



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

## INTRODUCCIÓN A KOTLIN PARTE 2

### ESTRUCTURA DE CONTROL

Aquí trabajaremos en las estructuras de control específicamente con: Condiciones simples, dobles y anidadas. Las estructuras de control van a permitir modificar el flujo de ejecución de un programa.

- **Secuencial**

```
1.8.21 ▾ JVM ▾ Program arguments 🔗 <> ▶

/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var num1=5.0
    var num2=5.0
    var num3=4.0
    var prom=(num1+num2+num3)/3
    println("El promedio es $prom")
    println("%.1f".format(prom))
    println("%.2f".format(prom))
    println("%.3f".format(prom))
}

El promedio es 4.666666666666667
4.7
4.67
4.667
```

### Tips ¡!

- Para reducir o aumentar los decimales de las cifras podemos trabajar con:  
“%.*(#decimales)*f”.format(*Variable*)  
“%.5f”.format(*prom*)

- **Selección o condicionales**

- If
- If else
- Else if
- When



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

- Condicional simple

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var num1=5.0
    var num2=5.0
    var num3=4.0
    var prom=(num1+num2+num3)/3
    println("El promedio es $prom")
    //println("%.1f".format(prom))
    //
    if (prom>3.5){
        println("El promedio es superior a 3.5 APRUEBA la materia")
    }
    println("Termina el proceso")

|
}
```

```
El promedio es 4.666666666666667
El promedio es superior a 3.5 APRUEBA la materia
Termina el proceso
```

- Condicional doble

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var num1=2.0
    var num2=3.0
    var num3=4.0
    var prom=(num1+num2+num3)/3
    println("El promedio es $prom")
    //println("%.1f".format(prom))
    //
    if (prom>3.5){
        println("El promedio es superior a 3.5 APRUEBA la materia")
    }
    else {
        println ("El promedio es menor a 3.5 NO APRUEBA y lo van a regañar!")
    }

}

}
```

```
El promedio es 3.0
El promedio es menor a 3.5 NO APRUEBA y lo van a regañar!
```



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

- Condicional doble anidada

1.8.21 JVM Program arguments Copy link Share code Run

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var num1=1.0
    var num2=2.0
    var num3=2.0
    var prom=(num1+num2+num3)/3
    println("El promedio es $prom")
    //println("%.1f".format(prom))
    //
    /*** Desarrollar un programa que valide cuando:
    // 1. Si es promedio es mayor o igual a 3.5 GANA
    // 2. Si el promedio está entre 2.0 y 3.5 Puede recuperar
    // 3. Si el promedio es menor a 2.0 PIERDE sin posibilidad de recuperar
    //
    if (prom>=3.5){
        println("El promedio es superior a 3.5 APRUEBA la materia")
    }
    else if (prom>=2.0 && prom<=3.5){
        println("El promedio está en el rango permitido y puede recuperar")
    }
    else {
        println ("El promedio es inferior a 2.0, no puede recuperar y PIERDE LA MATERIA")
    }
}

El promedio es 1.6666666666666667
El promedio es inferior a 2.0, no puede recuperar y PIERDE LA MATERIA
```

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var num1=1.0
    var num2=2.0
    var num3=2.0
    var prom=(num1+num2+num3)/3
    println("El promedio es $prom")
    //println("%.1f".format(prom))
    //
    /*** Desarrollar un programa que valide cuando:
    // 1. Si es promedio es mayor o igual a 3.5 GANA
    // 2. Si el promedio está entre 2.0 y 3.5 Puede recuperar
    // 3. Si el promedio es menor a 2.0 PIERDE sin posibilidad de recuperar
    //
    /** OTRA FORMA DE HACERLO CON OPERADOR DE RANGO**

    var a=2.0
    var b=3.5

    if (prom>=3.5){
        println("El promedio es superior a 3.5 APRUEBA la materia")
    }
    else if (prom in a..b){
        println("El promedio está en el rango permitido y puede recuperar")
    }
    else {
        println ("El promedio es inferior a 2.0, no puede recuperar y PIERDE LA MATERIA")
    }
}

El promedio es 1.6666666666666667
El promedio es inferior a 2.0, no puede recuperar y PIERDE LA MATERIA
```



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

- Selección múltiple con WHEN

Esta estructura nos sirve para determinar opciones múltiples

Ejemplo 1.

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var estacion=6
    when (estacion){
        1-> println("Inverno")
        2-> println("Primavera")
        3-> println("Verano")
        4-> println("Otoño")
        else -> println("No corresponde a ninguna estación")
    }
}
```

No corresponde a ninguna estación

Ejemplo 2.

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var mes=11
    when (mes){
        1,2,3-> println("Inverno")
        4,5,6-> println("Primavera")
        7,8,9-> println("Verano")
        10,11,12-> println("Otoño")
        else -> println("No corresponde a ninguna estación")
    }
}
```

Otoño



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

Ejemplo 3.

