

ED EstructuraDatosAplicadas master

Current File

EntradaSalidaDatos.kt

```
1 fun main() { new *
2     print("dame el valor de a: ")
3     var a = readln().toInt()
4     print("dame el valor de b: ")
5     var b = readln().toInt()
6     println("la suma de " + a + " + " + b + " = " + (a+b))
7     println("la suma de $a + $b = ${a+b}")
8     println("la resta de $a - $b = ${a-b}")
9     println("la multiplicacion de $a * $b = ${a*b}")
10    println("la division de $a / $b = ${a/b}")
11    println("el residuo de $a % $b = ${a%b}")
12 }
```

EstructuraDatosAplicadas > src > EntradaSalidaDatos.kt 12:2 CRLF UTF-8 4 spac...

ED EstructuraDatosAplicadas master

Current File

EntradaSalidaDatos.kt

Run EntradaSalidaDatosKt

```
C:\Users\luffy\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.1.1\lib\idea_rt.jar=594
dame el valor de a: 5
dame el valor de b: 7
la suma de 5 + 7 = 12
la suma de 5 + 7 = 12
la resta de 5 - 7 = -2
la multiplicacion de 5 * 7 = 35
la division de 5 / 7 = 0
el residuo de 5 % 7 = 5

Process finished with exit code 0
```

