INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

Ingeniería en Sistemas Computacionales

CONTROL DE FLUJO

Práctica III

DESARROLLO DE APLICACIONES MÓVILES NATIVAS

Alumno:

Saucedo Moreno César Enrique.

Profesora:

Morales Guitrón Sandra Luz.

Grupo 7CV1

17 / Septiembre / 2024

2025 ~ 1

INTRODUCCIÓN

Durante el aprendizaje de un nuevo lenguaje de programación como Kotlin, es fundamental comprender las estructuras de control y los conceptos básicos que forman la columna vertebral de la programación. Estas estructuras, como las sentencias condicionales if y when, los bucles for y while, y los conceptos de variables y funciones, son esenciales para aprender un nuevo lenguaje de programación.

Estudiar a fondo las estructuras de control y los fundamentos de Kotlin no solo es importante para escribir código que funcione correctamente, sino también para aprovechar al máximo las características avanzadas que ofrece el lenguaje. Al dominar estos conceptos, podemos desarrollar aplicaciones más robustas y eficientes, y estar mejor preparados para aprender y adaptarse a otros lenguajes.

DESARROLLO

Para esta práctica, se implementarán funciones por separado, que recibirán parámetros para realizar las tareas de los ejercicios. Dentro de la función main, se declararán las variables necesarias, lo que permitirá ajustar sus valores de manera flexible para visualizar correctamente la información deseada. Estas variables serán pasadas cómo parámetros a las funciones correspondientes en cada ejercicio.

Al estructurar el código de esta manera, se logra una mayor claridad y modularidad, facilitando la comprensión y el mantenimiento del programa. Los lectores podrán entender fácilmente cómo los datos fluyen desde la función principal hacia las funciones auxiliares, lo que hace más sencilla la posterior lectura y análisis del código.

Esta aproximación también permite ajustar y probar diferentes valores de entrada sin modificar la lógica interna de las funciones.

Ejercicio 1:

Este ejercicio exploramos la sentencia if y su funcionamiento dentro de Kotlin.

```kotlin
/**
 * Función principal que las funciones y ejercicios de la práctica 3.
 *
 * @author Saucedo Moreno César Enrique
 * @carnet 2021630676
 * @group 7CV1
 * @date 17-Septiembre-2024
 * @version 1.0
 */

fun main () {  new *
    var exUno = 10
    ejercicioUno(exUno)
```

```kotlin
fun ejercicioUno (variableEx1 : Int ){  new *
    if (variableEx1 > 30)
        println("Hello, I am If statement running now.....")
    println("The End")
    println("Fin del ejercicio 1")
}
```

Ejecución:

```
/usr/lib/jvm/java-17-openjdk-amd64/bin/java -javaagent:/snap/
The End
Fin del ejercicio 1
```

Ejercicio 2:

En este ejemplo seguimos explorando el funcionamiento del if.

```
fun main () {  new *
    var exUno = 10
    ejercicioUno(exUno)
    var exDos = 10
    ejercicioDOs(exDos)
```

```
fun ejercicioDOs (variableEx2 : Int ){  new *
    if (variableEx2 > 30)
        println("Hello, I am If statement running now.....")
    else
        println("Hello, I am Else statement running now.....")
    println("Fin del ejercicio 2")
}
```

Ejecución:

```
Hello, I am Else statement running now.....
Fin del ejercicio 2
```

Ejercicio 3:

En estos ejercicios exploramos aún mas la sentencia if, cómo se estructura y trabajan los if's anidados.

```
    var exTres = 80
    ejercicioTres(exTres,  variable_2Ex3: null)
    var exTres_2 = 50
    ejercicioTres(exTres_2,  variable_2Ex3: null)
```

```
fun ejercicioTres (variableEx3 : Int , variable_2Ex3 : String ?) {  new *
    var variable_2Ex3 = variable_2Ex3
    if (variableEx3 >=90) variable_2Ex3 = "Grado A"
    else if (variableEx3 >=80) variable_2Ex3 = "Grado B"
    else if (variableEx3 >=70) variable_2Ex3 = "Grado C"
    else if (variableEx3 >=60) variable_2Ex3 = "Grado D"
    else variable_2Ex3 = "Grado F"
    println(variable_2Ex3)
    println("Fin del ejercicio 3")
}
```

Ejecución:

```
Grado B
Fin del ejercicio 3
Grado F
Fin del ejercicio 3
```

Ejercicio 4:

