.Vertex<T>, Integer> gScore = new  
HashMap<Graph.Vertex<T>, Integer>(); // Cost from start along  
best known path.  
gScore.put(start, 0);
```

La función g es la encargada de decirte el coste del camino desde el nudo inicial hasta n, de la mejor forma, usando los nudos encontrados hasta ahora.

3. ¿Qué variable representa la función f?

```
// Estimated total cost from start to goal through y.  
final Map<Graph.Vertex<T>, Integer> fScore = new  
HashMap<Graph.Vertex<T>, Integer>();  
for (Graph.Vertex<T> v : graph.getVertices())  
    fScore.put(v, Integer.MAX_VALUE);  
fScore.put(start, heuristicCostEstimate(start, goal));
```

La función f es la suma de g+h, es decir, la suma de gScore + la estimación optimista desde n hasta el destino fijado.

4. ¿Qué método habría que modificar para que la heurística representara la distancia aérea entre vértices?

El 1 que devuelve, es la distancia aérea. Si quisieras modificar la distancia, debería modificar ese número.

```
protected int heuristicCostEstimate(Graph.Vertex<T> start,  
Graph.Vertex<T> goal) {  
    return 1;  
}
```

5. ¿Realiza este método reevaluación de nudos cuando se encuentra una nueva ruta a un determinado vértice? Justifique la respuesta

El código no reevalúa los nudos una vez expandidos, ya que es consistente y una vez expandido no se puede reevaluar.