


Open   Opened 2 weeks ago by  **Rubén Montero**

# Parámetros

## Resumen

- Entenderemos que los métodos admiten *parámetros*
- Veremos cómo se declaran los *parámetros formales*
- Veremos cómo se envían (al invocar el método) los *parámetros actuales*
- Escribiremos un ejemplo en `SimpleMathDemo` y lo invocaremos desde `Main.java`

## Descripción

Los métodos pueden devolver valores, pero también *recibirlos*.

Recordemos que en la primera tarea dijimos que en la **declaración** de un método, se indica:

- Paréntesis indicando los argumentos del método. Si no recibe argumentos, se escriben paréntesis vacíos `()` igualmente.

Todos nuestros métodos hasta ahora *no* reciben parámetros:

```
public void sayHello() { ... }
public String byeWorld() { ... }
public int sumTwoNumbers() { ... }
public void printMultiplicationTable() { ... }
public void loop50Times() { ... }

...
```

Sin embargo, además de *devolver* datos (salida) también está permitido el **flujo de datos de entrada**, *hacia* el método.

## ¿Y cómo?

### Parámetros formales

En primer lugar, debemos *declarar* **qué** parámetros espera el método.

Por ejemplo, si queremos recibir un número **entero** en un método `printNumber`, lo *declararíamos* así:

```
public void printNumber(int nombreArbitrario) {

}
```

Aquí, `nombreArbitrario` es un *nombre* cualquiera. Lo usaremos para referirnos al **valor del parámetro** desde **dentro** del método. Por ejemplo:

```
public void printNumber(int nombreArbitrario) {
    System.out.println(nombreArbitrario);
}
```

Nótese que **no** hace falta que `nombreArbitrario` coincida con ninguna variable en otras clases o ficheros de código.

### Parámetros actuales

Pero, ¿**qué** valor tiene `nombreArbitrario`?

Pues eso depende de con **qué** valor *invocamos* el método. Por ejemplo, imaginate en `Main.java`:

```
public class Main {
    public static void main(String[] args) {
        // ...
        SimpleMathDemo demo = new SimpleMathDemo();
        demo.printNumber(10);
    }
}
```

```
}
```

Aquí, `10` es lo que se conoce como *parámetro actual*. Es decir, el **valor real** en tiempo de ejecución.

## ¡Lo entiendo!

¡Muy bien!

Para terminar, deja escrito el método `printNumber` en tu clase `SimpleMathDemo.java`.

Invócalo desde `Main.java` con un valor arbitrario.

## Por último

Una vez verifiques que el *test* funciona correctamente, la tarea ha sido completada.

Haz `commit` y `push` para subir los cambios al repositorio.



[Rubén Montero @ruben.montero](#) changed milestone to [%Sprint 1](#) 2 weeks ago



[Ania Blanco @ania.blanco](#) mentioned in commit [1af1bb45](#) 2 weeks ago