

15 de mayo de 2025

Padrón:

Apellido y Nombre:

1. Mostrar la salidad y detallar el trabajo del GC

```
public static void main(String[] args) {
    int X, Y, Z = 5;
    char W;
    int T;
    int[] A = new int[4];
    int[] B = A;

    T = 77;
    W = 'C';
    X = T;
    A[0] = X;
    B[1] = A[0] / 2;

    System.out.println(X + "-" + W + "-" + B[0] + "-" + A[1]);

    W = (char)(T + 1);
    B[2] = A[0] * 2;

    System.out.println(W + "-" + T + "-" + B[1] + "-" + A[2]);

    while (A[0] > 30) {
        T -= 2;
        X -= 3;
        Y = X + Z;
        System.out.println(Y + "-" + T + "-" + B[1] + "-" + Z);
        Z = Y;
        A[0] -= 10;
    }

    // ¿Qué hace el garbage collector aquí? ¿Sobre qué variable?
}
```

2. Implementar el método **buscarMejoresAlumnos** de la clase **Profesor** a partir de las siguientes especificaciones:

```
public class Profesor {

    /**
     * Busca los mejores alumnos de cada materia segun su promedio de notas
     */
    public List<Alumno> buscarMejoresAlumnos(List<Alumno> alumnos, List<Nota> promedios);
}
```

<pre>public class Alumno { private String nombre; private int padron; //Devuelve el nombre del alumno public String getNombre() { return this.nombre;} //Devuelve el padron del alumno public int getPadron() { return this.padron;} }</pre>	<pre>public class Nota { private int padron; private String materia; private double valor; //Devuelve el padron public int getPadron() {return padron;} //Devuelve el nombre de la materia public String getMateria() {return materia;} //Devuelve la nota entre 0 y 10 public double getValor() {return valor;} }</pre>
--	---

3. Se desea modelar una **batería de auto eléctrico**, donde se pueden realizar cargas, desplazamientos (que consumen energía), y se lleva un control del estado de carga.

El TDA debe permitir:

1. **crearBateria**: recibe como parámetros:
 - o la capacidad máxima en kWh de la batería.
 - o el consumo promedio del auto en kWh por kilómetro.
2. **cargar**: recibe un valor en kWh y lo suma a la carga actual.
3. **desplazar**: recibe una distancia en kilómetros y reduce la carga en base al consumo promedio. No se puede desplazar si no hay carga suficiente.
4. **obtenerCargaActual**: devuelve la carga actual en kWh.
5. **obtenerAutonomia**: devuelve cuántos kilómetros se pueden recorrer con la carga actual.
6. **estaVacía**: devuelve true si la batería está sin carga.
7. **estaLlena**: devuelve true si la batería está completamente cargada.

Se aprueba por punto, para tener bien el punto hay que tener el 60% bien del ejercicio.
Una vez empezado el examen, no se puede utilizar el celular y no se puede salir del aula.
Duración del examen: 2:30 horas