

# 15 - Funciones: parámetros

Vimos en el concepto anterior que una función resuelve una parte de nuestro algoritmo.

Tenemos por un lado la declaración de la función por medio de un nombre y el algoritmo de la función seguidamente. Luego para que se ejecute la función la llamamos desde la función main.

Ahora veremos que una función puede tener parámetros para recibir datos. Los parámetros nos permiten comunicarle algo a la función y la hace más flexible.

## Problema 1

Confeccionar una aplicación que muestre una presentación en pantalla del programa. Solicite la carga de dos valores y nos muestre la suma.

Mostrar finalmente un mensaje de despedida del programa.

Proyecto74 - Principal.kt

```
fun mostrarMensaje(mensaje: String) {
    println("*****")
    println(mensaje)
    println("*****")
}

fun cargarSumar() {
    print("Ingrese el primer valor:")
    val valor1 = readln().toInt()
    print("Ingrese el segundo valor:")
    val valor2 = readln().toInt()
    val suma = valor1 + valor2
    println("La suma de los dos valores es: $suma")
}

fun main(parametro: Array<String>) {
    mostrarMensaje("El programa calcula la suma de dos
        valores ingresados por teclado.")
    cargarSumar()
    println("Gracias por utilizar este programa")
}
```

Ahora para resolver este pequeño problema hemos planteado una función llamada `mostrarMensaje` que recibe como parámetro un `String` (cadena de caracteres) y lo muestra en pantalla.

Los parámetros van seguidos del nombre de la función encerrados entre paréntesis (y en el caso de tener más de un parámetro los mismos deben ir separados por coma):

```
fun mostrarMensaje(mensaje: String) {
    println("*****")
    println(mensaje)
    println("*****")
}
```

Un parámetro podemos imaginarlo como una variable que solo se puede utilizar dentro de la función.

Ahora cuando llamamos a la función `mostrarMensaje` desde la `main` de nuestro programa debemos pasar una variable `String` o un valor de tipo `String`:

```
mostrarMensaje("El programa calcula la suma de dos valores ingresados por teclado.")
```

El `String` que le pasamos: "El programa calcula la suma de dos valores ingresados por teclado." lo recibe el parámetro de la función.

Una función con parámetros nos hace más flexible la misma para utilizarla en distintas circunstancias. En nuestro problema la función `mostrarMensaje` la utilizamos tanto para la presentación inicial de nuestro programa como para mostrar el mensaje de despedida. Si no existieran los parámetros estaríamos obligados a implementar dos funciones como el concepto anterior.

## Problema 2

Confeccionar una función que reciba tres enteros y nos muestre el mayor de ellos. La carga de los valores hacerlo por teclado en la función `main`.

Proyecto75 - Principal.kt

```
fun mostrarMayor(v1: Int, v2: Int, v3: Int) {
    print("Mayor:")
    if (v1 > v2 && v1 > v3)
        println(v1)
    else
        if (v2 > v3)
            print(v2)
        else
            print(v3)
}

fun main(parametro: Array<String>) {
    print("Ingrese primer valor:")
    val valor1 = readln().toInt()
    print("Ingrese segundo valor:")
    val valor2 = readln().toInt()
    print("Ingrese tercer valor:")
    val valor3 = readln().toInt()
    mostrarMayor(valor1, valor2, valor3)
}
```

Es importante notar que un programa en Kotlin no se ejecuta en forma lineal las funciones definidas en el archivo `*.kt` sino que arranca en la función `main`.

En la función `main` se solicita el ingreso de tres enteros por teclado y llama a la función `mostrarMayor` y le pasa a sus parámetros las tres variables enteras `valor1`, `valor2` y `valor3`.

La función `mostrarMayor` recibe en sus parámetros `v1`, `v2` y `v3` los valores cargados en las variables `valor1`, `valor2` y `valor3`.

Los parámetros son la forma que nos permite comunicar la función `main` con la función `mostrarMayor`.

Dentro de la función `mostrarMayor` no podemos acceder a las variables `valor1`, `valor2` y `valor3` ya que son variables locales de la función `main`.

## Problema 3

Desarrollar un programa que permita ingresar el lado de un cuadrado. Luego preguntar si quiere calcular y mostrar su perímetro o su superficie.

Proyecto76 - Principal.kt

```
fun mostrarPerimetro(lado: Int) {
    val perimetro = lado * 4
    println("El perímetro es $perimetro")
}

fun mostrarSuperficie(lado: Int) {
    val superficie = lado * lado
    println("La superficie es $superficie")
}

fun main(parametro: Array<String>) {
    print("Ingrese el valor del lado de un cuadrado:")
    val la = readln().toInt()
    print("Quiere calcular el perimetro o la superficie[ingresar texto: perimetro o superficie]:")
    var respuesta = readln()
    when (respuesta){
        "perimetro" -> mostrarPerimetro(la)
        "superficie" -> mostrarSuperficie(la)
    }
}
```

Definimos dos funciones que calculan y muestran el perímetro por un lado y por otro la superficie:

```
fun mostrarPerimetro(lado: Int) {
    val perimetro = lado * 4
    println("El perímetro es $perimetro")
}

fun mostrarSuperficie(lado: Int) {
    val superficie = lado * lado
    println("La superficie es $superficie")
}
```

En la función main cargamos el lado del cuadrado e ingresamos un String que indica que cálculo deseamos realizar si obtener el perímetro o la superficie. Una vez que se ingreso la variable respuesta procedemos a llamar a la función que efectúa el calculo respectivo pasando como dato la variable local "la" que almacena el valor del lado del cuadrado.

Los parámetros son la herramienta fundamental para pasar datos cuando hacemos la llamada a una función.

## Problemas propuestos

- En la función main solicitar que se ingrese una clave dos veces por teclado. Desarrollar una función que reciba dos String como parametros y muestre un mensaje si las dos claves ingresadas son iguales o distintas.
- Confeccionar una función que reciba tres enteros y los muestre ordenados de menor a mayor. En la función main solicitar la carga de 3 enteros por teclado y proceder a llamar a la primer función definida.

#### Proyecto77.kt

```
fun verificarClaves(clave1: String, clave2:String) {
    if (clave1 == clave2)
        println("Se ingresaron las dos veces la misma clave")
    else
        print("No se ingresó las dos veces con el mismo valor")
}

fun main(parametro: Array<String>) {
    print("Ingrese primer clave;")
    val clave1 = readln()
    print("Repita el ingreso de la misma clave;")
    val clave2 = readln()
    verificarClaves(clave1, clave2)
}
```

#### Proyecto78.rb

```
fun ordenadrMayorMenor(valor1: Int, valor2: Int, valor3: Int) {
    when {
        valor1 < valor2 && valor1 < valor3 -> {
            if (valor2 < valor3)
                println("$valor1 $valor2 $valor3 ")
            else
                println("$valor1 $valor3 $valor2 ")
        }
        valor2 < valor3 -> {
            if (valor1 < valor3)
                println("$valor2 $valor1 $valor3")
            else
                println("$valor2 $valor3 $valor1")
        }
        else -> {
            if (valor1 < valor2)
                println("$valor3 $valor1 $valor2")
            else
                println("$valor3 $valor2 $valor1")
        }
    }
}

fun main(parametros: Array<String>) {
    print("Ingrese primer valor:")
    val valor1 = readln().toInt()
    print("Ingrese segndo valor:")
    val valor2 = readln().toInt()
    print("Ingrese tercer valor:")
    val valor3 = readln().toInt()
    ordenadrMayorMenor(valor1, valor2, valor3)
}
```