

# 14 - Concepto de funciones

Hasta ahora hemos trabajado resolviendo todo el problema en la función main propuesta en Kotlin.

Esta forma de organizar un programa solo puede ser llevado a cabo si el mismo es muy pequeño (decenas de líneas)

Ahora buscaremos dividir o descomponer un problema complejo en pequeños problemas. La solución de cada uno de esos pequeños problemas nos trae la solución del problema complejo.

En Kotlin el planteo de esas pequeñas soluciones al problema complejo se hace dividiendo el programa en funciones. Una función es un conjunto de instrucciones en Kotlin que resuelven un problema específico.

Veamos ahora como crear nuestras propias funciones.

Los primeros problemas que presentaremos nos puede parecer que sea más conveniente resolver todo en la función main en vez de dividirlo en pequeña funciones. A medida que avancemos veremos que si un programa empieza a ser más complejo (cientos de líneas, miles de líneas o más) la división en pequeñas funciones nos permitirá tener un programa más ordenado y fácil de entender y por lo tanto en mantener.

## 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.

Implementar estas actividades en tres funciones.

Proyecto70 - Principal.kt

```

fun presentacion() {
    println("Programa que permite cargar dos valores por teclado.")
    println("Efectua la suma de los valores")
    println("Muestra el resultado de la suma")
    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 finalizacion() {
    println("*****")
    println("Gracias por utilizar este programa")
}

fun main(parametro: Array<String>) {
    presentacion()
    cargarSumar()
    finalizacion()
}

```

La forma de organizar nuestro programa cambia en forma radical.

El programa en Kotlin siempre comienza en la función main.

Es decir que el programa comienza en:

```

fun main(parametro: Array<String>) {
    presentacion()
    cargarSumar()
    finalizacion()
}

```

Como podemos ver en la función main llamamos a la tres funciones que declaramos previamente.

La sintaxis para declarar una función es mediante la palabra clave fun seguida por el nombre de la función (el nombre de la función no puede tener espacios en blanco ni comenzar con un número)

Luego del nombre de la función deben ir entre paréntesis los datos que llegan, si no llegan datos como es el caso de nuestras tres funciones solo se disponen paréntesis abierto y cerrado.

Todo el bloque de la función se encierra entre llaves y se indenta cuatro espacios como venimos trabajando con la función main.

Dentro de una función implementamos el algoritmo que pretendemos que resuelva esa función, por ejemplo la función presentacion tiene por objetivo mostrar en pantalla el objetivo del programa:

```

fun presentacion() {
    println("Programa que permite cargar dos valores por teclado.")
    println("Efectua la suma de los valores")
    println("Muestra el resultado de la suma")
    println("*****")
}

```

La función cargarSumar permite ingresar dos enteros por teclado, sumarlos y mostrarlos en pantalla:

```

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")
}

```

La función finalizacion() tiene por objetivo mostrar un mensaje que informe al operador que el programa finalizó:

```

fun finalizacion() {
    println("*****")
    println("Gracias por utilizar este programa")
}

```

Luego de definir las funciones tenemos al final de nuestro archivo Principal.kt las llamadas de las funciones dentro de la función main:

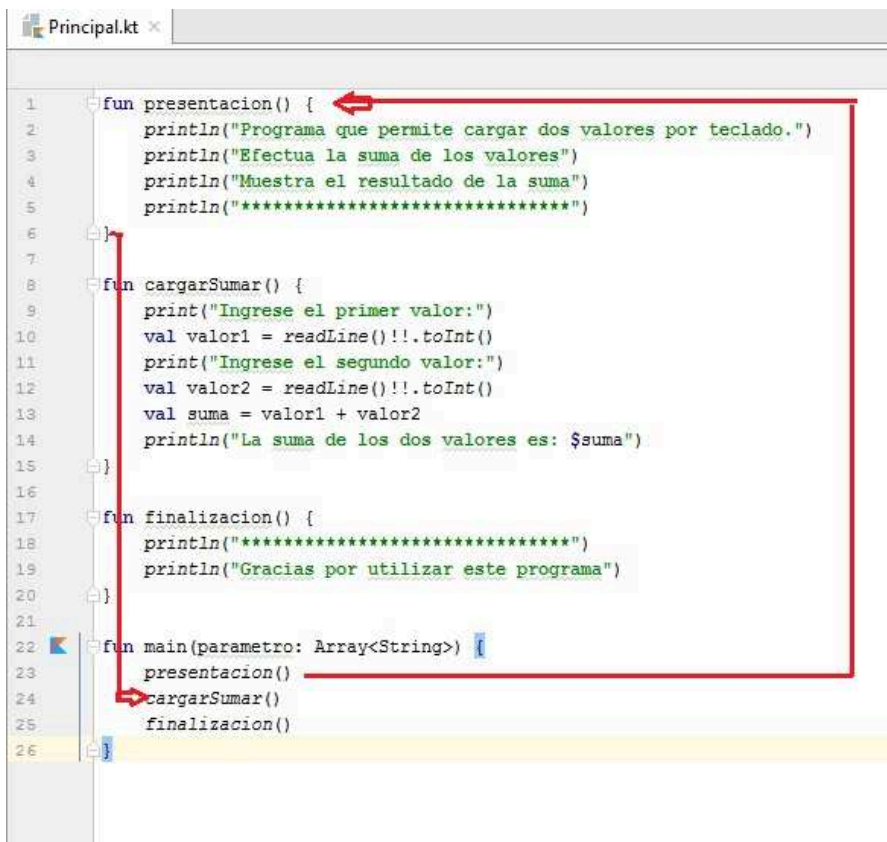
```

fun main(parametro: Array<String>) {
    presentacion()
    cargarSumar()
    finalizacion()
}

