



Universidad Tecnológica de Nezahualcóyotl

Carrera: Ing. Desarrollo y Gestión de Software.

Asignatura: Desarrollo Móvil para Dispositivos Inteligentes.

Profesor: Ricardo Armando Machorro Reyes.

Trabajo: Programas vistos en clase del primer parcial.

Alumno: Márquez Sánchez Eder Germán.

Matricula: 221271016.

Grupo: ITIC – 901M.

Periodo: Septiembre / diciembre 2022 – 3.

Fecha de Entrega: Domingo 02 de octubre del 2022.

**SEMANA 1: VARIABLES, OPERADORES,
FUNCIONES, SOBRECARGA, CICLOS Y ARRAYS**

Lunes 05 / 09 / 2022.

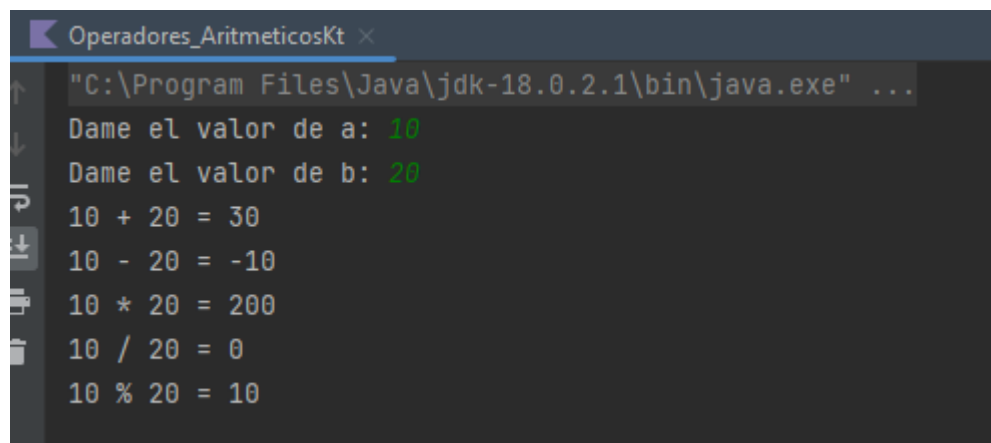
1. Operadores Aritméticos.

```
package Semanal.Lunes_05_09_2022

fun main(){
    print("Dame el valor de a: ")
    var a = readLine()!!.toInt()

    print("Dame el valor de b: ")
    var b = readLine()!!.toInt()

    println("$a + $b = ${a+b}")
    println("$a - $b = ${a-b}")
    println("$a * $b = ${a*b}")
    println("$a / $b = ${a/b}")
    println("$a % $b = ${a%b}")
}
```



```
Operadores_AritmeticosKt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Dame el valor de a: 10
Dame el valor de b: 20
10 + 20 = 30
10 - 20 = -10
10 * 20 = 200
10 / 20 = 0
10 % 20 = 10
```

2. Operadores de asignación.

```
package Semanal.Lunes_05_09_2022

fun main(){
    var a = 0
    var b = 10

    a = ++b //b = b + 1
    println("a = $a")
    println("b = $b")

    a = --b //b = b + 1
    println("a = $a")
    println("b = $b")

    a = b++
    println("a = $a")
}
```

```
println("b = $b")

a = b--
println("a = $a")
println("b = $b")
}
```

```
Run: Operadores_AsignacionKt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
a = 11
b = 11
a = 10
b = 10
a = 10
b = 11
a = 11
b = 10
Process finished with exit code 0
```

3. Variables.

```
package Semanal.Lunes_05_09_2022

const val HOLA = "Hola"
var MENSAJE = "ADIOS"

fun main() {
    println("Hola mundo")

    //immutable
    val x: Int = 12
    println(x)

    //Mutable
    var y: Int = 23
    println(y)

    y = 100
    println(y)
}
```

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...  
Hola mundo  
12  
23  
100  
  
Process finished with exit code 0
```

4. Variables por Teclado.

```
package Semanal.Lunes_05_09_2022  
  
fun main() {  
    print("Ingresa tu nombre: ")  
    val nombre = readLine()  
  
    print("Ingresa tu edad: ")  
    val edad = readLine()!!.toInt()  
  
    //Java  
    println("Hola "+nombre+" con edad de "+edad+" años")  
  
    //Kotlin  
    println("===String Templates===")  
    println("Hola $nombre con edad de $edad años")  
  
    println("===Raw String Templates===")  
    println("""  
        Hola $nombre con edad de $edad años  
        """)  
}
```

```
Variables_TecladoKt x  
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...  
Ingresa tu nombre: eder  
Ingresa tu edad: 20  
Hola eder con edad de 20 años  
===String Templates===  
Hola eder con edad de 20 años  
===Raw String Templates===  
  
    Hola eder con edad de 20 años
```

Miércoles 07 / 09 / 2022

1. Funciones con valores retornables.

