Pseudocódigo Algoritmo de Floyd-Warshall para un grafo con matriz de adyacencia A

1. Inicialización de matrices

$$D(i,j) = \begin{cases} A(i,j), si\ A(i,j) \neq 0, i \neq j \\ 0, si\ i = j \\ \infty, en\ cualquier\ otro\ caso \end{cases} \qquad P(i,j) = \begin{cases} i, si\ A(i,j) \neq 0, i \neq j \\ 0, en\ cualquier\ otro\ caso \end{cases}$$

2. Buscamos la distancia mínima de forma iterativa, empezando por D_1 hasta D_N. Al final obtenemos la matriz de distancias mínimas, D, y la matriz de vértices previos, P:

```
DESDE k=0; k<N; k++                //Analizamos matriz Dk
      DESDE i=0;i<N;i++            //arco i→k
            DESDE j=0;j<N;j++      //arco k→j
                 SI D(i,j)>D(i,k)+D(k,j) ENTONCES
                      D(i,j)=D(i,k)+D(k,j)
                      P(i,j)=P(k,j)
                 FIN_SI
            FIN_DESDE
      FIN_DESDE
FIN_DESDE
```

3. Para obtener el camino más corto entre un par de vértices origen, destino, tenemos la función recursiva `imprimir_camino(P,origen,destino)`:

```
SI origen≠destino
     imprimir_camino(P,origen,P(origen,destino))
FIN_SI
Imprimir destino
```

Pseudocódigo Algoritmo de Prim para obtener el árbol de expansión de coste mínimo

```
1. Inicialización de conjunto de vértices seleccionados
   Selected(i)=0; i=0,…,N-1
   numArcos=0, distanciaTotal=0
   //iniciamos el algoritmo seleccionando el primer vértice
   Selected(0)=1
2. MIENTRAS numArcos<N-1
   minimo=INFINITO
   vx=0, vy=0
   //Busco el arco x-y con valor mínimo, con Selected(vx)=1, Selected(vy)=0
   DESDE i=0;i<N;i++
      SI Selected(i)=1
         DESDE j=0;j<N;j++
            SI Selected(j)≠1 && existearco i-j
               SI minimo>A(i,j)
                  minimo=A(i,j), vx=i, vy=j
               FIN_SI
            FIN_SI
         FIN_DESDE
      FIN_SI
   FIN_DESDE
   //vx-vy es el arco con valor mínimo que añade vy al conjunto Selected
   Selected(vy)=1, numArcos++
   Imprimir VECTOR(x)→VECTOR(y): A(x,y) //Voy imprimiendo cada arco
   distanciaTotal=distanciaTotal+A(x,y)
FIN_MIENTRAS
```