



Taller Estructuras de Datos en Kotlin

Daniel Mauricio Villa Patiño, ADSI 2469283.

El objetivo de este taller es que los aprendices sean capaces de comprender y utilizar las principales estructuras de datos en Kotlin, incluyendo arreglos, listas, conjuntos, mapas y pares.

El aprendiz deberá realizar un informe donde se evidencien los siguientes puntos:

1. Introducción a las estructuras de datos en Kotlin

a. ¿Qué son las estructuras de datos y para qué se utilizan?

Son una manera de representar información en la computadora, acomodada determinadamente, para así, hacer operaciones de manera más eficiente.

b. Ventajas de utilizar estructuras de datos en Kotlin

- Estas nos permiten hacer el trabajo con mayor facilidad
- Se optimiza el tiempo al momento de hacer trabajos
- Se tienen las herramientas necesarias para resolver problemas

c. Diferencias entre las estructuras de datos en Kotlin y Java

La diferencia entre ambas es la sintaxis, por ende, la cantidad de código entre estas es diferente, en java el código es más extenso en comparación a kotlin por la cantidad de codificación, más reducida.

2. Arreglos en Kotlin

a. ¿Qué es un arreglo?

Es una manera de almacenar varios datos a la vez, como números letras o ambos.

b. Creación de arreglos en Kotlin

c. Accediendo a los elementos de un arreglo

d. Modificando los elementos de un arreglo

e. Recorriendo un arreglo

f. Funciones útiles para trabajar con arreglos en Kotlin

```
fun main (){

    //Creacion de un arreglo
    var array1 = arrayOf("Daniel", "Yuly", "Nicol", "Sharis")
    println("Cantidad: "+array1.size)

    //Accediendo a los elementos
    println("Simple: "+array1[2])
    println("Con .get: "+array1.get(1))

    //Modificando los elementos de un arreglo
    array1.set(0,"Sara")

    //Recorriendo un arreglo
    for (i in array1) {
        println(i)
    }

    println("Utilizando funcines utiles")

    //funciones utiles
    //el tipo es entero, solo recibira valores enteros
    var declarandoTipo = intArrayOf(1,2,3,4)

    //igual que el anterior, pero como no se le an asignado valores da
    //como outdefault 0 o el que se elija colocando en las llaves
    var declarandoTipo2 = IntArray(4) {1}

    for (i in declarandoTipo) {
        println(i)
    }
    for (i in declarandoTipo2) {
        println(i)
    }
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA 2022\trai
Cantidad: 4
Simple: Nicol
Con .get: Yuly
Sara
Yuly
Nicol
Sharis
Utilizando funcines utiles
1
2
3
4
1
1
1
1
```

3. Listas en Kotlin

a. ¿Qué es una lista?

Una lista es un conjunto de elementos con orden específico

b. Creación de listas en Kotlin

c. Accediendo a los elementos de una lista

d. Modificando los elementos de una lista

e. Recorriendo una lista

f. Funciones útiles para trabajar con listas en Kotlin

```
fun main () {
    //lista que no se modifica
    var lista1 = listOf("Pepito", "Julano", "Sutano")
    //lista modificable
    var lista2 = mutableListOf(1, 2, 3, 4)

    //agregar
    lista2.add(5)

    println(lista1)
    println(lista2)

    //hallar mediante la posición
    println(lista2.get(1))
    println(lista2[2])

    //reemplazar
    lista2[1] = 20
    println(lista2)
    lista2.set(0, 8)
    println(lista2)

    //remover
    lista2.removeAt(0)
    println(lista2)
    lista2.remove(5)
    println(lista2)

    // Recorriendo una lista
    for (i in lista2) {
        println(i)
    }

    //ordena de menor a mayor
    lista2.sort()
    //ordena de mayor a menor
    lista2.sortDescending()

    //funciones útiles para listas no mutables
    //verifica que este elemento esté en la lista
    println("esta Pepito: "+lista1.contains("Pepito"))
    println("esta nana: "+lista1.contains("nana"))
    //valor aleatorio
    println("aleatorio: "+lista1.random())
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA 2020"
-jar list.jar
[Pepito, Julano, Sutano]
[1, 2, 3, 4, 5]
2
3
[1, 20, 3, 4, 5]
[8, 20, 3, 4, 5]
[20, 3, 4, 5]
[20, 3, 4]
20
3
4
esta Pepito: true
esta nana: false
aleatorio: Julano
```

4. Conjuntos en Kotlin

a. ¿Qué es un conjunto?

