



**UNIDAD PROFESIONAL INTERDISCIPLINARIA DE  
INGENIERÍA CAMPUS ZACATECAS**

**ANÁLISIS DE ALGORITMOS**

**CRUZ LEIJA ROBERTO OSWALDO**

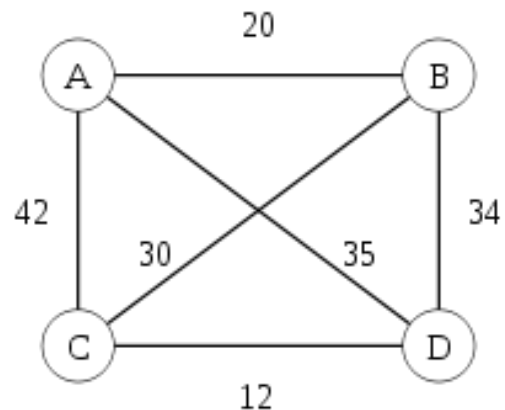
**ROBERTO COVARRUBIAS LUNA**

**TSP**

Introducción:

problema del agente viajero o problema del viajante (TSP por sus siglas en inglés (Travelling Salesman Problem)), responde a la siguiente pregunta: dada una lista de ciudades y las distancias entre cada par de ellas, ¿cuál es la ruta más corta posible que visita cada ciudad exactamente una vez y al finalizar regresa a la ciudad origen? Este es un problema NP-Hard dentro en la optimización combinatoria, muy importante en la investigación de operaciones y en la ciencia de la computación.

El problema fue formulado por primera vez en 1930 y es uno de los problemas de optimización más estudiados. Es usado como prueba para muchos métodos de optimización. Aunque el problema es computacionalmente complejo, una gran cantidad de heurísticas y métodos exactos son conocidos, de manera que, algunas instancias desde cien hasta miles de ciudades pueden ser resueltas.



Código:

Clase Principal:

```
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

public class Principal {

    String[] Ciudades = {"Zcatecas", "Colima", "Aguascalientes", "Guadalajara",
"Durango"};
    Double[] X = {5. ,10. ,5.,40.5,2.};
    Double[] R = {1.,2.,5.8,4.,5.};

    List<String> Lista_Ciudades = new ArrayList<>();
    List<Double> Xs = new ArrayList<>();
    List<Double> Ys = new ArrayList<>();
```

```

String[] n_Ciudades = new String[Lista_Ciudades.size()];
double[][] Cordenadas;
double[] ruta;

public void principal() {

    Pasar_Arrays_a_Listas();
    Pasar_Listas_a_Arrays();

    // Calculos distancias entre nodos.
    double n2 = Math.pow(Cordenadas.length, 0);
    double[] v_distancias = new double[(int) n2];
    v_distancias = getDistancia(Cordenadas);

    TSP tspfb = new TSP (Lista_Ciudades, v_distancias);

    // Mostrar datos
    Mostrar_Listas();
    Mostrar_Tabla(v_distancias, "\nTabla distancias:");
    System.out.println("\nCalculando ruta...");
    System.out.println(tspfb.getRuta());
    System.out.println("\nDistancia recorrida:");
    System.out.println((double) tspfb.getDistanciaRecorrida());

}

private void Pasar_Arrays_a_Listas() {
    Lista_Ciudades.addAll(Arrays.asList(Ciudades));
    Xs.addAll(Arrays.asList(X));
    Ys.addAll(Arrays.asList(R));
}

private void Pasar_Listas_a_Arrays() {
    n_Ciudades = Lista_Ciudades.toArray(n_Ciudades);
    Cordenadas = new double[Xs.size()][Ys.size()];
    for (int i = 0; i < Cordenadas.length; i++) {
        Cordenadas[0][i] = Xs.get(i);
        Cordenadas[1][i] = Ys.get(i);
    }
}

private void Mostrar_Listas() {
    System.out.println("\nCiudades:\n" + Lista_Ciudades);
    System.out.println("\nCoordenadas:    ");
}

```

```

        System.out.println("X:" + Xs);
        System.out.println("Y:" + Ys);
    }

    public void Mostrar_Tabla(double[] v1, String msg) {
        System.out.println(msg);
        int n = (int) Math.sqrt(v1.length);
        double t1[][] = new double[n][n];
        int cont = 0;
        String aux = "";
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                t1[j][i] = v1[cont];
                aux += ((int) (t1[j][i])) + "\t";
                cont++;
            }
            System.out.println(aux);
            aux = "";
        }
    }

    //Genera Distancia
    public double[] getDistancia(double[][] xy) {
        int cont = 0;
        double[] v1 = new double[xy[0].length * xy[0].length];
        for (int i = 0; i < xy[0].length; i++) {
            for (int j = 0; j < xy[0].length; j++) {
                v1[cont] = Math.sqrt(Math.pow((xy[0][i] - xy[0][j]), 2) + Ma
th.pow((xy[1][i] - xy[1][j]), 2));
                cont++;
            }
        }
        return v1;
    }
}

