Actividad 1 Aplicar la técnica de Divide y Vencerás en una lista de clientes con id, nombre y scoring, buscando el cliente con el scoring máximo. Resolver mediante pseudocódigo Implementar en java

Función encontrarMaximoScoring(clientes, inicio, fin):

Si inicio == fin:

Devolver clientes[inicio] // Caso base: solo un cliente

medio = (inicio + fin) / 2 // Dividir la lista en dos mitades

// Resolver recursivamente para cada mitad

maxIzquierda = encontrarMaximoScoring(clientes, inicio, medio)

maxDerecha = encontrarMaximoScoring(clientes, medio + 1, fin)

// Combinar: devolver el cliente con el scoring máximo

Si maxIzquierda.scoring > maxDerecha.scoring:

Devolver maxIzquierda

Sino:

Devolver maxDerecha

**Implementacion en java.**

class Cliente {

int id;

String nombre;

int scoring;

public Cliente(int id, String nombre, int scoring) {

this.id = id;

this.nombre = nombre;

this.scoring = scoring;

}

@Override

public String toString() {

return "Cliente{" +

"id=" + id +

", nombre='" + nombre + '\'' +

", scoring=" + scoring +

'}';

}

}

public class DivideYVencerasCliente {

public static void main(String[] args) {

// Lista de clientes

Cliente[] clientes = {

new Cliente(1, "Juan", 85),

new Cliente(2, "Ana", 90),

new Cliente(3, "Luis", 78),

new Cliente(4, "Maria", 95),

new Cliente(5, "Pedro", 88)

};

// Encontrar el cliente con el scoring máximo

Cliente clienteMaximo = encontrarMaximoScoring(clientes, 0, clientes.length - 1);

// Mostrar el resultado

System.out.println("El cliente con el scoring máximo es: " + clienteMaximo);

}

// Método para encontrar el cliente con el scoring máximo usando Divide y Vencerás

public static Cliente encontrarMaximoScoring(Cliente[] clientes, int inicio, int fin) {

// Caso base: si solo hay un cliente

if (inicio == fin) {

return clientes[inicio];

}

// Dividir la lista en dos mitades

int medio = inicio + (fin - inicio) / 2;

// Resolver recursivamente para cada mitad

Cliente maxIzquierda = encontrarMaximoScoring(clientes, inicio, medio);

Cliente maxDerecha = encontrarMaximoScoring(clientes, medio + 1, fin);

// Combinar: devolver el cliente con el scoring máximo

if (maxIzquierda.scoring > maxDerecha.scoring) {

return maxIzquierda;

} else {

return maxDerecha;

}

}

}