**Practica Extra - Scala:**

**Estructuras Condicionales:**

1. Dado un número, imprimir "Positivo" si es mayor que cero, "Negativo" si es menor que cero, o "Cero" si es igual a cero.
2. Dado un carácter, imprimir "Es vocal" si es una vocal (a, e, i, o, u), o "No es vocal" si no lo es.
3. Dado un número, imprimir "Dígito" si es un número de un solo dígito, o "Número de múltiples dígitos" si tiene más de un dígito.
4. Dada una cadena de texto, imprimir "Cadena vacía" si está vacía, o "Cadena no vacía" si contiene al menos un carácter.
5. Dado un año, imprimir "Año bisiesto" si es divisible entre 4 pero no entre 100, o si es divisible entre 400. En caso contrario, imprimir "Año no bisiesto".

**Estructuras de Iteración:**

1. Imprimir los números del 10 al 1 en orden descendente.
2. Imprimir los elementos de un arreglo de enteros.
3. Calcular la potencia de un número utilizando un bucle for.
4. Imprimir los números impares del 1 al 50 utilizando un bucle do-while.
5. Calcular la suma de los primeros n números naturales utilizando un bucle while.

**Funciones en Scala:**

1. Escribe una función en Scala llamada "obtenerPromedio" que tome una lista de números enteros como argumento y devuelva el promedio de esos números.
2. Escribe una función en Scala llamada "concatenarCadenas" que tome dos cadenas como argumento y las concatene.
3. Escribe una función en Scala llamada "esPalindromo" que tome una cadena como argumento y devuelva true si es un palíndromo y false si no lo es.
4. Escribe una función en Scala llamada "duplicarElementos" que tome una lista de enteros como argumento y devuelva una nueva lista con cada elemento duplicado.
5. Escribe una función en Scala llamada "esCapicua" que tome un número entero como argumento y devuelva true si es capicúa y false si no lo es.

#Resolucion Condicionales!

1.

object CondicionalesEj1 {  
 def main(args: Array[String]): Unit = {  
 def imprimirSigno(numero: Int): Unit = {  
 if (numero > 0) {  
 *println*("Positivo")  
 } else if (numero < 0) {  
 *println*("Negativo")  
 } else {  
 *println*("Cero")  
 }  
 }  
  
 // Ejemplo de uso  
 val numero1 = 5  
 val numero2 = -2  
 val numero3 = 0  
  
 imprimirSigno(numero1) // Imprime "Positivo"  
 imprimirSigno(numero2) // Imprime "Negativo"  
 imprimirSigno(numero3)  
  
 }  
}

2.

object CondicionalesEj2 {  
 def main(args: Array[String]): Unit = {  
 if (args.nonEmpty) {  
 *println*(if ("aeiou".contains(args(0).toLowerCase)) "Es vocal" else "No es vocal")  
 } else {  
 *println*("Por favor, proporciona un carácter como argumento.")  
 }  
 }  
}

3.

object CondicionalesEj3 {  
 def main(args: Array[String]): Unit = {  
 def verificarNumero(num: Int): Unit = {  
 if(num >= -9 && num <= 9) {  
 *println*("Digito")  
 } else  
 {  
 *println*("numero de multiples digitos")  
 }  
 }  
 verificarNumero(15)  
 }  
}

4.

object CondicionalesEj4 {  
 def main(args: Array[String]): Unit = {  
 def verificarCadena(cadena: String): Unit = {  
 if(cadena.isEmpty) {  
 *println*("Cadena vacia")  
 } else  
 {  
 *println*("Cadena no vacia")  
 }  
 }  
 verificarCadena(“”)  
 }  
}

5.

object CondicionalesEj5 {

def main(args: Array[String]): Unit = {

val año = args(0).toInt

println(if ((año % 4 == 0 && año % 100 != 0) || (año % 400 == 0)) "Año bisiesto" else "Año no bisiesto")

}

}

##Resolucion – Iteración

1.

object Iteracion1 {

def main(args: Array[String]): Unit = {

for (i <- 10 to 1 by -1) println(i)

}

}

2.

object Iteracion2 {

def main(args: Array[String]): Unit = {

val arreglo = Array(1, 2, 3, 4, 5)

for (elemento <- arreglo) println(elemento)

}

}

3.

object Iteracion3 {

def main(args: Array[String]): Unit = {

val base = args(0).toInt

val exponente = args(1).toInt

var resultado = 1

for (\_ <- 1 to exponente) resultado \*= base

println(resultado)

}

}

4.

object Iteracion4 {  
 def main(args: Array[String]): Unit = {  
 (1 to 50).foreach { i =>  
 if (i % 2 != 0) {  
 println(i)  
 }  
 }  
 }  
}

5.

object Iteracion5 {  
 def main(args: Array[String]): Unit = {  
 def calcularSumaNaturales(n: Int): Int = {  
 var suma = 0  
 var contador = 1  
  
 while (contador <= n) {  
 suma += contador  
 contador += 1  
 }  
  
 suma  
 }  
  
 // Ejemplo de uso  
 val n = 10  
 val suma = calcularSumaNaturales(n)  
 println(s"La suma de los primeros $n números naturales es: $suma")  
 }  
}

##RESOLUCION FUNCIONES!

1.

object Funciones1 {  
 def main(args: Array[String]): Unit = {  
 def obtenerPromedio(nums: List[Int]): Double = {  
 val suma = nums.sum  
 val cantidad = nums.length  
 suma.toDouble / cantidad.toDouble  
 }  
  
 val numeros = List(1, 2, 3, 4, 5)  
 val promedio = obtenerPromedio(numeros)  
 *println*(promedio) // Imprime: 3.0  
 }  
}

2.

object Funciones2 {  
 def main(args: Array[String]): Unit = {  
 def concatenarCadenas(cadena1: String, cadena2: String): String = {  
 cadena1 + cadena2  
 }  
  
 val resultado = concatenarCadenas("Hola ", "mundo!")  
 *println*(resultado) // Imprime "Hola mundo!"  
 }  
}

3.

object Funciones3 {

def main(args: Array[String]): Unit = {

def esPalindromo(cadena: String): Boolean = {

val cadenaSinEspacios = cadena.replaceAll("\\s", "").toLowerCase()

cadenaSinEspacios == cadenaSinEspacios.reverse

}

val cadena1 = "Anita lava la tina"

val cadena2 = "Hola mundo"

println(esPalindromo(cadena1)) // Output: true

println(esPalindromo(cadena2)) // Output: false

}

}

4.

object Funciones4 {

def duplicarElementos(lista: List[Int]): List[Int] = {

lista.map(\_ \* 2)

}

def main(args: Array[String]): Unit = {

println(duplicarElementos(List(1, 2, 3)))

}

}

5.

object Funciones5 {

def esCapicua(numero: Int): Boolean = {

val strNum = numero.toString

strNum == strNum.reverse

}

def main(args: Array[String]): Unit = {

println(esCapicua(12321)) // true

}

}