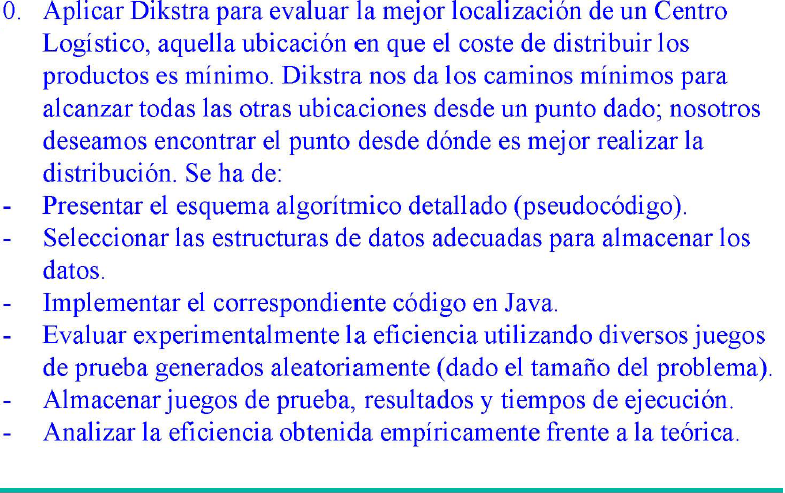
**Autores:** *Francisco Javier Marqués Gaona, Germán Ruano García y David Subires Parra*

***Ejercicio 0 Tema 3***

***Ejercicio 0.***



**Pseudocódigo mejorLocalizacion**

mejorLocalizacion(){

minimo = INFINITO

para v ∈ Vertices hacer

distancias[] = dijkstra(v,Grafo)

sumaDistancia = 0

para cada entero ∈ distancias[] hacer

sumaDistancia = sumaDistancia + distancia[i]

si suma < mínimo

minimo = suma

mejorVertice = v

devolver mejorVertice

}

**Pseudocódigo Dijkstra**

DIJKSTRA (Grafo G, nodo\_fuente s)

para u ∈ V[G] hacer

distancia[u] = INFINITO

padre[u] = NULL

distancia[s] = 0

adicionar (cola, (s, distancia[s]))

mientras que cola no es vacía hacer

u = extraer\_mínimo(cola)

para todos v ∈ adyacencia[u] hacer

si distancia[v] > distancia[u] + peso (u, v) hacer

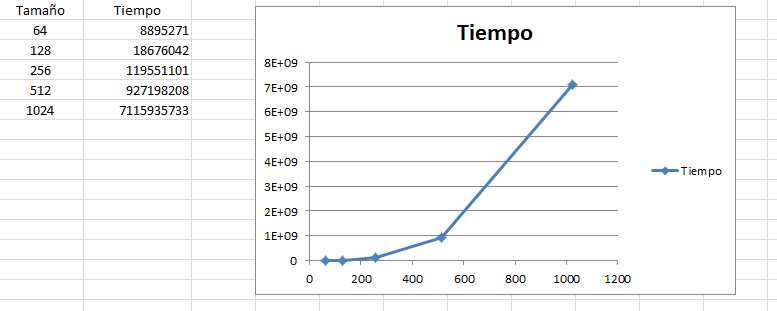
distancia[v] = distancia[u] + peso (u, v)

padre[v] = u

adicionar(cola,(v, distancia[v]))

**Tiempos de ejecución del algoritmo**

Tiempo resultado de ejecutar el programa para distintos tamaños, empezando por un grafo de 64 vértices para consecutivamente ir duplicando su tamaño:



**Eficiencia de los algoritmos analizados teóricamente**

* Algoritmo Dijkstra ():

El Algoritmo de Dijkstra realiza operaciones (sumas y comparaciones) para determinar la longitud del camino más corto entre dos vértices de un grafo ponderado simple, conexo y no dirigido con n vértices.

* *Algoritmo mejorLocalizacion ():*

Se trata de un algoritmo iterativo formado por dos bucles, en el primer bucle realiza una llamada al algoritmo Dijkstra y en el segundo realiza operaciones aritmético-lógicas. Por lo tanto su orden es de t(n) 🡪