```
package Semanal.Miercoles_07_09_2022

fun main(){
    println("Funcion con valores retornables")

    print("Dame el valor de a: ")
    var a = readLine()!!.toInt()

    print("Dame el valor de b: ")
    var b = readLine()!!.toInt()

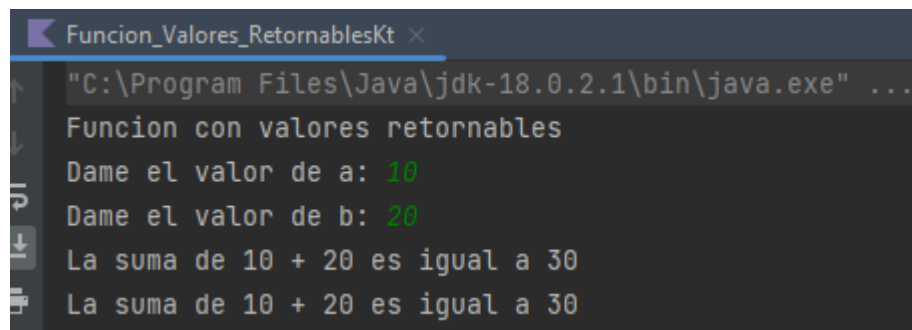
    var c = suma(a, b)

    var c1 = suma1(a,b)

    println("La suma de $a + $b es igual a $c")
    println("La suma de $a + $b es igual a $c1")
}

fun suma(a: Int, b: Int): Int{
    return a+b
}

fun suma1(a: Int, b: Int): Int = a+b
```



```
Funcion_Valores_RetornablesKt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Funcion con valores retornables
Dame el valor de a: 10
Dame el valor de b: 20
La suma de 10 + 20 es igual a 30
La suma de 10 + 20 es igual a 30
```

2. Operadores relacionales y lógicos.

```
package Semanal.Miercoles_07_09_2022

fun main(){
    println("=== Operadores Relacionales y Logicos")

    if(true){
        println("Esto siempre sera verdadero")
    }

    if(1 == 1){
```

```

        println("1 es igual a 1")
    }

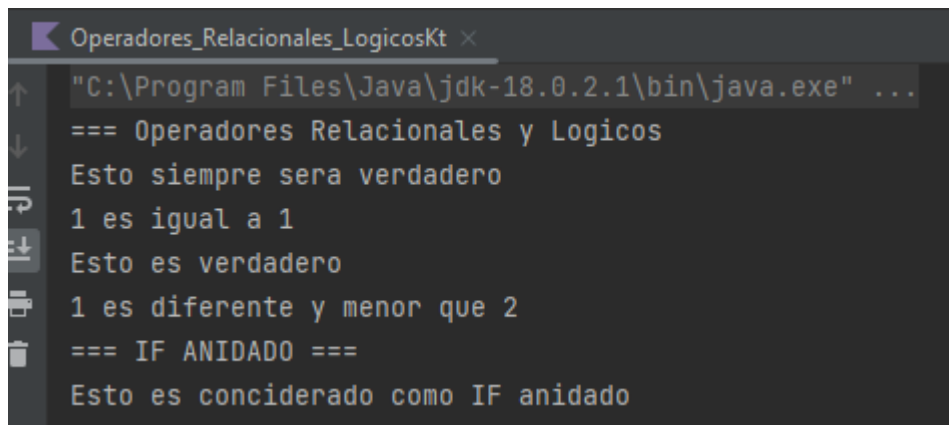
    if(1 != 1 || 1 < 2){
        println("Esto es verdadero")
    }

    if(1 != 2 && 1 < 2){
        println("1 es diferente y menor que 2")
    }else{
        println("1 es igual y menor que 2")
    }

    println("=== IF ANIDADO ===")

    if(1 == 1){
        if(1 < 2){
            println("Esto es conciderado como IF anidado")
        }
    }
}

```



```

C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
=== Operadores Relacionales y Logicos
Esto siempre sera verdadero
1 es igual a 1
Esto es verdadero
1 es diferente y menor que 2
=== IF ANIDADO ===
Esto es conciderado como IF anidado

```

3. Sobrecarga.

```

package Semanal.Miercoles_07_09_2022

fun main(){
    println("=== Sobrecarga ===")

    showProduct("Jugo")
    showProduct("Dulces","15%")
    showProduct("Caramelos","20%","30 de octubre")
    showProduct(name = "Chocolate",vigencia = "15 de noviembre")
}

fun showProduct(name: String = "Producto", promo: String = "Sin
promoción", vigencia: String = "15 de septiembre"){
    println("$name = $promo hasta el $vigencia")
}

```

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...  
=== Sobrecarga ===  
Jugo = Sin promoción hasta el 15 de septiembre  
Dulces = 15% hasta el 15 de septiembre  
Caramelos = 20% hasta el 30 de octubre  
Chocolate = Sin promoción hasta el 15 de noviembre
```

Viernes 09 / 09 / 2022

1. Ciclo When.

```
package Semanal.Viernes_09_09_2022  
  
fun main() {  
    println("==== WHEN ==== ")  
  
    print("Escribe un nombre: ")  
    var nombre = readLine()  
  
    when(nombre) {  
        "Anabel" -> println("Hola Ana")  
        "Karina" -> println("Hola Kary")  
        "Pamela" -> println("Hola Pam")  
        "Rick", "Programacion" -> println("Hola Rick")  
        "Juan" -> println("Hola Juanito")  
        else -> println("Hola desconocido")  
    }  
  
    println("Dame tu edad: ")  
    var edad = readLine()!!.toInt()  
  
    when(edad) {  
        in 1..17 -> println("Eres menor de edad")  
        in 18..50 -> println("Eres mayor de edad")  
        in 51..120 -> println("Eres de la tercera edad")  
        else -> println("Eres inhumano")  
    }  
}
```

```
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...  
==== WHEN ====  
Escribe un nombre: eder  
Hola desconocido  
Dame tu edad:  
20  
Eres mayor de edad  
  
Process finished with exit code 0
```

