# Java electrostatica

import java.util.ArrayList;

import java.util.List;

class Carga {

double carga; // en coulombs

double[] posicion; // posición en 3D [x, y, z]

public Carga(double carga, double[] posicion) {

this.carga = carga;

this.posicion = posicion;

}

}

public class Electrostatica {

private static final double K = 8.9875517873681764e9; // N m²/C²

public static void main(String[] args) {

List<Carga> cargas = new ArrayList<>();

// Agregar cargas al sistema

cargas.add(new Carga(1e-6, new double[]{0, 0, 0})); // 1 microcoulomb en el origen

cargas.add(new Carga(-2e-6, new double[]{1, 0, 0})); // -2 microcoulombs en (1,0,0)

// Punto donde se quiere calcular el campo eléctrico

double[] puntoCampo = {0.5, 0, 0}; // Punto en (0.5, 0, 0)

double[] campoTotal = calcularCampoElectrico(cargas, puntoCampo);

System.out.println("El campo eléctrico total en el punto es: (" + campoTotal[0] + ", " + campoTotal[1] + ", " + campoTotal[2] + ") N/C");

}

public static double[] calcularCampoElectrico(List<Carga> cargas, double[] punto) {

double[] campoTotal = {0, 0, 0};

for (Carga carga : cargas) {

double[] campo = calcularCampoPorCarga(carga, punto);

campoTotal[0] += campo[0];

campoTotal[1] += campo[1];

campoTotal[2] += campo[2];

}

return campoTotal;

}

public static double[] calcularCampoPorCarga(Carga carga, double[] punto) {

double[] campo = new double[3];

// Distancia desde la carga al punto

double dx = punto[0] - carga.posicion[0];

double dy = punto[1] - carga.posicion[1];

double dz = punto[2] - carga.posicion[2];

double distancia = Math.sqrt(dx \* dx + dy \* dy + dz \* dz);

// Evitar división por cero

if (distancia == 0) {

throw new IllegalArgumentException("El punto no puede coincidir con la posición de la carga.");

}

// Calcular la magnitud del campo eléctrico

double magnitudCampo = (K \* carga.carga) / (distancia \* distancia);

campo[0] = magnitudCampo \* (dx / distancia); // Componente x

campo[1] = magnitudCampo \* (dy / distancia); // Componente y

campo[2] = magnitudCampo \* (dz / distancia); // Componente z

return campo;

}

}