Colección sin orden de datos únicos, no deberían tener duplicados.

b. Creación de conjuntos en Kotlin

c. Accediendo a los elementos de un conjunto

d. Modificando los elementos de un conjunto

e. Recorriendo un conjunto

f. Funciones útiles para trabajar con conjuntos en Kotlin

```
fun main () {
    // creando un conjunto inmutable
    var conjunto = setOf(1,2,3,4,5)
    // creando un conjunto mutable
    var conjunto2 = mutableSetOf<Int>(6,7,8,9)

    //Accediendo al conjunto, puede lanzar un error si no se encuentra en el rango
    println (conjunto.elementAt(0))
    //Accediendo al conjunto, puede lanzar un Null si no se encuentra en el rango
    println (conjunto.elementAtOrNull(5))

    // agregar
    conjunto2.add(10)
    // Agregar cantidades mas grandes
    conjunto2.addAll(conjunto)

    println("Se agregaron: "+conjunto2)

    println("")

    //recorriendo conjunto
    println("Recorrido")
    var i = 0
    do{
        println (conjunto2.elementAt(i))
        i++
    }while (i < (conjunto2.size))

    //funciones utiles
    //imprime el primer valor
    println("Primero: "+conjunto.first())
    //imprime el ultimo valor
    println("Ultimo: "+conjunto.last())
    //suma los valores
    println("Suma: "+conjunto.sum())
    //hece el promedio de los valores
    println("Promedio: "+conjunto.average())
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA 2022\trabajo"
1
null
Se agregaron: [6, 7, 8, 9, 10, 1, 2, 3, 4, 5]

Recorrido
6
7
8
9
10
1
2
3
4
5
Primero: 1
Ultimo: 5
Suma: 15
Promedio: 3.0
```

5. Mapas en Kotlin

a. ¿Qué es un mapa?

Es un conjunto de pares,

b. Creación de mapas en Kotlin

c. Accediendo a los elementos de un mapa

d. Modificando los elementos de un mapa

e. Recorriendo un mapa

f. Funciones útiles para trabajar con mapas en Kotlin

```
fun main () {
    //map sin cambios posibles
    var maps = mapOf("David" to 12, "Juan" to 13)
    //map con modificaciones
    var maps2 = mutableMapOf("Daniel" to 12, "Mauricio" to 13)

    // por medio de la llave se obtiene el valor
    println(maps["David"])
    //todas las llaves
    println(maps.keys)
    //todos los valores
    println(maps.values)

    // Cambia el valor de la llave puesta
    maps2["Daniel"]=20

    // agregar
    maps2.put("Juan", 8)
    println(maps2)

    //eliminar
    maps2.remove("Juan")

    println(maps2)

    //recorrer un map
    for (i in maps2) {
        println(i)
    }
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA 2022\trabajo"
12
[David, Juan]
[12, 13]
{Daniel=20, Mauricio=13, Juan=8}
{Daniel=20, Mauricio=13}
Daniel=20
Mauricio=13
```

6. Pares en Kotlin

a. ¿Qué es un par?

b. Creación de pares en Kotlin

c. Accediendo a los elementos de un par

d. Modificando los elementos de un par

No se pueden modificar los elementos de un par debido a que es inmutable, pero se puede hacer un nuevo objeto con valores actualizados.

e. Recorriendo un par

f. Funciones útiles para trabajar con pares en Kotlin

<pre> fun main(){ // se crea un par de datos var pair1 = Pair("Daniel", 18) //accede a los elementos println(pair1.first) println(pair1.second) //recorrer a los elementos for (i in pair1.toList()){ println(i) } //nuevo objeto para asi actualizar un dato par var update = pair1.copy(first = "Adios") println(pair1) println(update) // funciones utiles // retorna primer el elemento var elemento1 = update.component1() // retorna segundo elemento var elemento2 = update.component2() // convierte en lista var list = update.toList() // convierte en cadena var cadena = update.toString() println(elemento1) println(elemento2) println(list) println(cadena) } </pre>	<pre> [Running] cd "c:\Users\Usuario\Desktop\ Daniel 18 Daniel 18 (Daniel, 18) (Adios, 18) Adios 18 [Adios, 18] (Adios, 18) [Done] exited with code=0 in 36.231 sec </pre>
--	---

7. Prácticas de estructuras de datos en Kotlin

- Ejercicios prácticos para aplicar los conceptos aprendidos
- Solución a los ejercicios prácticos

```

1
2  fun main () {
3      // ejercicio1 ()
4      // ejercicio2 ()
5      // ejercicio3 ()
6      // ejercicio4 ()
7      // ejercicio5 ()
8  }

```

```

// pepe quiere saber cual es su promedio en matematicas ya que sus notas son 3,4,5 y 2.
fun ejercicio1 () {
    var notas = arrayOf (3.0,4.0,5.0,2.0)
    var suma = 0.0;
    for (i in notas) {
        suma += i;
    }
    var prom: Double = suma/notas.size;
    println ("El promedio de pepe es: "+prom)
}