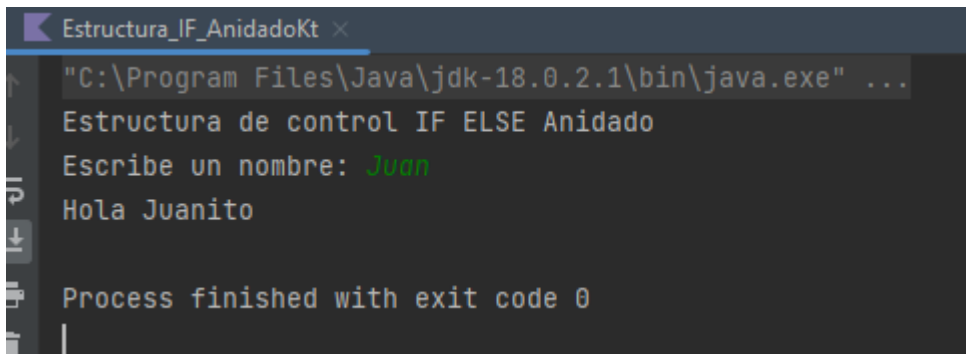

2. IF ELSE anidado.

```
package Semanal.Viernes_09_09_2022

fun main() {
    println("Estructura de control IF ELSE Anidado")

    print("Escribe un nombre: ")
    var nombre = readLine()

    if (nombre.equals("Anable")) {
        println("Hola Ana")
    } else if (nombre.equals("Karina")) {
        println("Hola Kary")
    } else if (nombre.equals("Pamela")) {
        println("Hola Pam")
    } else if (nombre.equals("Rick") || nombre.equals("Programacion")) {
        println("Hola Rick")
    } else if (nombre.equals("Juan")) {
        println("Hola Juanito")
    } else {
        println("Hola desconocido")
    }
}
```



```
Estructura_IF_AnidadoKt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Estructura de control IF ELSE Anidado
Escribe un nombre: Juan
Hola Juanito
Process finished with exit code 0
```

3. List Array.

```
package Semanal.Viernes_09_09_2022

fun main() {
    println("==== List (Arreglo Inmutable)====")
    val arr = listOf<String>("Rojo", "Verde", "Azul", "Amarillo")
    println(arr)

    println("El color verde esta en la posicion:
    ${arr.indexOf("Verde")}")

    println(main2())
}

fun main2() {
```

```

println(" ==== List (Arreglo Mutable) ====")

val mutableList = mutableListOf<String>()
mutableList.add("Paco")
mutableList.add("Luis")
mutableList.add("Carlos")
mutableList.add("Predo")
println(mutableList)
println("Voy a modificar el arreglo")
mutableList.removeAt(1)
println(mutableList)
println("Voy a modificar con un set")
mutableList.set(2, "Michel")
println(mutableList)

println(" ==== Recorido con WHILE ====")
var index = 0

while(index < mutableList.size){
    println(mutableList[index])
    index++
}

}

```

```

List_Array.kt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
==== List (Arreglo Inmutable)====
[Rojo, Verde, Azul, Amarillo]
El color verde esta en la posicion: 1
==== List (Arreglo Mutable) ====
[Paco, Luis, Carlos, Predo]
Voy a modificar el arreglo
[Paco, Carlos, Predo]
Voy a modificar con un set
[Paco, Carlos, Michel]
==== Recorido con WHILE ====
Paco
Carlos
Michel
kotlin.Unit

Process finished with exit code 0

