

```

CLASE GraphSimple(n)
  MÉTODO constructor(n)
    adj ← arreglo de tamaño n, donde cada entrada es una lista vacía
  FIN MÉTODO

  MÉTODO addEdge(u, v, w)
    AGREGAR (v, w) A adj[u]
    AGREGAR (u, w) A adj[v] // grafo no dirigido
  FIN MÉTODO

  MÉTODO dijkstraSimple(source) → arreglo dist
    n ← longitud de adj
    dist ← arreglo[0...n-1] inicializado con  $\infty$ 
    visited ← arreglo[0...n-1] inicializado con false
    dist[source] ← 0

    PARA i DESDE 0 HASTA n-1 HACER
      u ← -1
      PARA j DESDE 0 HASTA n-1 HACER
        SI NOT visited[j] Y (u = -1 O dist[j] < dist[u]) ENTONCES
          u ← j
        FIN SI
      FIN PARA

      SI u = -1 O dist[u] =  $\infty$  ENTONCES
        ROMPER
      FIN SI

      visited[u] ← true

      PARA CADA (v, w) EN adj[u] HACER
        SI NOT visited[v] Y dist[u] + w < dist[v] ENTONCES
          dist[v] ← dist[u] + w
        FIN SI
      FIN PARA
    FIN PARA

    RETORNAR dist
  FIN MÉTODO
FIN CLASE

```