EstructuraDatosAplicadas > src > EntradaSalidaDatos.kt 12:2 CRLF UTF-8 4 spac...

```
import java.text.SimpleDateFormat
import java.util.Date

fun main() { new *
    println("escribe tu nombre: ")
    val name = readln()
    println("escribe tu edad: ")
    val edad = readln().toInt()

    print("dame el valor de a: ")
    var a = readln().toDouble()
    print("dame el valor de b: ")
    var b = readln().toDouble()
    val c = suma(a, b)
    val c1 = suma1(a, b)

    println(
        """
        Asi como lo ven en el editor asi aparece
        Nombre: $name
        """.trimIndent()
    )

    mostrarMensaje()

    funcionConParametros( nombre = name, age = edad)

    println("el valor de c es: $c")
}
```

```
fun main() { new *
    println("el valor de c es: $c")
    println("el valor de c1 es: $c1")

    println(
        """
        Un empleado con salario base obtiene un salario neto de ${calcularSalario()}

        Un empleado con salario de $40000.00 que paga de impuesto 30%
        obtiene un salario neto de ${calcularSalario( salario = 40000.0, pctImp = 0.3)}

        Un empleado con salario de $30000.00 con impuesto base ${calcularSalario( salario = 30000.0)}

        Un empleado con salario de $15000.00 con 5% impuesto ${calcularSalario(pctImp=0.5, salario=15000.0)}
        """.trimIndent()
    )

    lambda()

    val fx = { println("Hola") }

    fx()

    val fy = { f: String -> println(f) }

    fy("Esto es un paso de parametro cadena")

    val fw = { n: Int -> n + 1 }
```

```
ED EstructuraDatosAplicadas mast...
Current File

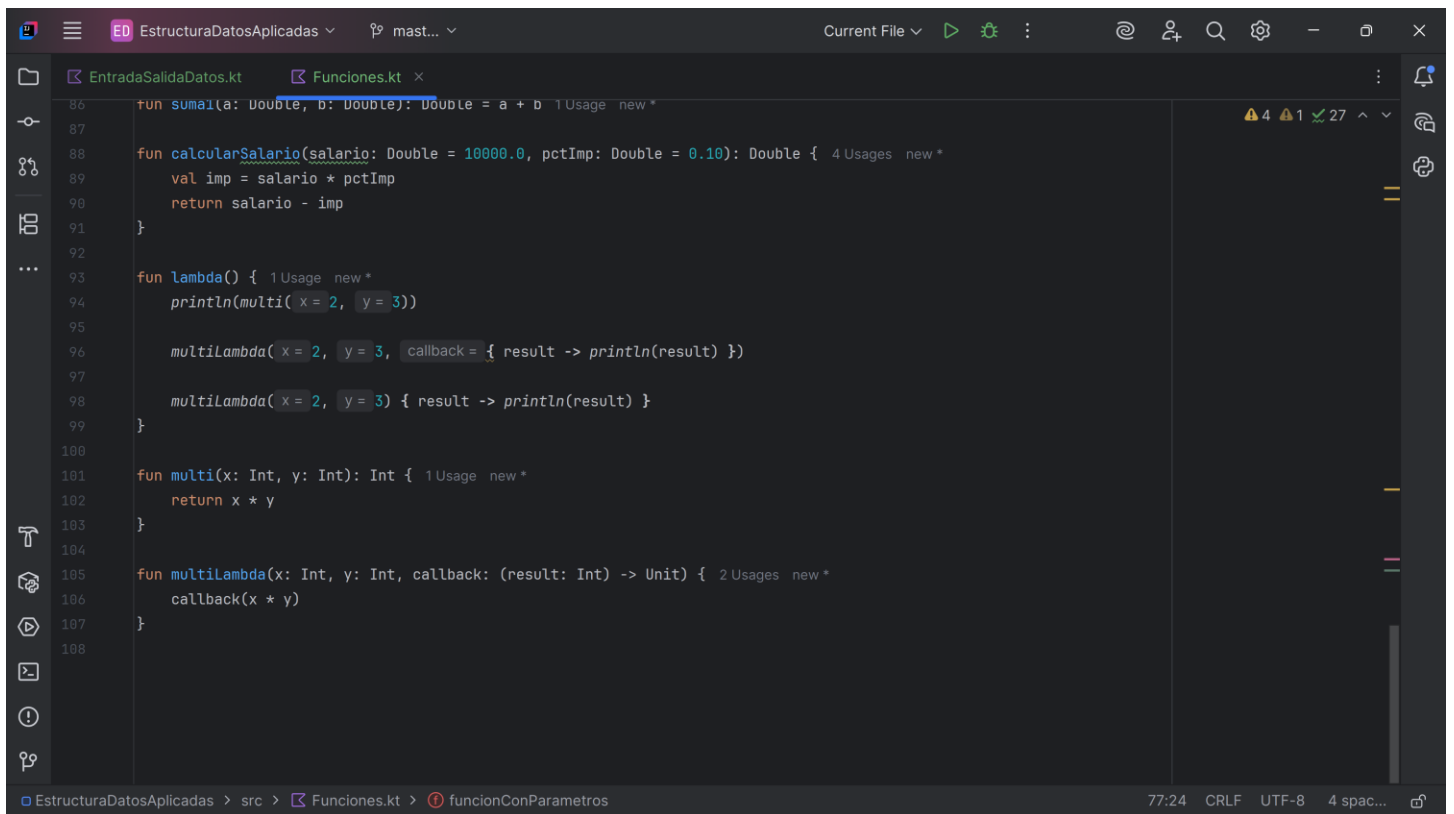
EntradaSalidaDatos.kt Funciones.kt x
4 fun main() { new *
54     val fw = { g: Int -> g + 1 }
55     println("fw = ${fw(4)}")
56
57     val x = { a: Int, b: Int, c: Int ->
58         println(a + b)
59         c * 6
60     }
61     println("x = ${x(8, 4, 2)}")
62
63     val fa = { i: Int, j: Int, k: Int -> println(i / j * k) }
64 }
65
66 fun mostrarMensaje() { 1 Usage new *
67     val fmt = SimpleDateFormat( pattern = "dd/MM/yyyy")
68     val fecha = fmt.format(Date())
69     println("""
70         mensaje: Hola desde kotlin
71         fecha: $fecha
72         """)
73 }
74
75 fun funcionConParametros(nombre: String, age: Int) { 1 Usage new *
76     println("""
77         Nombre: $nombre
78         Edad: $age
79         """).trimIndent()
80 }
```

