

9 - Estructura repetitiva while

Hasta ahora hemos empleado estructuras SECUENCIALES y CONDICIONALES. Existe otro tipo de estructuras tan importantes como las anteriores que son las estructuras REPETITIVAS.

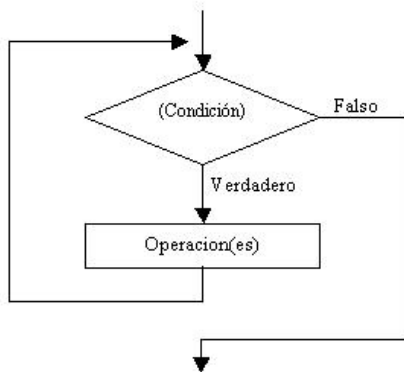
Una estructura repetitiva permite ejecutar una instrucción o un conjunto de instrucciones varias veces.

Una ejecución repetitiva de sentencias se caracteriza por:

- La o las sentencias que se repiten.
- El test o prueba de condición antes de cada repetición, que motivará que se repitan o no las sentencias.

Estructura repetitiva while.

Representación gráfica de la estructura while:



No debemos confundir la representación gráfica de la estructura repetitiva while (Mientras) con la estructura condicional if (Si)

Funcionamiento: En primer lugar se verifica la condición, si la misma resulta verdadera se ejecutan las operaciones que indicamos por la rama del Verdadero.

A la rama del verdadero la graficamos en la parte inferior de la condición. Una línea al final del bloque de repetición la conecta con la parte superior de la estructura repetitiva.

En caso que la condición sea Falsa continúa por la rama del Falso y sale de la estructura repetitiva para continuar con la ejecución del algoritmo.

El bloque se repite MIENTRAS la condición sea Verdadera.

Importante: Si la condición siempre retorna verdadero estamos en presencia de un ciclo repetitivo infinito. Dicha situación es un error de programación, nunca finalizará el programa.

Problema 1

Realizar un programa que imprima en pantalla los números del 1 al 100.

Proyecto31 - Principal.kt

```
fun main(parametro: Array<String>) {  
    var x = 1  
    while (x <= 100) {  
        println(x)  
        x = x + 1  
    }  
}
```

Lo primero que podemos hacer notar que definimos una variable mutable de tipo Int con el valor 1:

```
var x = 1
```

Esta variable x se irá modificando dentro del while por eso la necesidad de definirla con la palabra clave var.

La estructura repetitiva while siempre tiene una condición, si se verifica verdadera pasa a ejecutar el bloque de instrucciones encerradas entre llaves (si no hay llaves solo ejecuta la siguiente instrucción del while a modo similar del if)

Cada vez que se ejecuta el bloque de instrucciones del while vuelve a verificarse la condición para ver si repite nuevamente o corta el ciclo repetitivo.

En nuestro problema x comienza con el valor 1, al ejecutarse la condición retorna VERDADERO porque el contenido de x (1) es menor o igual a 100. Al ser la condición verdadera se ejecuta el bloque de instrucciones que contiene la estructura while. El bloque de instrucciones contiene una salida y una operación.

Se imprime el contenido de x, y seguidamente se incrementa la variable x en uno.

La operación $x = x + 1$ se lee como "en la variable x se guarda el contenido de x más 1". Es decir, si x contiene 1 luego de ejecutarse esta operación se almacenará en x un 2.

Al finalizar el bloque de instrucciones que contiene la estructura repetitiva se verifica nuevamente la condición de la estructura repetitiva y se repite el proceso explicado anteriormente.

Mientras la condición retorne verdadero se ejecuta el bloque de instrucciones; al retornar falso la verificación de la condición se sale de la estructura repetitiva y continua el algoritmo, en este caso finaliza el programa.

Lo más difícil es la definición de la condición de la estructura while y qué bloque de instrucciones se van a repetir. Observar que si, por ejemplo, disponemos la condición $x \geq 100$ (si x es mayor o igual a 100) no provoca ningún error sintáctico pero estamos en presencia de un error lógico porque al evaluarse por primera vez la condición retorna falso y no se ejecuta el bloque de instrucciones que queríamos repetir 100 veces.

No existe una RECETA para definir una condición de una estructura repetitiva, sino que se logra con una práctica continua solucionando problemas.

