



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGON**



TAREA: DIVIDE Y VENCERAS

P R E S E N T A

Alexis Hernández Zamudio

APROFESOR

Jesús Hernández Cabrera

Gpo:1558

Ciudad Nezahualcóyotl, EDOMEX. 3 DE noviembre del 2025

CODIGO

```
import java.util.Scanner;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class ProblemaMochila {  ▲ AlextyrB *

    public static int[][] construirTablaDP(int[] pesos, int[] valores, int capacidad) { 1 usage  ▲ AlextyrB *
        int n = pesos.length;
        int[][] dp = new int[n + 1][capacidad + 1];

        for (int i = 1; i <= n; i++) {
            int pesoActual = pesos[i - 1];
            int valorActual = valores[i - 1];

            for (int w = 1; w <= capacidad; w++) {
                if (pesoActual <= w) {
                    dp[i][w] = Math.max(dp[i - 1][w], valorActual + dp[i - 1][w - pesoActual]);
                } else {
                    dp[i][w] = dp[i - 1][w];
                }
            }
        }

        return dp;
    }

    public static List<String> rastrearSolucion(int[] pesos, int[] valores, int capacidad, int[][] dp) { 1 usage  ▲ AlextyrB *
        List<String> solucion = new ArrayList<>();
        int i = pesos.length;

        int w = capacidad;
        while (i > 0 && w > 0) {
            if (dp[i][w] != dp[i - 1][w]) {
                solucion.add("Objeto " + i + " (Peso=" + pesos[i - 1] + ", Valor=" + valores[i - 1] + ")");
                w -= pesos[i - 1];
            }
            i--;
        }

        Collections.reverse(solucion);
        return solucion;
    }

    public static void main(String[] args) {  ▲ AlextyrB *
        Scanner scanner = new Scanner(System.in);

        System.out.println("PROBLEMA DE LA MOCHILA");

        System.out.print("Ingrese la capacidad máxima de la mochila: ");
        int capacidad = scanner.nextInt();

        System.out.print("Ingrese el número total de objetos: ");
        int n = scanner.nextInt();
    }
}
```

```

53
54     int[] pesos = new int[n];
55     int[] valores = new int[n];
56
57     System.out.println("\n***** Ingreso de Datos de Objetos *****");
58     for (int i = 0; i < n; i++) {
59         System.out.println("Objeto " + (i + 1) + ":");
60         System.out.print("  Peso: ");
61         pesos[i] = scanner.nextInt();
62         System.out.print("  Valor: ");
63         valores[i] = scanner.nextInt();
64     }
65
66     scanner.close();
67     System.out.println("\n*****RESULTADOS *****");
68
69     try {
70         int[][] dp = construirTablaDP(pesos, valores, capacidad);
71         int valorMaximo = dp[n][capacidad];
72         List<String> objetosIncluidos = rastrearSolucion(pesos, valores, capacidad, dp);
73
74         System.out.println("Valor Máximo Obtenido: " + valorMaximo);
75         System.out.println("Objetos seleccionados:");

```

```

75         System.out.println("Objetos seleccionados:");
76
77         if (objetosIncluidos.isEmpty()) {
78             System.out.println(" - La mochila está vacía (capacidad muy pequeña o no hay objetos).");
79         } else {
80             for (String objeto : objetosIncluidos) {
81                 System.out.println(" - " + objeto);
82             }
83         }
84     } catch (Exception e) {
85         System.out.println("Error al procesar los datos. Asegúrese de ingresar números enteros válidos.");
86     }
87
88     System.out.println("*****");
89 }
90 }

```

PROBLEMA DE LA MOCHILA

Ingrese la capacidad máxima de la mochila: 5

Ingrese el número total de objetos: 5

***** Ingreso de Datos de Objetos *****

Objeto 1:

Peso: 2

Valor: 3

Objeto 2:

Peso: 3

Valor: 3

Objeto 3:

Peso: 1

Valor: 1

Objeto 4:

Peso: 4

Valor: 2

Objeto 5:

Peso: 1

Valor: 6

*****RESULTADOS *****

Valor Máximo Obtenido: 10

Objetos seleccionados:

- Objeto 1 (Peso=2, Valor=3)
- Objeto 3 (Peso=1, Valor=1)
- Objeto 5 (Peso=1, Valor=6)

Process finished with exit code 0