

# Kotlin tabla ejercicios 2

## Ejercicio 1:

**Sumar los números del 1 al 100 usando un bucle `for`**

Escribe una función llamada `sumarNumeros` que utilice un bucle `for` para sumar todos los números del 1 al 100. Luego, imprime el resultado.

**Pista:** Usa un bucle `for` con un rango de 1 a 100.

```
fun sumarNumeros(){
    var suma = 0
    for (i in 1..100){
        suma += i
    }
    println("La suma de los números del 1 al 100 es $suma")
}
sumarNumeros()
```

## Ejercicio 2:

**Contar números pares entre 1 y 50 con un bucle `while`**

Crea una función llamada `contarNumerosPares` que utilice un bucle `while` para contar cuántos números pares hay entre 1 y 50. Imprime el resultado.

**Pista:** Incrementa un contador solo cuando el número sea par.

```
fun contarNumerosPares(){
    var contadorPares = 0
    var numero = 1
    while (numero<50){
        numero++
        if (numero%2 == 0){
            contadorPares++
        }
    }
    println("Hay $contadorPares números pares del 1 al 50")
}
contarNumerosPares()
```

### Ejercicio 3:

#### Mostrar un menú hasta que el usuario seleccione salir (bucle `do-while`)

Crea una función llamada `mostrarMenu` que imprima un menú simple con tres opciones:

1. Sumar dos números.
2. Restar dos números.
3. Salir.

El menú debe repetirse hasta que el usuario elija la opción "Salir". Utiliza un bucle `do-while` para esto.

**Pista:** Usa la clase `Scanner` para leer las entradas del usuario.

```
fun isInteger(opcion : String) : Boolean{
    return try {
        opcion.toInt()
        true
    } catch (e: NumberFormatException) {
        false
    }
}
```

```

fun mostrarMenu(){
    val sc = Scanner(System.`in`)
    var opcion : String

    do {
        println("\nElige una de estas opciones: ")
        println("1. Sumar dos números")
        println("2. Restar dos números")
        println("3. Salir")
        opcion = sc.nextLine()
        if (isInteger(opcion)){
            when(opcion.toInt()){
                1 -> {
                    println("Introduce el primer número:")
                    val num1 : Int = sc.nextInt()
                    println("Introduce el segundo número:")
                    val num2 : Int = sc.nextInt()
                    println("La suma es: ${num1 + num2}")
                    sc.nextLine()
                }
                2 -> {
                    println("Introduce el primer número:")
                    val num1 : Int = sc.nextInt()
                    println("Introduce el segundo número:")
                    val num2 : Int = sc.nextInt()
                    println("La resta es: ${num1 - num2}")
                    sc.nextLine()
                }
                3 -> {}
            }
            else -> println("La opción seleccionada no se reconoce, vuelve a intentarlo")
        }
        else{
            println("La opción seleccionada no se reconoce, vuelve a intentarlo")
        }
    }while (opcion.toInt() != 3)

    println("Adiós!")
}
mostrarMenu()

```

## Ejercicio 4:

### Imprimir una tabla de multiplicar usando un bucle for anidado

Escribe una función llamada `imprimirTablaMultiplicar` que imprima la tabla de multiplicar del 1 al 10 utilizando bucles `for` anidados.

Pista: Un bucle controla el número base, y el otro controla los multiplicadores.

```
fun imprimirTablaMultiplicar(){
    println("Tablas de multiplicar:")
    for (i in 1 .. 10){
        println("\n_____ \nTABLA DEL $i\n-----")
        for (j in 1..10){
            println("$i*$j = ${i*j}")
        }
    }
}
imprimirTablaMultiplicar()
```

## Ejercicio 5:

### Invertir una cadena con un bucle `for`

Escribe una función llamada `invertirCadena` que reciba una cadena y la invierta utilizando un bucle `for`.

**Pista:** Recorre la cadena desde el final hasta el principio.

```
fun invertirCadena(){
    val sc = Scanner(System.`in`)
    var cadena : String
    var cadenaInvertida = ""

    println("Dame una cadena para invertirla: ")
    cadena = sc.nextLine()

    //EJEMPLO: Dabale arroz a la zorra el abad
    for (i in cadena.length - 1 downTo 0){
        cadenaInvertida += cadena[i]
    }
    println(cadenaInvertida)
}
invertirCadena()
```

## Ejercicio 6:

### Encontrar el primer número múltiplo de 7 mayor que 100 con un bucle `while`

Crea una función llamada `primerMultiploDeSiete` que encuentre el primer número múltiplo de 7 mayor que 100 utilizando un bucle `while`. Imprime el resultado.

```
fun primerMultiploDeSiete(){
    var numero : Int = 101

    while (numero%7 != 0){
        numero++
    }
    println("El primer múltiplo de 7 mayor que 100 es $numero")
}
primerMultiploDeSiete()
```

## Ejercicio 7:

### Controlar un bucle con `break` y `continue`

Crea una función llamada `bucleConControl` que recorra los números del 1 al 10:

- Si el número es divisible por 3, usa `continue` para saltarte esa iteración.
- Si el número es mayor que 8, usa `break` para salir del bucle.

Imprime los números que no se salten ni rompan el bucle.

```
fun bucleConControl(){
    var numero : Int = 1
    for (i in 1..10){
        if(numero>8){
            println("$numero -Bucle roto al ser mayor que 8")
            numero++
            break
        }else if (numero%3 == 0){
            println("$numero -Numero multiplo de 3")
            numero++
            continue
        }else{
            println("$numero -Numero normal")
            numero++
        }
    }
}
bucleConControl()
```