```

4. Ciclo for.

```
package Semanal.Viernes_09_09_2022

fun main() {
    println("Arreglos y Estructuras de control repetitivas")

    muestraNombres("Hugo", "Juan", "Diego", "Rick")
}

fun muestraNombres(vararg nombres: String) {
    println("Usos del FOR")

    for(nombre in nombres) println(nombre)

    println("For con indice")
    for(i in 0..nombres.size-1) {
        println(nombres[i])
    }

    println("For con until")
    for(i in 0 until nombres.size) {
        println(nombres[i])
    }

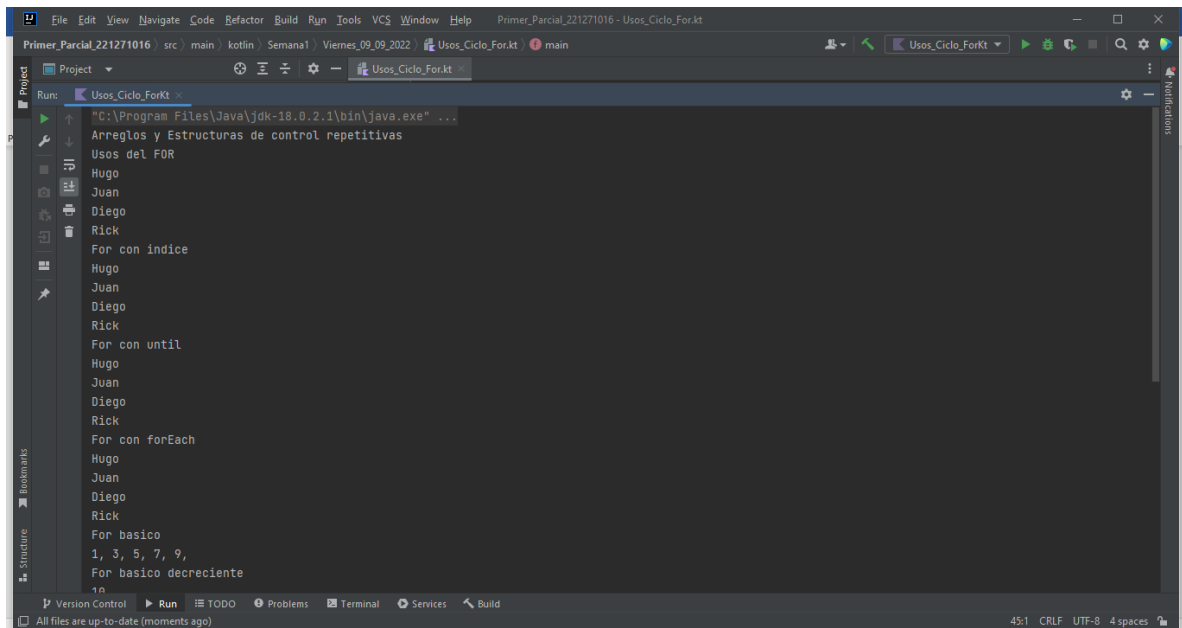
    println("For con forEach")
    nombres.forEach {
        println(it)
    }

    println("For basico")
    for(i in 1..10 step 2) {
        print("$i, ")
    }

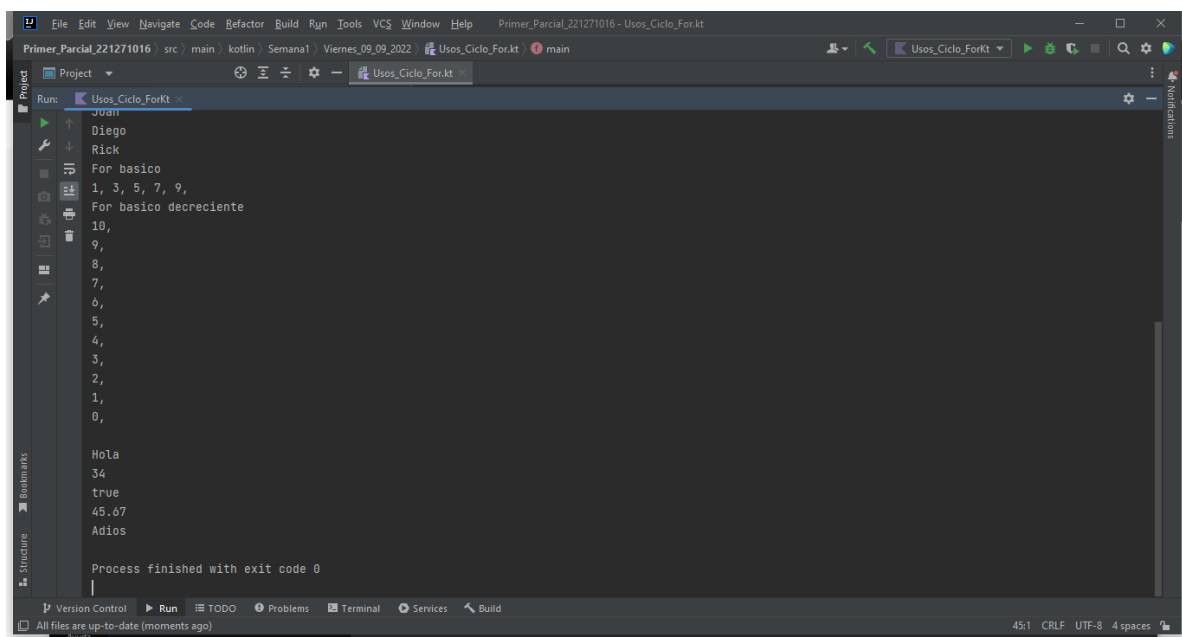
    println("\nFor basico decreciente")
    for(i in 10 downTo 0) {
        println("$i, ")
    }

    println()

    val arreglo = arrayOf("Hola", 34, true, 45.67, "Adios")
    for(i in 0 until arreglo.size) {
        println(arreglo.get(i))
    }
}
```



```
Run: Usos_Ciclo_ForKt
"C:\Program Files\Java\jdk-18.0.2\bin\java.exe" ...
Arreglos y Estructuras de control repetitivas
Usos del FOR
Hugo
Juan
Diego
Rick
For con indice
Hugo
Juan
Diego
Rick
For con until
Hugo
Juan
Diego
Rick
For con forEach
Hugo
Juan
Diego
Rick
For basico
1, 3, 5, 7, 9,
For basico decreciente
10
```



```
Run: Usos_Ciclo_ForKt
Juan
Diego
Rick
For basico
1, 3, 5, 7, 9,
For basico decreciente
10,
9,
8,
7,
6,
5,
4,
3,
2,
1,
0,

Hola
34
true
45.67
Adios

Process finished with exit code 0
```