EstructuraDatosAplicadas > src > Funciones.kt > functionConParametros 77:24 CRLF UTF-8 4 spac...

```
ED EstructuraDatosAplicadas mast...
Current File

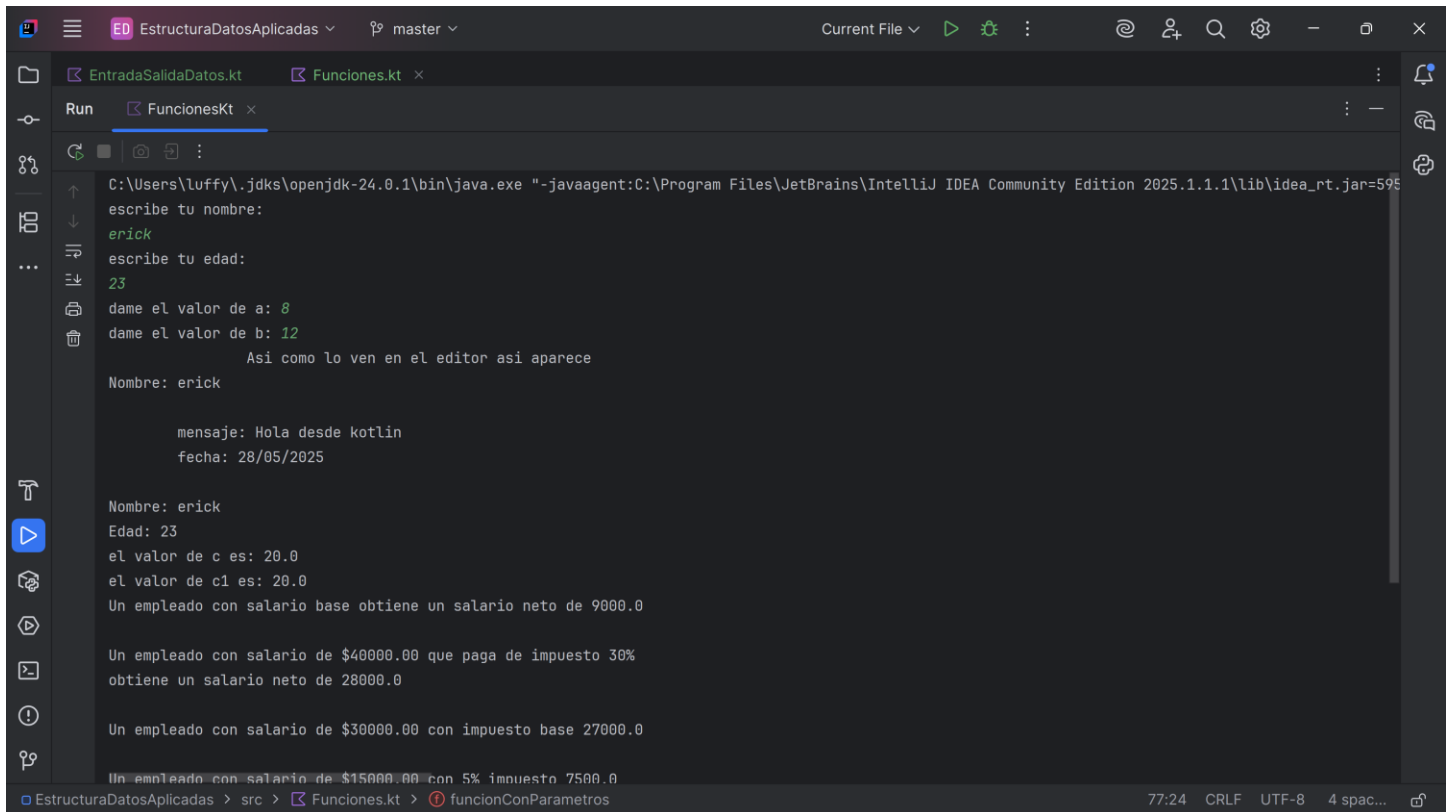
EntradaSalidaDatos.kt Funciones.kt x
75 fun funcionConParametros(nombre: String, age: Int) { 1 Usage new *
80 }
81
82 fun suma(a: Double, b: Double): Double { 1 Usage new *
83     return a + b
84 }
85
86 fun sumal(a: Double, b: Double): Double = a + b 1 Usage new *
87
88 fun calcularSalario(salario: Double = 10000.0, pctImp: Double = 0.10): Double { 4 Usages new *
89     val imp = salario * pctImp
90     return salario - imp
91 }
92
93 fun lambda() { 1 Usage new *
94     println(multi( x = 2, y = 3))
95
96     multiLambda( x = 2, y = 3, callback = { result -> println(result) })
97
98     multiLambda( x = 2, y = 3) { result -> println(result) }
99 }
100
101 fun multi(x: Int, y: Int): Int { 1 Usage new *
102     return x * y
103 }
104
105 fun multiLambda(x: Int, y: Int, callback: (result: Int) -> Unit) { 2 Usages new *
106     callback(y * x)
107 }
```

