

**Refactorización del nombre de las variables principales y de la clase, ya que los nombres eran ambiguos:**

### Antes

```
1  ✓ public class p {  
2  
3  ✓      public static void main(String[] args) {  
4  
5          int[] n = {5, 7, 3, 7, 2, 9, 7};  
6          int a = 0;  
7          int b = 0;  
8          int c = n[0];  
9
```

### Después

```
//Refactorización de nombre a un nombre mas entendible  
public class AnalizadorNotas { new *
```

```
//Refactorizado por un nombre mas claro  
int[] notas = {5, 7, 3, 7, 2, 9, 7};  
//Refactorizado por un nombre mas claro  
int max = obtenerNotaMaxima(notas);  
//Refactorizado por un nombre mas claro  
int repeticiones = contarRepeticiones(notas, max);  
//Refactorizado por un nombre mas claro  
int notaMax = notas[0];
```

**Cambio de los bucles for que saturaban el main a métodos a parte los cuales cumplen con lo que hacían esos bucles de una manera más clara:**

## **Antes**

```
10         for (int i = 1; i < n.length; i++) {
11             if (n[i] > c) {
12                 c = n[i];
13             }
14         }
15
16         for (int i = 0; i < n.length; i++) {
17             if (n[i] == c) {
18                 a++;
19             }
20         }
21
22         if (a > 1) {
23             b = 1;
24         }
25
26         if (b == 1) {
27             System.out.println("SI");
28         } else {
29             System.out.println("NO");
30         }
31
32         int s = 0;
33         for (int i = 0; i < n.length; i++) {
34             s += n[i];
35         }
36
37         double r = (double) s / n.length;
38         System.out.println(r);
39
40         if (r >= 5) {
41             System.out.println("BIEN");
42         } else {
43             System.out.println("MAL");
44         }
45     }
```

## Después

```
34  /**
35   * Busca el valor más alto dentro de un array de notas.
36   *
37   * @param notas Array de enteros con las calificaciones.
38   * @return El valor máximo encontrado.
39   */
40  @ public static int obtenerNotaMaxima(int[] notas) { 1 usage new *
41      int maximo = notas[0]; // Asumimos que el array no está vacío
42      for (int i = 1; i < notas.length; i++) {
43          if (notas[i] > maximo) {
44              maximo = notas[i];
45          }
46      }
47      return maximo;
48  }
49
50  /**
51   * Cuenta cuántas veces aparece un número específico en el array.
52   *
53   * @param notas Array de enteros con las calificaciones.
54   * @param valorBuscado El número del cual queremos contar sus apariciones.
55   * @return La cantidad de veces que aparece el valor.
56   */
57  @ public static int contarRepeticiones(int[] notas, int valorBuscado) { 1 usage new *
58      int contador = 0;
59      for (int i = 0; i < notas.length; i++) {
60          if (notas[i] == valorBuscado) {
61              contador++;
62          }
63      }
64      return contador;
65  }
66
67  /**
68   * Calcula la media aritmética de las notas.
69   *
70   * @param notas Array de enteros con las calificaciones.
71   * @return El promedio calculado como un valor decimal.
72   */
73  @ public static double calcularPromedio(int[] notas) { 1 usage new *
74      int sumaTotal = 0;
75      for (int i = 0; i < notas.length; i++) {
76          sumaTotal += notas[i];
77      }
78      // Hacemos cast a (double) para no perder los decimales en la división
79      return (double) sumaTotal / notas.length;
80  }
81
82  }
```

**Eliminación de la variable b ya que era redundante y hacia algo en su condición que se puede hacer con la variable a o ahora llamada repeticiones:**

**Antes**

```
if (a > 1) {  
    b = 1;  
}  
  
if (b == 1) {  
    System.out.println("SI");  
} else {  
    System.out.println("NO");  
}
```

**Después**

```
if (repeticiones > 1) {  
    System.out.println("SI"); // El máximo se repite  
} else {  
    System.out.println("NO");  
}
```