SEMANA 2: CICLOS Y RETURN

Miércoles 14 / 09 / 2022

1. Ciclos, return y return@foreach

```
package Semana2.Miercoles_14_09_2022

fun main(){
    println(" ===== Ciclos ===== ")

    println(" ===== return@foreach ===== ")

    (1..5).forEach {
        if(it == 2){
            return@forEach
        }
        println(it)
    }

    println(" ===== Otra version de return@foreach ===== ")

    (1..5).forEach {num->
        if(num == 2){
            return@forEach
        }
        println(num)
    }

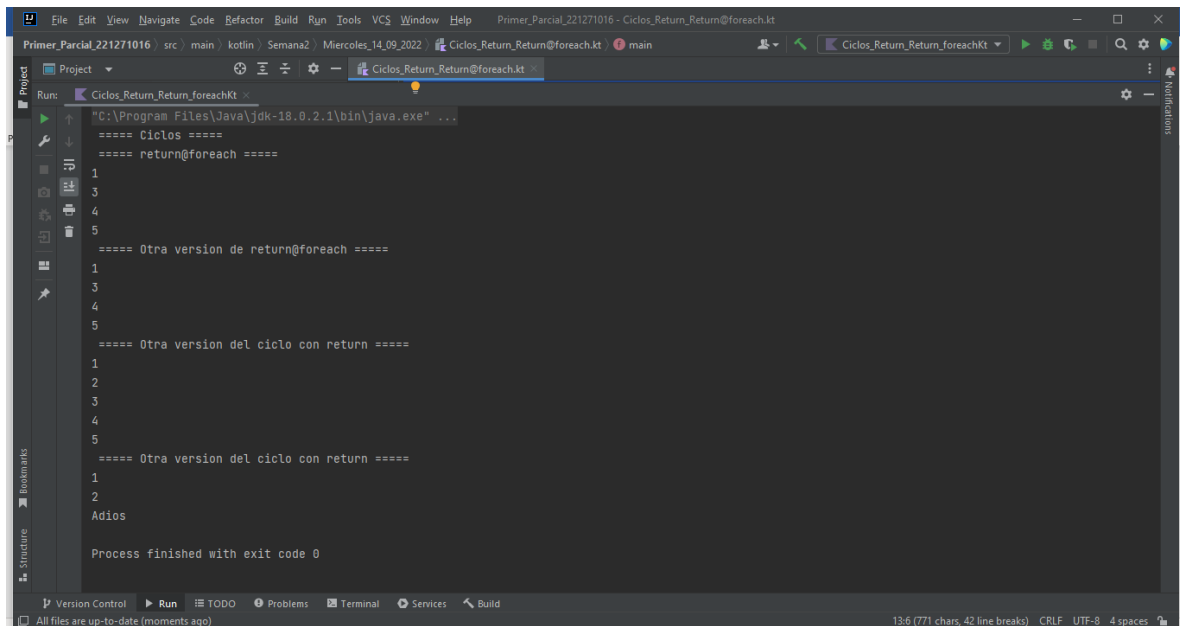
    println(" ===== Otra version del ciclo con return ===== ")

    (1..5).forEach {
        if(it == 2){
            //return
        }
        println(it)
    }

    println(" ===== Otra version del ciclo con return ===== ")

    for(indice in 1..5){
        if(indice == 3){
            break
        }
        println(indice)
    }

    println("Adios")
}
```



2. Repaso de la semana 1.

```

package Semana2.Miercoles_14_09_2022

fun main() {
    val mutableMap = mutableMapOf<String,String>()
    mutableMap.put("Al","Alain")
    mutableMap.put("Ri","Ricardo")
    mutableMap.put("Pam","Pamela")
    mutableMap.put("Hu","Hugo")
    mutableMap.put("Pa","Paco")
    mutableMap.put("Lu","Luis")
    println("Map: $mutableMap")
    println("Get by key: ${mutableMap.get("Ri")}")
    mutableMap.remove("Pam")
    mutableMap.set("Al","Alejandro")
    println("Remove & set: $mutableMap")

    println(" ===== ")

    println("Lista de Nombres")
    println("Claves: ${mutableMap.keys}")
    println("Valores: ${mutableMap.values}")
    for(nombre in mutableMap.values){
        println(nombre)
    }

    println(" ===== ")

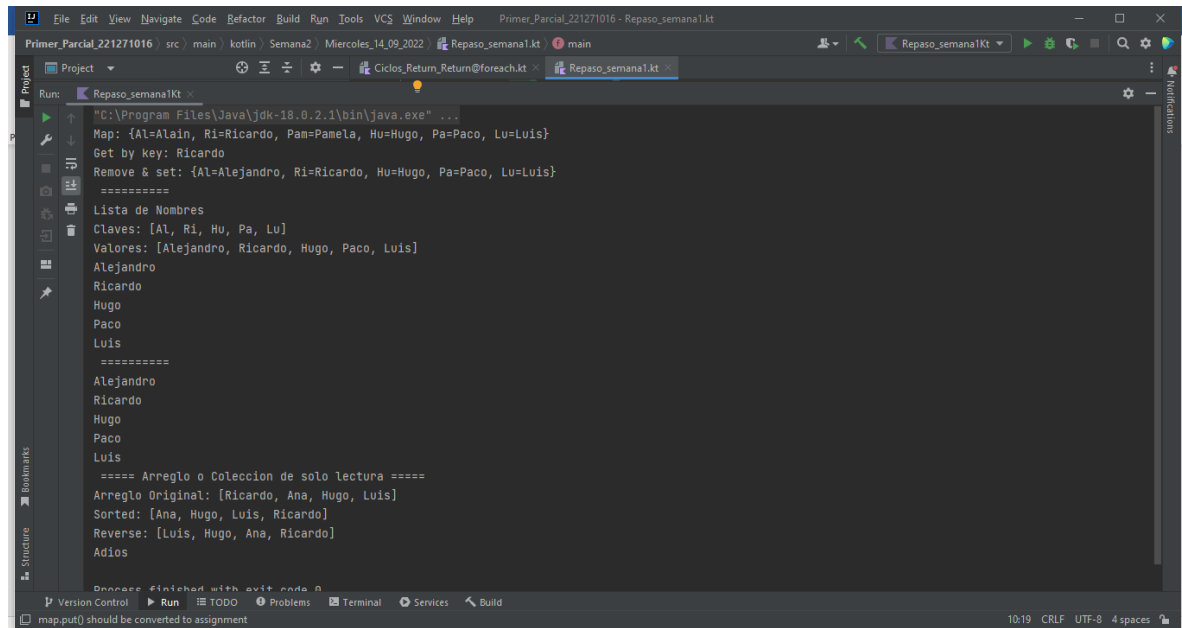
    for (i in mutableMap.keys){
        println(mutableMap[i])
    }
}