EstructuraDatosAplicadas > src > Funciones.kt > functionConParametros 77:24 CRLF UTF-8 4 spac...



```
86 fun sumai(a: Double, b: Double): Double = a + b 1 Usage new *
87
88 fun calcularSalario(salario: Double = 10000.0, pctImp: Double = 0.10): Double { 4 Usages new *
89     val imp = salario * pctImp
90     return salario - imp
91 }
92
93 fun lambda() { 1 Usage new *
94     println(multi(x = 2, y = 3))
95
96     multiLambda(x = 2, y = 3, callback = { result -> println(result) })
97
98     multiLambda(x = 2, y = 3) { result -> println(result) }
99 }
100
101 fun multi(x: Int, y: Int): Int { 1 Usage new *
102     return x * y
103 }
104
105 fun multiLambda(x: Int, y: Int, callback: (result: Int) -> Unit) { 2 Usages new *
106     callback(x * y)
107 }
108
```

EstructuraDatosAplicadas > src > Funciones.kt > functionConParametros 77:24 CRLF UTF-8 4 spac...



```
C:\Users\luffy\jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.1.1\lib\idea_rt.jar=595
escribe tu nombre:
erick
escribe tu edad:
23
dame el valor de a: 8
dame el valor de b: 12
    Así como lo ven en el editor así aparece
Nombre: erick

    mensaje: Hola desde kotlin
    fecha: 28/05/2025

Nombre: erick
Edad: 23
el valor de c es: 20.0
el valor de c1 es: 20.0
Un empleado con salario base obtiene un salario neto de 9000.0

Un empleado con salario de $40000.00 que paga de impuesto 30%
obtiene un salario neto de 28000.0

Un empleado con salario de $30000.00 con impuesto base 27000.0

Un empleado con salario de $15000.00 con 5% impuesto 7500.0
```

EstructuraDatosAplicadas > src > Funciones.kt > functionConParametros 77:24 CRLF UTF-8 4 spac...

EstructuraDatosAplicadas master

Current File

EntradaSalidaDatos.kt Funciones.kt

Run FuncionesKt

Nombre: erick  
Edad: 23  
el valor de c es: 20.0  
el valor de c1 es: 20.0  
Un empleado con salario base obtiene un salario neto de 9000.0  
  
Un empleado con salario de \$40000.00 que paga de impuesto 30%  
obtiene un salario neto de 28000.0  
  
Un empleado con salario de \$30000.00 con impuesto base 27000.0  
  
Un empleado con salario de \$15000.00 con 5% impuesto 7500.0  
6  
6  
6  
Hola  
Esto es un paso de parametro cadena  
fw = 5  
12  
x = 12  
  
Process finished with exit code 0

EstructuraDatosAplicadas > src > Funciones.kt > funcionConParametros 77:24 CRLF UTF-8 4 spac...