```

Clase Ruta:

```

public class Ruta {
    public static void main(String[] args) {
        Principal p = new Principal();
        p.principal();
    }
}

```

Clase TSP:

```
import java.util.List;

public class TSP {

    private final List<String> Lista_Ciudades;
    private final double[] Distancia;
    private String RutaCorta = "";
    private long minim = Long.valueOf("100000");
    String Ciudades = "0123456789abcdefghijklmnopqrstuvwxyz";

    //Constructor
    public TSP(List<String> lista_nombre_nodos, double[] v_distancias) {
        this.Lista_Ciudades = lista_nombre_nodos;
        this.Distancia = v_distancias;
    }

    //Método para obtener ruta corta
    public String getRuta() {
        //Control parametros correctos
        if (Lista_Ciudades.size() == Math.sqrt(Distancia.length)) {
            double[][] tabla_distancias = conversorVT(Distancia);
            int nCiudades = Lista_Ciudades.size();
            int vMiniRuta[] = new int[nCiudades];
            String nodos = Ciudades.substring(0, nCiudades);
            String ruta = "";
            long index = 0;
            while (ruta.length() <= nCiudades) {
                int vRuta[] = new int[Distancia.length];
                double suma = 0;
                ruta = Long.toString(index, nCiudades);
                // control del 0
                if (ruta.length() == nCiudades - 1) {
                    ruta = "0" + ruta;
                }
                if (ruta.length() == nCiudades) {
                    if (validar(ruta, nodos)) { // eliminar repetidos
                        for (int i = 0; i < ruta.length(); i++) {
                            vRuta[i] = Integer.parseInt("" + ruta.charAt(i),
36);
                        }
                        // calculo distancias
                        suma += tabla_distancias[vRuta[nCiudades - 1]][vRuta
[0]];
                    }
                }
            }
        }
    }
}
```

```

        for (int i = 0; i < (nCiudades - 1); i++) {
            suma += tabla_distancias[vRuta[i]][vRuta[i + 1]]
;

            if (suma > minim) { // podador
                break;
            }
        }
        if (minim > suma) {
            minim = (long) suma;
            vMiniRuta = vRuta;
        }
    }
    index++;
}
for (int i = 0; i < nCiudades; i++) {
    RutaCorta += Lista_Ciudades.get(vMiniRuta[i]) + " - ";
}
return RutaCorta;
} else {
    return null;
}
}

public long getDistanciaRecorrida() {
    return minim;
}

public double[][] conversorVT(double[] v_distancias) {
    int nNodos = (int) Math.sqrt(v_distancias.length);
    double[][] tabla_distancias = new double[nNodos][nNodos];
    int cont = 0;
    for (int i = 0; i < nNodos; i++) {
        for (int j = 0; j < nNodos; j++) {
            tabla_distancias[j][i] = v_distancias[cont] ;
            cont++;
        }
    }
    return tabla_distancias;
}

private static boolean validar(String ruta, String nodos) {
    for (int i = 0; i < nodos.length(); i++) {
        if (!ruta.contains("" + nodos.charAt(i))) {
            return false;
        }
    }
}

```

```

    }
}
return true;
}
}

```

Capturas de pantalla:

Ciudades	Zacatecas	Colima	Aguascalientes	Guadalajara	Durango
Eje X	5	10	5	40	2
Eje y	1	2	5	4	5

```

Ciudades:
[Zcatecas, Colima, Aguascalientes, Guadalajara, Durango]

Coordenadas:
X:[5.0, 10.0, 5.0, 40.0, 2.0]
Y:[1.0, 2.0, 5.0, 4.0, 5.0]

Tabla distancias:
0      5      4      35      5
5      0      5      30      8
4      5      0      35      3
35     30     35     0      38
5      8      3      38     0

Calculando ruta...
Zcatecas - Colima - Guadalajara - Aguascalientes - Durango -

Distancia recorrida:
78.0
BUILD SUCCESSFUL (total time: 0 seconds)

```

Ciudades	Zacatecas	Colima	Aguascalientes	Guadalajara	Durango	México	Juárez	Tijuana
Eje X	5	10	5	40	2	2	4	5
Eje y	1	2	5	4	5	2	3	4

```

Ciudades:
[Zcatecas, Colima, Aguascalientes, Guadalajara, Durango, Mexico, Jurez, Tijuana]

Coordenadas:
X:[5.0, 10.0, 5.0, 40.0, 2.0, 2.0, 4.0, 5.0]
Y:[1.0, 2.0, 5.0, 4.0, 5.0, 2.0, 3.0, 4.0]

Tabla distancias:
0      5      4      35      5      3      2      3
5      0      5      30      8      8      6      5
4      5      0      35      3      4      2      1
35     30     35     0      38     38     36     35
5      8      3      38     0      3      2      3
3      8      4      38     3      0      2      3
2      6      2      36     2      2      0      1
3      5      1      35     3      3      1      0

Calculando ruta...
Zcatecas - Colima - Guadalajara - Aguascalientes - Tijuana - Durango - Mexico - Jurez -

Distancia recorrida:
81.0

```