```

```

[Running] cd "c:\Users\Usuario\Des
El promedio de pepe es: 3.5

```

```
// se necesita saber el listado de personas registradas, cantidad de personas, verificar
// si esta Daniel y Diego, en caso de no estar ninguno agregarlo.
```

```
fun ejercicio2 (){
    var nombres = mutableListOfOf ("Daniel","Sara","Mauricio","Susana")
    println("Lista personas: ")
    for (i in nombres) {
        println(i)
    }
    println("Cantidad de personas registradas: "+nombres.size)

    if (nombres.contains("Daniel") == false){
        println ("Daniel no esta registrado, se registrará inmediatamente")
        nombres.add("Daniel")
        println(nombres)
    } else {
        println("Daniel esta registrado")
    }

    if (nombres.contains("Diego") == false){
        println ("Diego no esta registrado, se registrara inmediatamente")
        nombres.add("Diego")
        println(nombres)
    } else {
        println("Diego esta registrado")
    }

    println("Cantidad de personas registradas: "+nombres.size)
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA 2022\trabajos d
Lista personas:
Daniel
Sara
Mauricio
Susana
Cantidad de personas registradas: 4
Daniel esta registrado
Diego no esta registrado, se registrara inmediatamente
[Daniel, Sara, Mauricio, Susana, Diego]
Cantidad de personas registradas: 5
```

```
// en la tienda de maria manejan dos cuentas de empleados, las cuales se tienen que fucionar y
// hacer un conteo de personal. tambien hay que agregar 2 nit que se le olvido a maria poner en
// la cuenta de empleados. cada empleadoo le pagan 100,000 al dia, cuanto dinero tiene que apartar
// maria para pagar sus empleados
```

```
fun ejercicio3 (){
    var cuenta1 = setOf(1,2,3,4)
    var cuenta2 = mutableSetOf (5,6,7)
    println("Cuenta nit #1: ")
    for (i in cuenta1) {
        println(i)
    }
    println("Cuenta nit #2: ")
    for (i in cuenta2) {
        println(i)
    }
    cuenta2.add(9)
    cuenta2.add(8)
    cuenta2.addAll(cuenta1)

    var suma = 0;
    for (i in cuenta2) {
        suma ++
    }

    println("Cuenta completa de empleados: "+cuenta2)
    println("La suma total de los empleados es: "+suma)
    println("El dinero total a apartar para sus empleados es de: "+(suma*100000))
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA 2022\trabajos de cristi
Cuenta nit #1:
1
2
3
4
Cuenta nit #2:
5
6
7
Cuenta completa de empleados: [5, 6, 7, 9, 8, 1, 2, 3, 4]
La suma total de los empleados es: 9
El dinero total a apartar para sus empleados es de: 900000
```

```
// paula decea saber que edad tiene Daniel y Juan, tambien busca actulizar la edad de salome a 24,  
// si hay menores de edad se eliminaran.
```

```
fun ejercicio4 (){  
    var personas = mutableMapOf("Daniel" to 18,"Juan" to 15,"Salome" to 23)  
    println("Daniel tiene una edad de: "+personas["Daniel"])  
    println("Juan tiene una edad de: "+personas["Juan"])  
    personas["Salome"] = 24  
    println("")  
    println("Personas")  
    println(personas)
```

```
    val person1 = personas["Daniel"]  
    val person2 = personas["Juan"]  
    val person3 = personas["Salome"]
```

```
    println("Se elimina los menores de edad")
```

```
    if (person1 != null && person1 < 18){  
        personas.remove("Daniel")  
        println(personas)  
    } else if (person2 != null && person2 < 18) {  
        personas.remove("Juan")  
        println(personas)  
    } else if (person3 != null && person3 < 18){  
        personas.remove("Salome")  
        println(personas)  
    }  
}
```

```
}
```

```
[Running] cd "c:\Users\Usuario\Desktop\SENA"  
Daniel tiene una edad de: 18  
Juan tiene una edad de: 15
```

```
Personas  
{Daniel=18, Juan=15, Salome=24}  
Se elimina los menores de edad  
{Daniel=18, Salome=24}
```

```
[Done] exited with code=0 in 37.865 seconds
```

```
//hacer un elemento que pida el nombre y la edad y que calcule cuanto tendra en 45 años
```

```
fun ejercicio5 (){  
    println("Ingrese su nombre: ")  
    var nombre = readLine()  
  
    println("Ingrese su edad: ")  
    var edad = readLine()!!.toInt()  
  
    var datos = Pair(nombre,edad)  
    var edadFuturo = Pair("futuro",45)  
  
    var edadF = datos.second + edadFuturo.second  
    // var nombrePair =  
    // var edad =  
  
    println(datos.first+", tiene una edad de "+datos.second+" en 45 años tendrá una edad de $edadF")  
}
```

```
PS C:\Users\Usuario\Desktop\SENA 2022\trabajos de cristian\kotlin\taller1> java -ja  
Ingrese su nombre:  
Angela  
Ingrese su edad:  
18  
Angela, tiene una edad de 18 en 45 años tendrá una edad de 63
```