```

```
println(" ===== Arreglo o Coleccion de solo lectura ===== ")

val readOnlyList = listOf("Ricardo", "Ana", "Hugo", "Luis")
println("Arreglo Original: $readOnlyList")
println("Sorted: ${readOnlyList.sorted()}")
println("Reverse: ${readOnlyList.reversed()}")

println("Adios")
}
```



SEMANA 3: NULLS Y EXCEPTIONS

Miércoles 21 / 09 / 2022

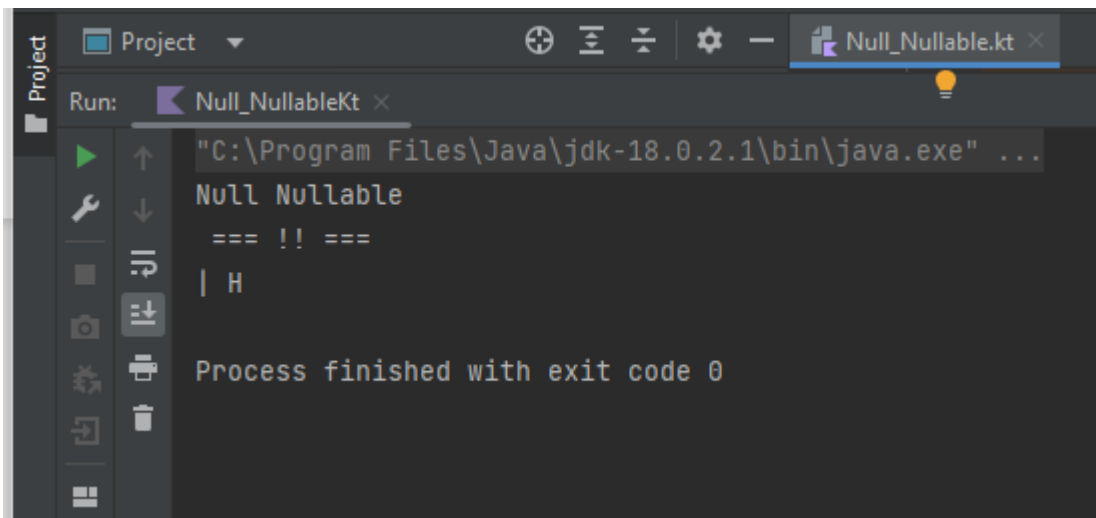
1. Null nullable.

```
package Semana3.Miercoles_21_09_2022

private var nullStrgGlobal: String? = null

fun main() {
    println("Null Nullable")
    println(" === !! === ")
    var nullStr: String? = null
    showMessage(nullStr)
    nullStr = "Hola"
    nullStrgGlobal = null //"kotlin"
    showMessage(nullStr)
}

private fun showMessage(msg: String?) {
    if (msg != null) {
        println("| ${msg.get(0)}")
    }
    if (nullStrgGlobal != null) {
        println("!! ${nullStrgGlobal?.get(0)}")
    }
}
```



2. Null Safety.

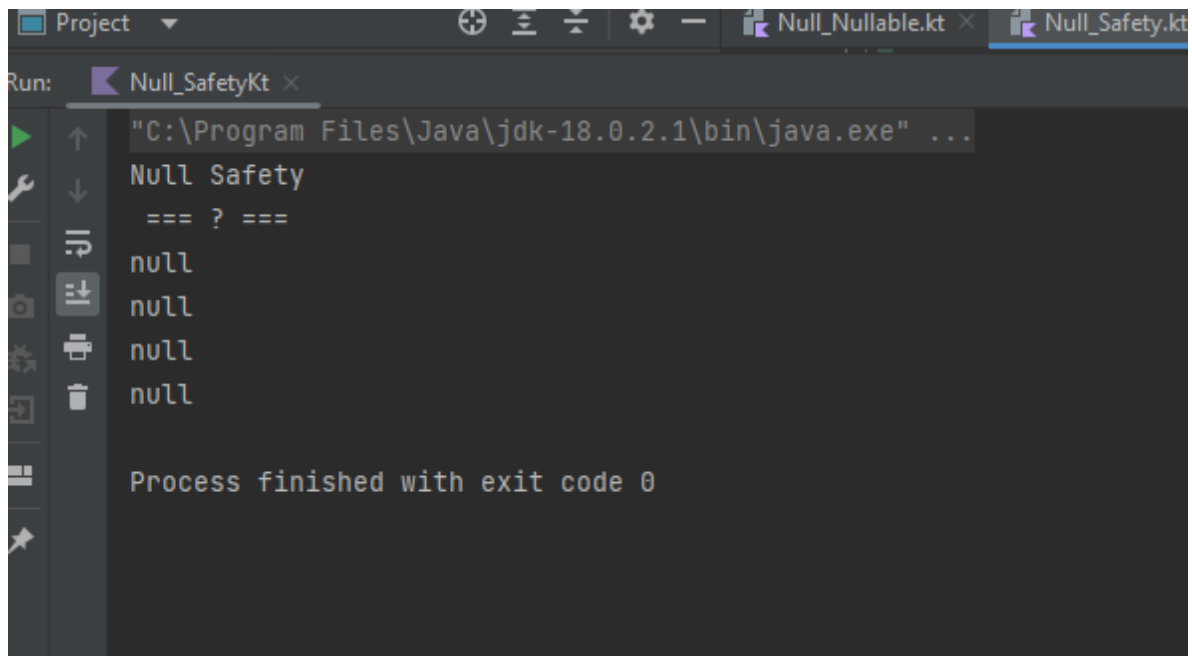
```
package Semana3.Miercoles_21_09_2022
private var nullStrGlobal: String? = null
fun main() {
    println("Null Safety")
    println(" === ? ===")
}
```

```

var nullStr: String? = null
println(nullStr)

println(nullStr?.get(0))
println(nullStr)
println(nullStrGlobal)
nullStrGlobal?.reversed()
}