En este ejercicio seguimos trabajando con el if pero ahora con operadores lógicas, cómo lo es un OR, donde esta podra ser cierta si al menos una de las dos condiciones se cumple.

```
    var exCuatro = 16
    var exCuatro_2 = 1998
    ejercicioCuatro(exCuatro, exCuatro_2)
```

Función que implementa la lógica del ejercicio.

```
fun ejercicioCuatro (variableEx4 : Int , variable_2Ex4 : Int){  new *
    if(variableEx4 >= 18 || variable_2Ex4 >= 1998) println("He is Authorized")
    else println("He is not Authorized")
    println("Fin del ejercicio 4")
}
```

Ejecución:

```
He is Authorized
Fin del ejercicio 4
================================================
```

Ejercicio 5:

En este ejercicio exploramos y ocupamos la sentencia When, la cuál tiene un comportamiento cómo lo es el equivalente de Switch en otros lenguajes cómo C o

Java, la cuál es más cómoda para trabajar, sin embargo, su sintaxis es un poco distinto, ya que utiliza una "->" para señalar el comportamiento que va a realizar.

```
    println("================================================")
    println("====\tEjercicio 5 Pizza\t====")
    println("""Selecciona el tamaño de la pizza : \n
        1-> Chica \n
        2-> Mediana \n
        3-> Grande \n""")
    var exCinco = readLine()!!.toInt()
    var exCinco_2 : Int = 0
    ejercicioCinco(exCinco, exCinco_2)
```

```
fun ejercicioCinco (variableEx5 : Int , variable_2Ex5 : Int){  new *
    var variable_2Ex5 = variable_2Ex5
    when (variableEx5){
        1-> variable_2Ex5 = 5
        2-> variable_2Ex5 = 7
        3-> variable_2Ex5 = 9
        else -> println("NO seleccionaste el tamaño correcto")
    }
    println("El precio total es : $variable_2Ex5 USD")
    println("Fin del ejercicio 5")
}
```

Ejecución:

```
================================================
====    Ejercicio 5 Pizza   ====
Selecciona el tamaño de la pizza : \n
         1-> Chica \n
         2-> Mediana \n
         3-> Grande \n"
3
El precio total es : 9 USD
Fin del ejercicio 5
```

Ejercicio 6:

En este ejercicio repasamos el bucle for, el cuál es algo similar al for in range de python.

```
fun ejercicioSeis(){  new *
    for (x in 0 ≤ .. ≤ 5){
        println("El valor de x es : $x")
    }
    println("Fin del ejercicio 6")
}
```

Ejecución:

```
El valor de x es : 0
El valor de x es : 1
El valor de x es : 2
El valor de x es : 3
El valor de x es : 4
El valor de x es : 5
Fin del ejercicio 6
```

Ejercicio 7:

En este ejercicio repasamos el bucle for, el cuál es algo similar al for in range de python, sin embargo, podemos ver que utiliza un control de flujo con el if, donde activa una bandera o rompe el ciclo con el break.

```
fun ejercicioSiete(){  new *
    for (x in 0 ≤ .. ≤ 5){
        println("El valor de x es : $x")
        if (x == 3) break
    }
    println("Fin del ejercicio 7")
}
```

Ejecución:

```
El valor de x es : 0
El valor de x es : 1
El valor de x es : 2
El valor de x es : 3
Fin del ejercicio 7
```

Ejercicio 8:

En este ejercicio repasamos el bucle for, el cuál es algo similar al for in range de python, sin embargo, utilizamos una variable cómo bandera para que vaya iterando el bucle for.

```
fun ejercicioOcho (){  new *
    var y = 10
    for (x in 0 ≤  .. ≤ y){
        println("El valor de x es : $x")
    }
    println("Fin del ejercicio 8")
}
```

Ejecución:

```
El valor de x es : 0
El valor de x es : 1
El valor de x es : 2
El valor de x es : 3
El valor de x es : 4
El valor de x es : 5
El valor de x es : 6
El valor de x es : 7
El valor de x es : 8
El valor de x es : 9
El valor de x es : 10
Fin del ejercicio 8
```

Ejercicio 9:

En este ejercicio repasamos el bucle for, sin embargo, utilizamos una estructura de datos simple cómo lo es un array, sin embargo, vamos a recorrer nuestro arreglo mediante un for para que nos imprima su dato y lo vamos a recorrer mediante su index.

```
fun ejercicioNueve(){  new *
    var x = arrayOf(10, 20, 30, 40, 50)
    for (index in 0 ≤ .. ≤ 4){
        println("El valor de x es : ${x[index]}")
    }
    println("Fin del ejercicio 9")
}
```