Problema 2

Escribir un programa que solicite la carga de un valor positivo y nos muestre desde 1 hasta el valor ingresado de uno en uno.

Ejemplo: Si ingresamos 30 se debe mostrar en pantalla los números del 1 al 30.

Proyecto32 - Principal.kt

```

fun main(parametro: Array<String>) {
    print("Ingrese un valor:")
    val valor = readln().toInt()
    var x = 1
    while (x <= valor) {
        println(x)
        x = x + 1
    }
}

```

La variable x recibe el nombre de CONTADOR. Un contador es un tipo especial de variable que se incrementa o disminuye con valores constantes durante la ejecución del programa.

El contador x nos indica en cada momento la cantidad de valores impresos en pantalla.

Problema 3

Desarrollar un programa que permita la carga de 10 valores por teclado y nos muestre posteriormente la suma de los valores ingresados y su promedio.

Proyecto33 - Principal.kt

```

fun main(parametro: Array<String>) {
    var x = 1
    var suma = 0
    while (x <= 10) {
        print("Ingrese un valor:")
        val valor=readln().toInt()
        suma = suma + valor
        x = x + 1
    }
    println("La suma de los 10 valores ingresados es $suma")
    val promedio = suma / 10
    println("El promedio es $promedio")
}

```

En este problema, a semejanza de los anteriores, llevamos un CONTADOR llamado x que nos sirve para contar las vueltas que debe repetir el while.

También aparece el concepto de ACUMULADOR (un acumulador es un tipo especial de variable que se incrementa o disminuye con valores variables durante la ejecución del programa)

Hemos dado el nombre de suma a nuestro acumulador. Cada ciclo que se repita la estructura repetitiva, la variable suma se incrementa con el contenido ingresado en la variable valor.

Problema 4

Una planta que fabrica perfiles de hierro posee un lote de n piezas.

Confeccionar un programa que pida ingresar por teclado la cantidad de piezas a procesar y luego ingrese la longitud de cada perfil; sabiendo que la pieza cuya longitud esté comprendida en el rango de 1.20 y 1.30 son aptas. Imprimir por pantalla la cantidad de piezas aptas que hay en el lote.

Proyecto34 - Principal.kt

```

fun main(parametro: Array<String>) {
    print("Cuántas piezas procesará:")
    val n = readln().toInt()
    var x = 1
    var cantidad = 0
    while (x <= n) {
        print("Ingrese la medida de la pieza:")
        val largo = readln().toDouble()
        if (largo >= 1.20 && largo <= 1.30)
            cantidad = cantidad + 1
        x = x + 1;
    }
    print("La cantidad de piezas aptas son: $cantidad")
}

```

Podemos observar que dentro de una estructura repetitiva puede haber estructuras condicionales (inclusive puede haber otras estructuras repetitivas que veremos más adelante)

En este problema hay que cargar inicialmente la cantidad de piezas a ingresar (*n*), seguidamente se cargan *n* valores de largos de piezas dentro del while.

Cada vez que ingresamos un largo de pieza (*largo*) verificamos si es una medida correcta (debe estar entre 1.20 y 1.30 el largo para que sea correcta), en caso de ser correcta la CONTAMOS (incrementamos la variable cantidad en 1)

Al contador cantidad lo inicializamos en cero porque inicialmente no se ha cargado ningún largo de medida.

Cuando salimos de la estructura repetitiva porque se han cargado *n* largos de piezas mostramos por pantalla el contador cantidad (que representa la cantidad de piezas aptas)

En este problema tenemos dos CONTADORES:

<i>x</i>	(Cuenta la cantidad de piezas cargadas hasta el momento)
<i>cantidad</i>	(Cuenta los perfiles de hierro aptos)

Problemas propuestos

- Escribir un programa que solicite ingresar 10 notas de alumnos y nos informe cuántos tienen notas mayores o iguales a 7 y cuántos menores.
- Se ingresan un conjunto de *n* alturas de personas por teclado (*n* se ingresa por teclado). Mostrar la altura promedio de las personas.
- En una empresa trabajan *n* empleados cuyos sueldos oscilan entre \$100 y \$500, realizar un programa que lea los sueldos que cobra cada empleado e informe cuántos empleados cobran entre \$100 y \$300 y cuántos cobran más de \$300. Además el programa deberá informar el importe que gasta la empresa en sueldos al personal.
- Realizar un programa que imprima 25 términos de la serie 11 - 22 - 33 - 44, etc. (No se ingresan valores por teclado)
- Mostrar los múltiplos de 8 hasta el valor 500. Debe aparecer en pantalla 8 - 16 - 24, etc.
- Realizar un programa que permita cargar dos listas de 5 valores cada una. Informar con un mensaje cual de las dos listas tiene un valor acumulado mayor (mensajes "Lista 1 mayor", "Lista 2 mayor", "Listas iguales")
Tener en cuenta que puede haber dos o más estructuras repetitivas en un algoritmo.

