



## 4. Sobrecarga de funciones

La sobrecarga de funciones es una propiedad de la programación orientada a objetos. Como en Java estamos usando, desde el primer momento, clases y objetos, podemos aplicar la sobrecarga a cualquier función aunque todavía no estamos creando nuestras propias clases. De hecho, ya hemos aplicado la sobrecarga, pero con operadores: al usar el operador + con números, estamos realizando su suma, mientras que el mismo operador utilizado con cadenas las concatena.

En Java, como en otros lenguajes que admitan programación orientada a objetos, podemos llamar con el mismo nombre a varias funciones siempre y cuando tengan un número o un tipo de parámetros diferentes. Esta condición es indispensable para que el programa, cuando llamemos a una función y existan varias con el mismo nombre, sea capaz de diferenciar cada una de ellas y saber exactamente a cuál nos estamos refiriendo.

La ventaja de trabajar con funciones sobrecargadas es que podemos llamar de la misma forma a funciones que realizan las mismas tareas, lo que redundará en una mayor legibilidad del programa.

Pongamos por ejemplo la siguiente función, a la que le pasamos dos números y nos devuelve su suma:

```
public static int suma(int numero1, int numero2) {  
    return numero1 + numero2;  
}
```

Si quisiéramos disponer de otra función que haga lo mismo, pero con tres números, podría llamarse de la misma manera:

```
public static int suma(int numero1, int numero2, int numero3) {  
    return numero1 + numero2 + numero3;  
}
```

Cuando llamemos a la función suma, el programa decidirá cuál de las dos debe ejecutar en función del número de parámetros que contenga:

```
int suma1 = suma(numero1, numero2); // llamaría a la primera función suma  
int suma2 = suma(numero1, numero2, numero3); // llamaría a la segunda función  
suma
```

Supongamos que tenemos otra función suma que devuelve la suma de dos números con decimales:

```
public static float suma(float numero1, float numero2) {  
    return numero1 + numero2;  
}
```

En este caso, el número de parámetros es el mismo que en la primera función suma que hemos visto, pero como su tipo es distinto, el programa seguirá siendo capaz de diferenciarlas y de saber a cuál de ellas nos referimos. Cuando hagamos:

```
suma1 = suma(numero1, numero2);
```

se llamará a la función suma correspondiente en función del tipo de numero1 y numero2, siempre que sean ambos int o float —las dos opciones que hemos programado—.

**IMPORTANTE**

La sobrecarga de funciones será de más utilidad cuando trabajemos con nuestras propias clases. En próximas unidades veremos cómo rentabilizar al máximo esta posibilidad de dar el mismo nombre a funciones distintas pero que realicen las mismas (o equivalentes) tareas.

Como sería muy poco práctico disponer de funciones que sumen cualquier número de parámetros, podríamos diseñar una función a la que le pasemos un array de números y nos devuelva su suma. De hecho, podríamos diseñar dos funciones: una para sumar una cantidad indeterminada de enteros y otra para sumar una cantidad indeterminada de números con decimales.

```
public static float suma(float[] arrayDecimales) {  
    float total = 0.0F;  
    for(int i = 0; i < arrayDecimales.length; i++) {  
        total = total + arrayDecimales[i];  
    }  
    return total;  
}  
  
public static int suma(int[] arrayEnteros) {  
    int total = 0;  
    for(int i = 0; i < arrayEnteros.length; i++) {  
        total = total + arrayEnteros[i];  
    }  
    return total;  
}
```

Ahora disponemos de dos funciones para sumar cualquier cantidad de números enteros o con decimales. El programa decidirá a cuál de ellas llamar en función del tipo de array que le pasemos como parámetro. Si desde nuestro programa principal hacemos lo siguiente:

```
int[] arrayEnteros = {5, 4, 3, 2, 1};  
float[] arrayDecimales = {5.3F, 4.2F, 3.7F, 2.15F, 1.08F};  
System.out.println("La suma de los enteros es " + suma(arrayEnteros));  
System.out.println("La suma de los decimales es " + suma(arrayDecimales));
```

el sistema nos mostrará por pantalla:

La suma de los enteros es 15

La suma de los decimales es 16.43