```

Si no hacemos las llamadas a las funciones los algoritmos que implementan las funciones nunca se ejecutarán.

Cuando en el bloque del programa principal se llama una función hasta que no finalice no continua con la llamada a la siguiente función:



```

1 fun presentacion() {
2     println("Programa que permite cargar dos valores por teclado.")
3     println("Efectua la suma de los valores")
4     println("Muestra el resultado de la suma")
5     println("*****")
6 }
7
8 fun cargarSumar() {
9     print("Ingrese el primer valor:")
10    val valor1 = readLine()!!.toInt()
11    print("Ingrese el segundo valor:")
12    val valor2 = readLine()!!.toInt()
13    val suma = valor1 + valor2
14    println("La suma de los dos valores es: $suma")
15 }
16
17 fun finalizacion() {
18     println("*****")
19     println("Gracias por utilizar este programa")
20 }
21
22 fun main(parametro: Array<String>) {
23     presentacion()
24     cargarSumar()
25     finalizacion()
26 }

```

En Kotlin la función main en realidad puede estar definida al principio del archivo y luego las otras funciones, lo que es importante decir que siempre un programa en Kotlin comienza a ejecutarse en la función main.

## Problema 2

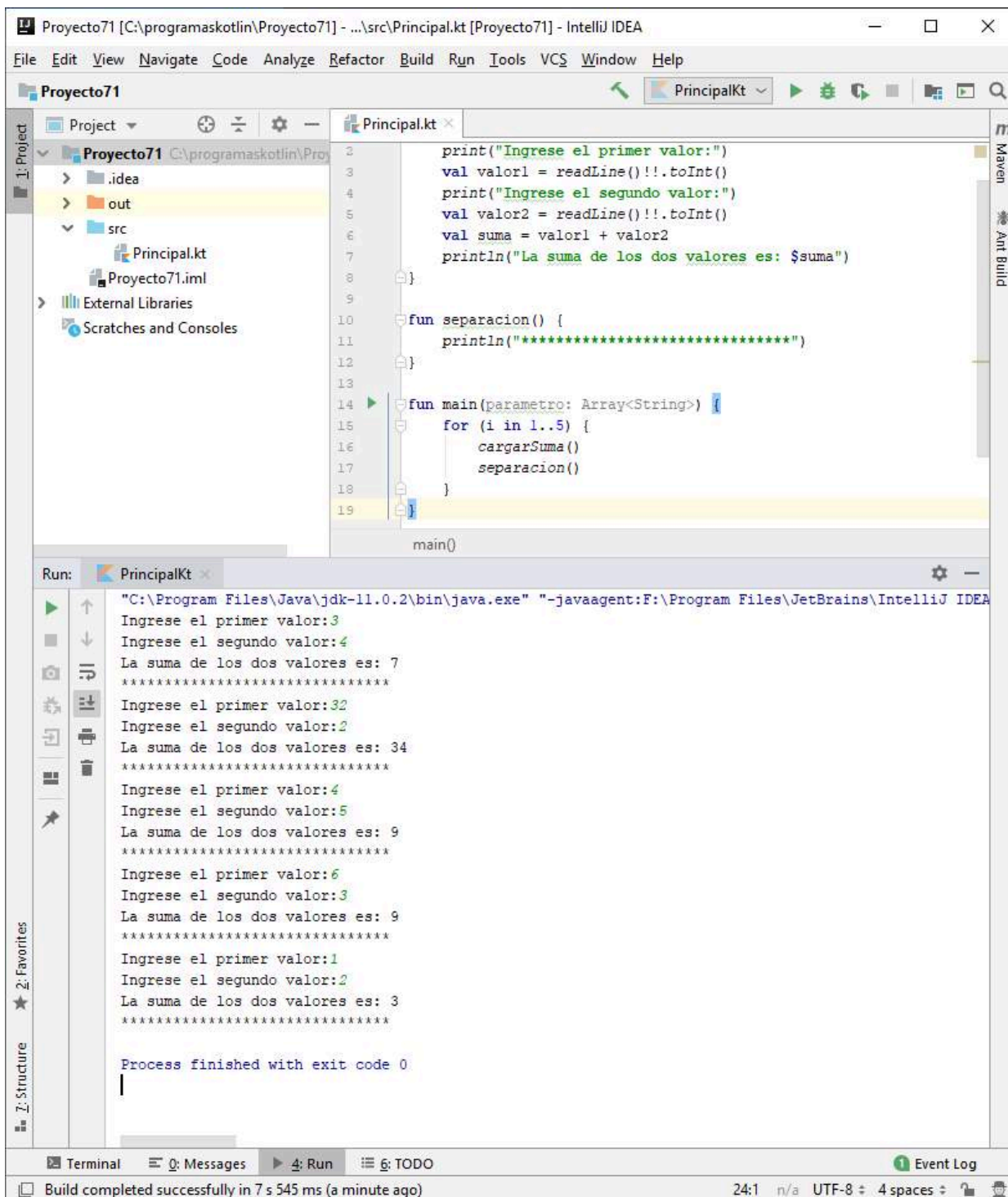
Confeccionar una aplicación que solicite la carga de dos valores enteros y muestre su suma.