```



3. Operador Elvis.

```

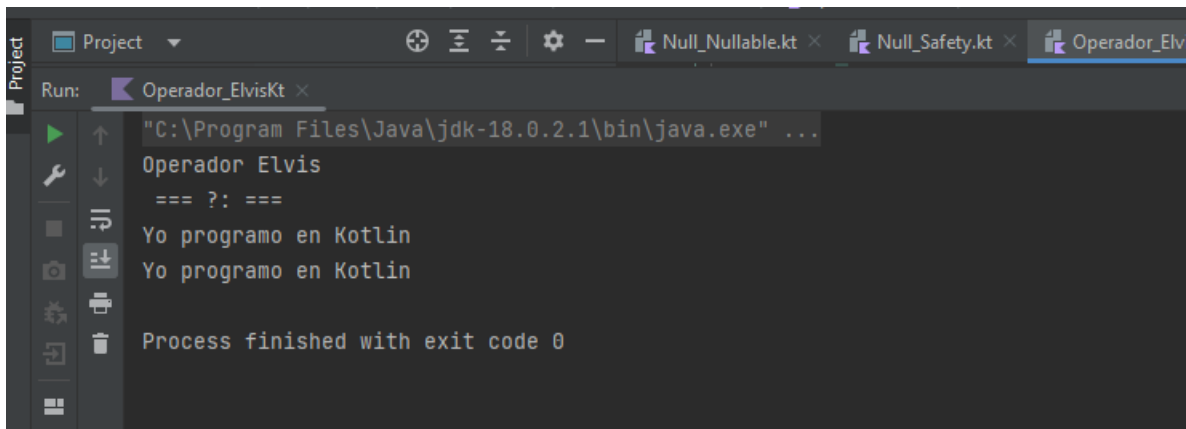
package Semana3.Miercoles_21_09_2022

private var nullStrgGlobal: String? = null

fun main() {
    println("Operador Elvis")
    println(" === ?: === ")
    nullStrgGlobal = "Kotlin"
    val elvis = nullStrgGlobal ?: "java"
    println("Yo programo en $elvis")

    val noElvis = if (nullStrgGlobal != null) {
        nullStrgGlobal
    } else {
        "java"
    }
    println("Yo programo en $noElvis")
}

```



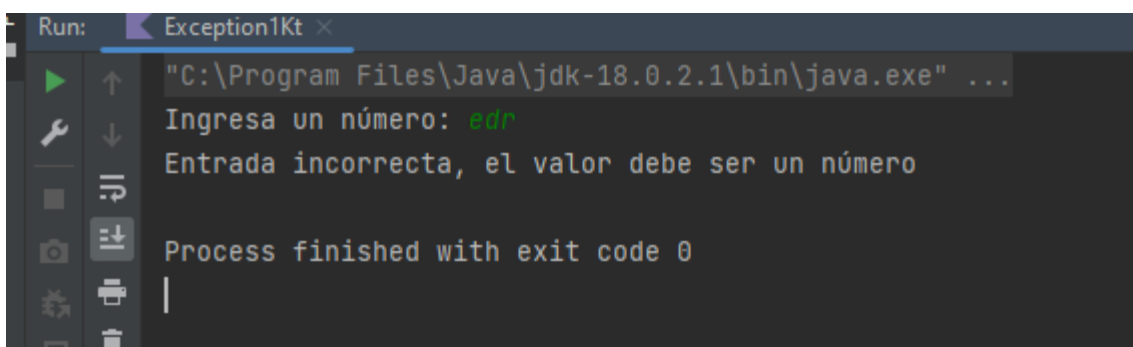
Viernes 23 / 09 / 2022.

