

TRABAJO PRACTICO ENTREGABLE NRO 4 - pseudocodigo

```
public ArrayList<Ciudad> getCaminoMasCortoApuerto(ArrayList<Ciudad> ciudades) {
    ArrayList<Ciudad> arrSolucion = new ArrayList<Ciudad>();
    while( !ciudades.vacio && !solucion(arrSolucion) ) {
        ArrayList<Ciudad> ciudadesVisitadas = new ArrayList<Ciudad>();
        Ciudad c = seleccionar(ciudades);           //retorna la primer ciudad que
                                                    //encuentra en el array

        ciudades.borrar(c);
        ciudadesVisitadas = getCaminoDijkstra(c, ciudades); // retorna el camino
                                                    // mas corto de una ciudad a todas las demas

        Ciudad puerto = factible(ciudadesVisitadas); //retorna el puerto + cercano
        if(puerto != null) {
            arrSolucion.add(puerto); //agrego ese puerto a mi solucion
            c.setPuerto(puerto)
        }
    }
    if(solucion(arrSolucion)) {
        return arrSolucion;
    }
    return null;
}
```

```
public Ciudad seleccionar(ArrayList<Ciudad> ciudades) {
    return ciudades.get(0);
}
```

```
public Ciudad factible( ArrayList<Ciudad> ciudadesVisitadas ) {
    ArrayList<Ciudad> puertos = new ArrayList();
    Ciudad puertoCercano;
    for (int i = 0; i < ciudadesVisitadas.size(); i++){
        if(ciudadesVisitadas.get(i).tipo() === "Es Puerto") {
            if(!puertos.contains(ciudadesVisitadas.get(i)){
                puertos.add(ciudadesVisitadas.get(i));
            }
        }
    }
    int menor = puertos.get(0);
    for(int j = 0; j < puertos.size(); j++) {
        if (menor > puertos.get(j+1).getPeso() ) {
            menor = puertos.get(j).getPeso();
            puertoCercano = puertos.get(j);
        }
    }
    return puertoCercano;
}
```

El arreglo “ciudades” contiene todas las ciudades que se representan en el grafo y que están conectadas. Cada ciudad tiene un nombre de tipo *String*, un peso de tipo *int* y un *tipo* que es un *String*: “Es Puerto”, “No es puerto”.

En las primeras iteraciones el arreglo “ciudades” contiene todas las ciudades, al ingresar en el método “*getCaminoMasCortoApuerto*”, dichas ciudades se van a ir eliminando a medida que son elegidas en el método “*seleccionar*”, luego dicha ciudad es eliminada, es por ello que este último método siempre devuelve la primer posición en el arreglo.

El método “*getCaminoDijkstra*”, nos devuelve un arreglo, donde en cada posición está almacenada la ciudad con la distancia más corta a la ciudad pasada por parámetro. Luego en “*factible*”, lo que hacemos es recorrer este arreglo, y elegimos el puerto más cercano a la ciudad elegida desde un principio, y agregamos ese puerto a nuestro arreglo “*arrSolucion*”. Para las últimas iteraciones, vamos a notar que el arreglo “*ciudades*” ya va a tener menos ciudades, ya que se van eliminando a medida que se encuentra el mejor camino. Al final de todo el recorrido, vamos a poder comparar el arreglo “*ciudades*”, guiándonos por sus índices con el “*arrSolucion*”, y así poder saber qué puerto es el más cercano para qué ciudad.