

# MÉTODOS EN JAVA

## MÉTODOS EN JAVA

### Definición general de un método en Java:

```
class NombreClase{  
  
    tipo nombreMetodo(lista_de_argumentos) {  
        //Cuerpo del método  
    }  
}
```

### Ejemplo de un método en Java:

```
public class Aritmetica{  
  
    int sumar(int a, int b){  
        //realiza la suma y regresa el resultado como un entero  
        return a + b;  
    }  
}
```

Como mencionamos anteriormente, una clase consiste de atributos y métodos. El tema de métodos es muy amplio, sin embargo en esta primera introducción veremos cómo podemos crear y usar los métodos que declaremos.

Podemos observar que en Java, es necesario que para crear un método debe estar dentro de una clase. Esto se debe a que Java es un lenguaje orientado a objetos, y por lo tanto una de las prácticas es que los métodos deben ser parte de una clase para estar correctamente definidos.

Por otro lado, podemos observar que la definición general de un método. En primer lugar podemos observar el tipo que regresará este método. Algunos métodos no regresan ningún valor en particular, por lo que en Java utilizaremos la palabra reservada **void** para este caso donde no se regrese ningún valor. Podemos especificar tipos primitivos, o de tipo Object, incluyendo los que nosotros creamos, como el tipo Persona que creamos anteriormente.

Posteriormente del tipo, especificamos el nombre del método, debemos recordar que Java es un lenguaje sensible a mayúsculas y minúsculas, por lo que debemos respetar el nombre del método y usarlo tal como se escribe. Es común que el nombre del método sea un acción, por ejemplo insertarPersona, borrarAlumno, etc. También es común en Java utilizar la notación de camello para escribir los nombres tanto de variables como de métodos, y en general de los nombres que escribimos. Esta notación consiste en escribir la primera letra en minúscula y posteriormente la primera letra de cada palabra debe ir en mayúscula. Conforme avancemos iremos observando ejemplos de este tipo de notación a lo largo del curso.

Posterior al nombre, especificamos la lista de argumentos, especificando primero el tipo y posteriormente el nombre del argumento. Por cada argumento que deseemos agregar, debemos separarlos por una coma. Si el método no tiene lista de argumentos, sólo veremos que se abren y cierran paréntesis ( ).

Finalmente escribiremos el cuerpo del método entre llaves { }. Todo lo que escribamos dentro de estas llaves es lo que ejecutará el método. Podemos ver el ejemplo del método sumar. En este caso debido a que el método regresa un valor, usamos la palabra reservada return seguida del valor a regresar al método que hizo la llamada al método sumar.

Un concepto importante al definir métodos en Java, se conoce como la firma del método. En el caso del método sumar, la firma del método sería: `int sumar(int, int);` //firma del método

Es decir, el método devuelve un valor entero, su nombre es sumar, y recibe dos argumentos de tipo entero. Así la firma de este método nos indica cómo podemos utilizarlo, que valor esperar y también si debemos enviar argumentos o no, y de que tipo son estos argumentos. Posteriormente pondremos en práctica este ejemplo.

# LLAMANDO A UN METODO

## Llamada general de un método en Java:

```
//Crear un objeto de la clase a llamar su método
TipoClase objeto = new TipoClase();

//Llamamos el método, enviando argumentos si se requieren
//Si el método regresa un valor podemos recibirlo según el tipo
tipoDevuelto resultado = objeto.nombreMetodo(arg1, arg2, etc);
```

## Ejemplo de llamada a un método en Java:

```
//Creamos un objeto de la clase Aritmetica
Aritmetica a = new Aritmetica();

//Llamamos el metodo sumar y recibimos el valor devuelto
int resultado = a.sumar(5, 3);
```

Una vez que hemos agregado un método a una clase, podemos hacer uso de él haciendo lo siguiente:

- 1) Debemos declarar un objeto del tipo de la clase que tiene el método que nos interesa utilizar.
- 2) Por medio del objeto declarado, utilizamos el operador punto (.) y posterior a este operador escribimos el nombre del método que nos interesa llamar. Recordemos que en Java, los nombres son sensibles a mayúsculas y minúsculas, por lo que debemos de hacer la llamada al método tal y como lo hayamos escrito, respetando las mayúsculas y minúsculas que hayamos utilizado para definir el método.
- 3) Posterior del nombre del método, abrimos y cerramos paréntesis. Si el método no recibe ningún argumento, solo abrimos y cerramos paréntesis, sin embargo si el método fue escrito para recibir argumentos, debemos proporcionar los argumentos del tipo esperado, separando por comas cada uno de ellos.
- 4) Finalmente, si el método fue escrito para regresar un valor, es opcional el recibirlo con una variable del tipo que regresa el método. Debemos notar que el tipo devuelto por un método no tiene que ser igual a los tipos de los argumentos recibidos. Para ello debemos observar la firma del método que vayamos a utilizar. Si recordamos, la firma del método sumar sería:

```
int sumar(int, int); //firma del método
```

Por lo que observamos que el método recibo dos argumentos de tipo entero, pero podría ser cualquier otro tipo, se llama sumar y devuelve un tipo entero, pero también podría ser cualquier otro tipo. Posteriormente pondremos en práctica este ejemplo y varios más para que vaya quedando cada vez más claro la declaración y uso de métodos.