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    var mes=11
    when (mes){
        in 1..3-> {
            println("Inverno")
            println("Enero, Febrero o Marzo")
        }
        in 4..6-> {
            println("Primavera")
            println("Abril, Mayo o Junio")
        }
        in 7..9-> {
            println("Verano")
            println("Julio, Agosto o Septiembre")
        }
        in 10..12-> {
            println("Otoño")
            println("Octubre, Noviembre o Diciembre")
        }
        else -> println("No corresponde a ninguna estación")
    }
}
```

Otoño  
Octubre, Noviembre o Diciembre





# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

## ESTRUCTURA ITERATIVAS – CICLOS

Nos van a permitir determinar el flujo del sistema, pero iterativamente, es decir si queremos que un proceso se repita constantemente o hasta cierto punto, básicamente, el concepto es el mismo que con otros lenguajes de programación, repetir procesos hasta la instrucción de parada a diferencia del ciclo **FOR** que tiene un cambio considerable en este lenguaje.

- **Ciclo Do While**

Hace el proceso y luego pregunta si continua o no, es decir si se cumple la condición de parada.

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    // do while :: Repita mientras

    var i=0
    do {
        println("$i")
        i++ //Operador postfijo, nos indica que aumente de 1 en 1
    }while (i<=10) // Repitase mientras i sea menor o igual a 10
    }
```

```
0
1
2
3
4
5
6
7
8
9
10
```

- **Ciclo While**

Primero pregunta y luego hace el proceso.

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    // while :: mientras

    var i=0
    while (i<=10)// Repitase mientras i sea menor o igual a 10
    {
        println("$i")
        i++ //Operador postfijo, nos indica que aumente de 1 en 1
    }
    println("Termina el proceso")
    }
```

```
1
2
3
4
5
6
7
8
9
10
Termina el proceso
```



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

- Ciclo for

Ejemplo 1.

```
/**
 * Demostraciones para la guía 1. introducción a
 */
fun main() {

    //for :: para
    //rango de números

    for (i in 0..10){
        print("$i ")
    }
}
```

0 1 2 3 4 5 6 7 8 9 10

Ejemplo 2 - Step

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin Segunda parte
 */
fun main() {

    //for :: para
    //rango de números

    for (i in 0..10 step 3) // El Step nos permite saltar,
                           //por lo menos acá para números impares
    {
        print("$i ")
    }
}
```

0 3 6 9



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

## Ejemplo 3 - Step

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin Segunda par
 */
fun main() {

    //for :: para
    //rango de números

    for (i in 0..10 step 2) // EL Step nos permite saltar,
                           //por lo menos acá para números pares
    {
        print("$i ")
    }
}
```

0 2 4 6 8 10

## Ejemplo 4 – Until

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin Segunda par
 */
fun main() {

    //for :: para
    //rango de números

    for (i in 0 until 10) // EL Until nos permite descartar el
                        //ultimo número de la cadena
    {
        print("$i ")
    }
}
```

0 1 2 3 4 5 6 7 8 9

## Ejemplo 5 – DownTo

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin Segunda par
 */
fun main() {

    //for :: para
    //rango de números

    for (i in 10 downTo 0) // EL DownTo nos permite trabajar
                        //con decrementos
    {
        print("$i ")
    }
}
```

10 9 8 7 6 5 4 3 2 1 0





# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

## EJERCICIOS

1. En video play se hace un 50% de descuento a las personas cuya compra sea mayor a 100 mil pesos. Indicar el monto descontado y el descuento.
2. Hallar si un número es PAR o IMPAR.
3. A partir de un número, determinar si este número es múltiplo de 3 y 5
4. A partir de un número, en caso de que termine en cifra cero devolver la mitad, en caso contrario devolver el doble.
5. Ordenar 5 números enteros y devolver los números de forma ASCEDENTE y de forma DESCENDENTE.
6. Dado 3 longitudes que conforman los lados de un triángulo, devolver el tipo de triángulo de acuerdo a sus lados, partiendo de lo siguiente:
  - a. Equilátero – Tres lados iguales.
  - b. Isósceles – Dos lados iguales.
  - c. Escaleno – Tres lados diferentes
7. A partir de una nota de 0 a 10 para 5 materias, mostrar de la forma: Reprobado, Aprobado, Excelente.
8. Realizar un programa que pase de gramos a otras unidades de masa, mostrar en la pantalla de menú con las opciones posibles.
9. Ingresar la cantidad de personas que se encuentran en un supermercado. Debido a la pandemia, el aforo permitido es de 40% y su totalidad es de 200. El programa deberá indicar si el aforo es aceptable, esta a tope, ha sido sobrepasado.
10. Realizar un programa que registre dos números, y con ayuda de un menú, mostrar la respuesta de acuerdo a la operación básica seleccionada (suma, resta, multiplicación, división)
11. Construir un algoritmo que calcule el sueldo final de un empleado que se le paga \$30.000 por cada hora trabajada en el mes; así mismo, todos los empleados reciben un pago de gratificación de acuerdo a la cantidad de años que tiene trabajando en la empresa, como se muestra en el siguiente cuadro:

Años laborados en la empresa	% de gratificación
0 a 5	60%
6 a 10	80%
11 a más	100%

12. A partir de 10 números determinar cantidad de pares e impares sin contar múltiplos de 5.
13. Calcular la factorial de un número determinado.

// Cargar el archivo al repo creado en Github, nombrado así:::  
Ejercicios\_condi\_ciclos\_ "Nombre del aprendiz\_CC"