Repetir la carga e impresión de la suma 5 veces.

Mostrar una línea separadora después de cada vez que cargamos dos valores y su suma.

Proyecto71 - Principal.kt

```
fun cargarSuma() {  
    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 separacion() {  
    println("*****")  
}  
  
fun main(parametro: Array<String>) {  
    for (i in 1..5) {  
        cargarSuma()  
        separacion()  
    }  
}
```



Hemos declarado dos funciones, una que permite cargar dos enteros sumarlos y mostrar el resultado:

```
fun cargarSuma() {  
    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")  
}
```

Y otra función que tiene por objetivo mostrar una línea separadora con asteriscos:

```
fun separacion() {  
    println("*****")  
}
```

Desde la función main llamamos a las funciones de cargarSumar y separación 5 veces:

```
fun main(parametro: Array<String>) {  
    for (i in 1..5) {  
        cargarSuma()  
        separacion()  
    }  
}
```

Lo nuevo que debe quedar claro es que la llamada a las funciones desde la función main de nuestro programa puede hacerse múltiples veces.

## Problemas propuestos

- Desarrollar un programa con dos funciones. La primer solicite el ingreso de un entero y muestre el cuadrado de dicho valor. La segunda que solicite la carga de dos valores y muestre el producto de los mismos. LLamar desde la main a ambas funciones.
- Desarrollar una función que solicite la carga de tres valores y muestre el menor. Desde la función main del programa llamar 2 veces a dicha función (sin utilizar una estructura repetitiva)

## Proyecto72

```
fun calculaCuadrado() {
    print("Ingrese un entero:")
    val valor = readln().toInt()
    val cuadrado = valor * valor
    println("El cuadrado es $cuadrado")
}

fun calcularProducto() {
    print("Ingrese primer valor:")
    val valor1 = readln().toInt()
    print("Ingrese segundo valor:")
    val valor2 = readln().toInt()
    val producto = valor1 * valor2
    println("El producto de los valores es: $producto")
}

fun main(parametro: Array<String>) {
    calculaCuadrado()
    calcularProducto()
}
```

## Proyecto73

```
fun menorValor() {
    print("Ingrese primer valor:")
    val valor1 = readln().toInt()
    print("Ingrese segundo valor:")
    val valor2 = readln().toInt()
    print("Ingrese tercer valor:")
    val valor3 = readln().toInt()
    print("Menor de los tres:")
    when {
        valor1 < valor2 && valor1 < valor3 -> println(valor1)
        valor2 < valor3 -> println(valor2)
        else -> println(valor3)
    }
}

fun main(parametro: Array<String>) {
    menorValor()
    menorValor()
}
```