

## **RecorridoCaminoMinimo(DatoOrigen, DatoDestino): cadena**

```
/* Algoritmo genérico para el recorrido del camino mínimo */

1.- CREAR ESTRUCTURA DE NodosEnEspera
2.- CREAR ESTRUCTURA DE NodosExaminados
3.- NodoCaminoMinimo = CREAR OBJETO ClaseNodoCaminoMinimo()
4.- NodoCaminoMinimo.Dato = DatoOrigen
5.- NodoCaminoMinimo.Origen = ""
6.- NodoCaminoMinimo.SumasPesos = 0.0

// Al insertarlo lo ordena en forma ascendente de acuerdo a la suma de
sus pesos
7.- NodosEnEspera.Insertar(NodoCaminoMinimo)

8.- MIENTRAS NOT NodosEnEspera.EstaVacía() REPETIR

    //Obtiene y elimina el NodoCaminoMinimo con la menor suma de pesos
    8.1. NodoCaminoMinimo = NodosEnEspera.Eliminar(NodoCaminoMinimo)

    // Si encuentra el destino: Recorre los NodosExaminados
    8.2. SI NodoCaminoMinimo.Dato == DatoDestino ENTONCES
        8.2.1. NodosExaminados.Insertar(NodoCaminoMinimo)
        8.2.2. Resultado = RECORRER NodosExaminados
        8.2.3. RETURN(Resultado)
    8.3. {FIN DE LA CONDICIONAL DEL PASO 8.2}

    8.4. NodosExaminados.Insertar(NodoCaminoMinimo)

    8.5. NodoGrafo = CREAR OBJETO ClaseNodoGrafo()
    8.6. NodoGrafo.Dato = NodoCaminoMinimo.Dato

    // Busca el nodo en la ListaOrdenadaNodos para obtener sus arcos
    8.7. NodoGrafo = ListaOrdenadaNodos.Buscar(NodoGrafo)

    // Ciclo para analizar cada Arco del NodoGrafo
    8.5. PARA CADA Arco EN NodoGrafo.ListaOrdenadaArcos REPETIR
        8.8.1. Descendiente = CREAR OBJETO ClaseNodoCaminoMinimo()
        8.8.2. Descendiente.Dato = Arco.Destino
        8.8.2. Descendiente.Origen = NodoGrafo.Dato
        8.8.3. Descendiente.SumasPesos = NodoCamino.Minimo.SumasPesos + Arco.Peso

        // Si no existe en NodosEnEspera y en NodosExaminados: insertarlo
en NodosEnEspera (ordenándolo ascendentemente de acuerdo a la suma de
```

```

sus pesos)
    8.8.4. SI NodosEnEspera.Buscar(Descendiente)==nulo AND
NodosExaminados.Buscar(Descendiente)==nulo ENTONCES

        // Al insertarlo lo ordena en forma ascendente de acuerdo a
la suma de sus pesos
        8.8.4.1.1. NodosEnEspera.Insertar(Descendiente)

SINO
    8.8.4.2.1. Descendiente = NodosEnEspera.Buscar(Descendiente)

        // Si existe en NodosEnEspera y es por trayectoria más corta:
actualizar
    8.8.4.2.2. SI Descendiente ≠ nulo AND
NodoCaminoMinimo.SumaPesos + Arco.Peso < Descendiente.SumaPesos
ENTONCES

        // Eliminado de los Nodos en espera
        8.8.4.2.2.1.1. Descendiente =
NodosEnEspera.Eliminar(Descendiente)
        8.8.4.2.2.1.2. Descendiente.Origin =
NodoCaminoMinimo.Dato
        8.8.4.2.2.1.3. Descendiente.SumaPesos =
NodoCaminoMinimo.SumaPesos + Arco.Peso

        // Lo inserta nuevamente con el nuevo valor de la suma
de sus pesos
        8.8.4.2.2.1.4. NodosEnEspera.Insertar(Descendiente)
8.8.4.2.2.2. {FIN DE LA CONDICIONAL 8.8.4.2.2.1}
        8.8.4.2.3. {FIN DE LA CONDICIONAL DEL PASO 8.8.4.2.2}

    8.8.4.2.4. Descendiente = CREAR OBJETO
ClaseNodoCaminoMinimo()
    8.8.4.2.5. Descendiente.Dato = Arco.Destino
    8.8.4.2.6. Descendiente.Origin = NodoGrafo.Dato
    8.8.4.2.7. Descendiente.SumaPesos =
NodoCaminoMinimo.SumaPesos + Arco.Peso
    8.8.4.2.8. Descendiente =
NodosExaminados.Buscar(Descendiente)

        // Si existe en NodosExaminados y es por trayectoria más
corta: actualizar y mover de NodosExaminados a NodosEnEspera
    8.8.4.2.9. SI Descendiente ≠ nulo AND
NodoCaminoMinimo.SumaPesos + Arco.Peso < Descendiente.SumaPesos
ENTONCES

```

```
8.8.4.2.9.1. Descendiente.SumasPesos =
NodoCaminoMinimo.SumasPesos + Arco.Peso

// Al insertarlo lo ordena en forma ascendente de acuerdo
a la suma de sus pesos
8.8.4.2.9.2. NodosEnEspera.Insertar(Descendiente)

// Eliminado de NodosExaminados
8.8.4.2.9.3. NodosExaminados.Eliminar(Descendiente)
8.8.4.2.10.{FIN DE LA CONDICIONAL DEL PASO 8.8.4.2.9}
8.8.5.{FIN DE LA CONDICIONAL DEL PASO 8.8.4}
8.6.{FIN DEL CICLO DEL PASO 8.5}

8.7. NodosExaminados.Insertar(NodoCaminoMinimo)

9.- {FIN DEL CICLO MIENTRAS DEL PASO 8}
10.- RETURN("No hay camino")
```