1. Exception 1

```
package Semana3.Viernes_23_09_2022

import java.lang.Exception

fun main(){
    try{
        print("Ingresa un número: ")
        val numero = readLine()!!.toDouble()
        println(numero)
    }catch (e:Exception){
        println("Entrada incorrecta, el valor debe ser un número")
    }
}
```



2. Exception 2

```
package Semana3.Viernes_23_09_2022

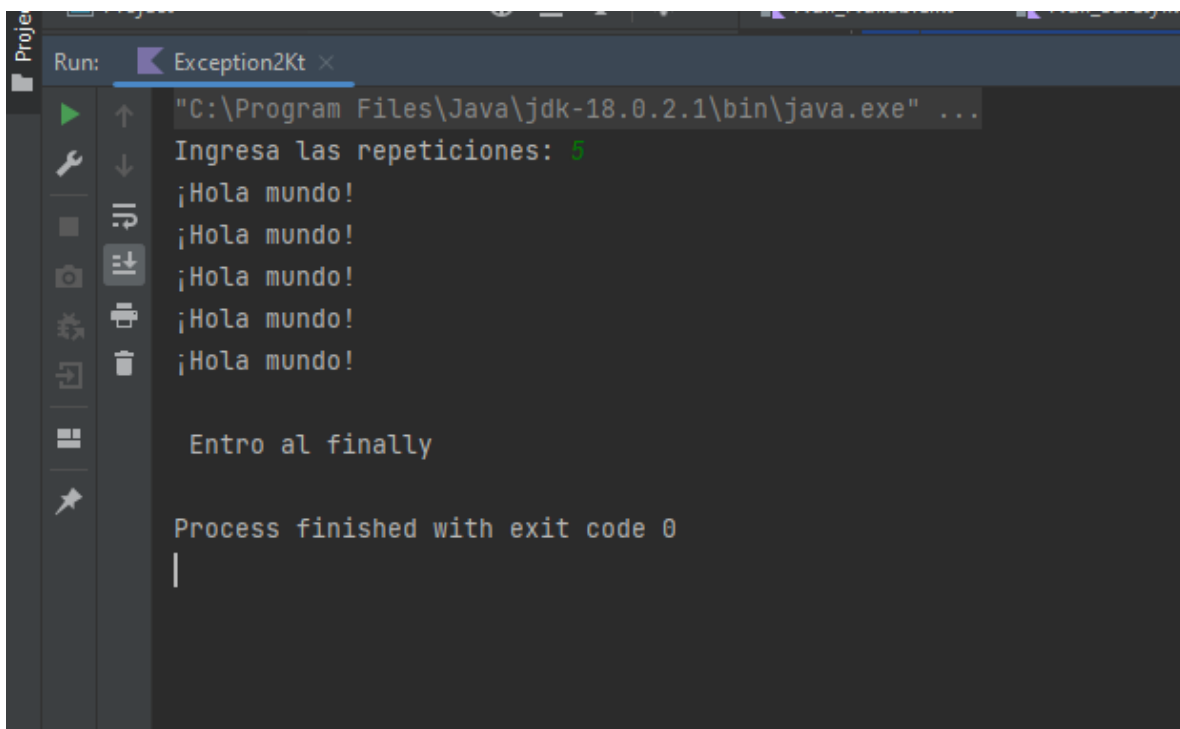
import java.io.*
import java.lang.*
```

```

fun main() {
    try {
        print("Ingresa las repeticiones: ")
        val repeticiones = readLine()!!.toInt()

        val mensaje =
File("C:\\Users\\German010519\\.android\\Archivo.txt").readLines()[0]
        for (i in 0 until repeticiones) println(mensaje)
    } catch (e: FileNotFoundException) {
        println("El archivo no existe.")
    } catch (e: NumberFormatException) {
        println("El valor de repeticiones debe ser un número.")
    } catch (e: Exception) {
        println("Error: ${e.localizedMessage}")
    } finally {
        println("\n Entro al finally")
    }
}

```



```

Run: Exception2Kt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Ingresa las repeticiones: 5
¡Hola mundo!
¡Hola mundo!
¡Hola mundo!
¡Hola mundo!
¡Hola mundo!

Entro al finally

Process finished with exit code 0

```

3. Exception 3

```

package Semana3.Viernes_23_09_2022

fun main() {
    print("Ingresa un numero: ")
    var num = readLine()!!

    var res = esNumero(num)
}

```

```

        if(res){
            println("$num es un numero")
        }else{
            println("$num no es un numero")
        }
    }
}

fun esNumero(valor: Any): Boolean{
    val resultado = try{
        valor.toString().toDouble()
        true
    }catch (e:Exception){
        false
    }
    return resultado
}

```

```

Run: Exceptio3Kt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Ingresa un numero: yh
yh no es un numero
Process finished with exit code 0

```

4. Exception 4

```

package Semana3.Viernes_23_09_2022

fun main(){
    println("Ejemplo del calculo de intereses de un prestamo bancario.")

    try {
        print("Ingresa la cantidad de dinero del prestamo: ")
        var intereses = calcularIntereses(readLine()!!.toDouble())
        println("\nEl interes a pagar es de ${intereses}")
    }catch (e:Exception){
        println(e.message)
    }
}

fun calcularIntereses(valor: Double): Double{
    if(valor > 100_000.00){
        throw Exception("El prestamo no puede ser mayor de $100,000.00")
    }
    return valor * 0.10
}

```

```
Run: Exception4Kt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Ejemplo del calculo de intereses de un prestamo bancario.
Ingresa la cantidad de dinero del prestamo: 159999
El prestamo no puede ser mayor de $100,000.00

Process finished with exit code 0
|
```

```
n: Exception4Kt x
"C:\Program Files\Java\jdk-18.0.2.1\bin\java.exe" ...
Ejemplo del calculo de intereses de un prestamo bancario.
Ingresa la cantidad de dinero del prestamo: 15000
El interes a pagar es de $1500.0

Process finished with exit code 0
|
```