- Desarrollar un programa que permita cargar n números enteros y luego nos informe cuántos valores fueron pares y cuántos impares.

Emplear el operador "%" en la condición de la estructura condicional:

```
if (valor % 2 == 0)           //Si el if se verifica verdadero luego es par.
```

Proyecto35

```
fun main(parametro: Array<String>) {
    var x = 1
    var conta1 = 0
    var conta2 = 0
    while (x <= 10) {
        print("Ingrese nota:")
        val nota = readln().toInt()
        if (nota >= 7)
            conta1 = conta1 + 1
        else
            conta2 = conta2 + 1
        x = x + 1
    }
    println("Cantidad de alumnos con notas mayores o iguales a 7: $conta1")
    println("Cantidad de alumnos con notas menores a 7: $conta2")
}
```

Proyecto36

```
fun main(parametro: Array<String>) {
    print("Cuantas alturas ingresará?:")
    val n = readln().toInt()
    var x = 1
    var suma = 0.0
    while (x <= n) {
        print("Ingrese la altura de la persona(Ej:1.76) :")
        val altura = readln().toDouble()
        suma = suma + altura
        x = x + 1
    }
    val promedio = suma / n
    println("Altura promedio: $promedio")
}
```

Proyecto37

```
fun main(parametro: Array<String>) {
    print("Cuantos empleados tiene la empresa:")
    val n = readln().toInt()
    var x = 1
    var conta1 = 0
    var conta2 = 0
    var gastos = 0.0
    while (x <= n) {
        print("Ingrese el sueldo del empleado:")
        val sueldo = readln().toDouble()
        if (sueldo <= 300)
            conta1 = conta1 + 1
        else
            conta2 = conta2 + 1
        gastos = gastos + sueldo;
        x = x + 1
    }
    println("Cantidad de empleados con sueldos entre 100 y 300: $conta1")
    println("Cantidad de empleados con sueldos mayor a 300: $conta2")
    System.out.print("Gastos total de la empresa en sueldos: $gastos")
}
```

Proyecto38

```
fun main(parametro: Array<String>) {  
    var termino = 11  
    var x = 1  
    while (x <= 25) {  
        println(termino)  
        termino = termino + 11  
        x = x + 1  
    }  
}
```

Proyecto39

```
fun main(parametro: Array<String>) {  
    var mult8 = 8  
    while (mult8 <= 500) {  
        print("$mult8 -")  
        mult8 = mult8 + 8  
    }  
}
```

Proyecto40

```
fun main(parametro: Array<String>) {  
    var x = 1  
    var sumal = 0  
    println("Ingreso de la primer lista de valores")  
    while (x <= 5) {  
        print("Ingrese valor:")  
        val valor=readln().toInt()  
        sumal = sumal + valor  
        x = x + 1  
    }  
    println("Ingreso de la segunda lista de valores")  
    x = 1  
    var suma2 = 0  
    while (x <= 5) {  
        print("Ingrese valor:")  
        val valor=readln().toInt()  
        suma2 = suma2 + valor  
        x = x + 1  
    }  
    if (sumal > suma2)  
        print("Lista 1 mayor.")  
    else  
        if (suma2 > sumal)  
            print("Lista2 mayor.")  
        else  
            print("Listas iguales.")  
}
```

Proyecto41

```
fun main(parametro: Array<String>) {  
    var pares = 0  
    var impares = 0  
    print("Cuantos números ingresará:")  
    val n = readln().toInt()  
    var x = 1
```

```
while (x <= n) {  
    print("Ingrese el valor:")  
    val valor = readln().toInt()  
    if (valor % 2 == 0)  
        pares = pares + 1  
    else  
        impares = impares + 1  
    x = x + 1  
}  
println("Cantidad de pares: $pares")  
println("Cantidad de impares: $impares");  
}
```