Ejecución:

```
El valor de x es : 10
El valor de x es : 20
El valor de x es : 30
El valor de x es : 40
El valor de x es : 50
Fin del ejercicio 9
```

Ejercicio 10:

En este ejercicio solo repasamos el bucle while, que mientras se cumpla una condición esta ejecuta solo su bloque de código mientras no exista una bandera que rompa el bucle.

```
fun ejercicioDiez(){  new *
    var x = 1
    while (x <= 5){
        println("El valor de x es : $x")
        x++
    }
    println("Fin del ejercicio 10")
}
```

Ejecución:

```
El valor de x es : 1
El valor de x es : 2
El valor de x es : 3
El valor de x es : 4
El valor de x es : 5
Fin del ejercicio 10
```

Ejercicio 11:

En este ejercicio repasamos el bucle do while, el cuál mientras se cumpla una condición éste se seguirá ejecutando, sin embargo, en este utilizamos la variable x para que se vaya incrementando según el valor de la variable.

```
fun ejercicioOnce(){  new *
    var x = 1
    do {
        println("El valor de x es : $x")
        x++
    } while (x <= 5)
    println("Fin del ejercicio 11")
}
```

Ejecución:

```
El valor de x es : 1
El valor de x es : 2
El valor de x es : 3
El valor de x es : 4
El valor de x es : 5
Fin del ejercicio 11
```

Ejercicio 12:

En este bucle for, tiene un break, el cual cómo en C, comprobara la variable x y cuando se cumpla rompera el bucle.

```
fun ejercicioDoce (){  new *
    var x = 1
    do {
        println("El valor de x es : $x")
        x++
        if (x == 3) break
    } while (x <= 5)
    println("Fin del ejercicio 12")
}
```

Ejecución:

```
El valor de x es : 1
El valor de x es : 2
Fin del ejercicio 12
```

Ejercicio 13:

```
fun ejercicioTrece () {  new *
    var x = 0
    do {
        x++
        if (x == 3) continue
        println("El valor de x es : $x")
    } while (x <= 5)
    println("Fin del ejercicio 13")
}
```

Ejecución:

```
El valor de x es : 1
El valor de x es : 2
El valor de x es : 4
El valor de x es : 5
El valor de x es : 6
Fin del ejercicio 13
```

Ejercicio 14:

```
fun ejercicioCatorce(x : Int , y : Int){  new *
    var z = x + y
    println("La suma de $x y $y es : $z")
    println("Fin del ejercicio 14")
}
```

Ejecución:

```
La suma de 10 y 20 es : 30
Fin del ejercicio 14
```

```
La suma de 50 y 100 es : 150
Fin del ejercicio 14
```

Ejercicio 15:

Declarando la variable global antes del main:

```
/**
 * Función principal que las funciones y ejerc
 *
 * @author Saucedo Moreno César Enrique
 * @carnet 2021630676
 * @group 7CV1
 * @date 17-Septiembre-2024
 * @version 1.0
 */
var myCompania = "Cesar7CV1"
```

```
fun name (myCompania : String){  new *
    println("Mi compañia es : $myCompania")
}
```

```
var exCatorce = 10
var exCatorce_2 = 20
ejercicioCatorce(exCatorce, exCatorce_2)
var resultadoQuince = ejercicioQuince( x: 50 , y: 100)
println("El resultado de la suma es : $resultadoQuince")
name( myCompania: "Ford vs Ferrari")
```

Ejecución:

```
Mi compañia es : Ford vs Ferrari
```

CONCLUSIÓN

El trabajar con estructuras de control y los bucles s esencial al aprender cualquier lenguaje de programación, ya que son herramientas fundamentales para dirigir el flujo y la lógica de un programa. Estudiar lenguajes como C es de gran importancia, puesto que su sintaxis y conceptos básicos han servido como base para el desarrollo de muchos otros lenguajes modernos. Esto nos proporciona una comprensión sólida de los principios subyacentes que se aplican en diversos entornos de programación.

En definitiva, una sólida base en estos lenguajes y conceptos fundamentales no solo enriquece nuestras habilidades como programadores, sino que también nos prepara para afrontar con éxito los desafíos en el mundo de la programación actual.