EstructuraDatosAplicadas master

Current File

IfElseWhen.kt

fun main () { new \*  
println("If else anidado")  
print("Escribe un nombre: ")  
val nombre = readln()  
if (nombre == "karina"){  
println("Hola Kary")  
} else if (nombre == "Pablo"){  
println("Hola Pablo")  
} else if (nombre == "Alain" || nombre == "Kotlin"){  
println("Hola AL")  
} else if (nombre == "Pamela"){  
println("Hola Pam")  
} else {  
println("Hola desconocido :)")  
}  
  
println("=== when ===")  
when (nombre) {  
"Karina"-> println("Hola Kary")  
"Pablo"-> println("Hola Pablin")  
"Alain", "Kotlin"-> println("Hola AL")  
"Pamela"-> println("Hola desconocido :)")  
}  
  
print("escribe tu edad: ")  
val edad = readln().toInt()  
when (edad) {  
in 0..5 -> println("Es un bebe")  
}

EstructuraDatosAplicadas > src > IfElseWhen.kt > main 72:12 CRLF UTF-8 4 spaces

```
ED EstructuraDatosAplicadas master
Current File
IfElseWhen.kt
1 fun main () { new *
28     in 0 ≤ .. ≤ 5 -> println("Es un bebe")
29     in 6 ≤ .. ≤ 12 -> println("Es menor de edad")
30     in 13 ≤ .. ≤ 17 -> println("Es un adolescente")
31     in 18 ≤ .. ≤ 29 -> println("Es un joven")
32     in 30 ≤ .. ≤ 49 -> println("Es un adulto")
33     in 50 ≤ .. ≤ 129 -> println("Es una persona de la tercera edad")
34     else-> println("no es humano")
35 }
36
37 when{
38     edad <= 5 -> println("Es un bebe")
39     edad <= 12 -> println("Es menor de edad")
40     edad <= 17 -> println("Es un adolescente")
41     edad <= 29 -> println("Es un joven")
42     edad <= 49 -> println("Es un adulto")
43     edad <= 129 -> println("Es una persona de la tercera edad")
44     else-> println("no es humano")
45 }
46
47 println("while")
48 var i = 1
49 while (i<=10){
50     println("$i")
51     i++
52 }
53 println("do while")
54 }
```

EstructuraDatosAplicadas > src > IfElseWhen.kt > main 72:12 CRLF UTF-8 4 spaces

```
ED EstructuraDatosAplicadas master
Current File
IfElseWhen.kt
1 fun main () { new *
54     i=1
55     do {
56         print("$i")
57         i++
58     }while (i<=10)
59     println("\nfor")
60     for(i in 1 ≤ .. ≤ 10)
61         print("$i")
62         i++
63     println("\nfor con incremento variable")
64     for (i in 1 ≤ .. ≤ 10 step 2) {
65         print("$i")
66     }
67     println("\nfor descendente")
68     for (i in 10 ≥ downTo ≥ 1){
69         print("$i")
70     }
71     println("\nfor descendente con decrementos")
72     for (i in 10 ≥ downTo ≥ 1 step 3){
73         print("$i")
74     }
75     println("\nforEach con rango it")
76     (1 ≤ .. ≤ 5).forEach {
77         print("$it")
78     }
79     println("\nforEach con rango renombrado")
80     (1 ≤ .. ≤ 5).forEach { numero ->
```

EstructuraDatosAplicadas > src > IfElseWhen.kt > main 72:12 CRLF UTF-8 4 spaces

ED EstructuraDatosAplicadas master

Current File

72:12 CRLF UTF-8 4 spaces

IfElseWhen.kt

```
1 fun main () { new *
// print("$it")
78 }
79 println("\nforEach con rango renombrado")
80 (1..5).forEach { numero ->
81     print("$numero")
82 }
83 println()
84 val cadena = "Jose De Jesus"
85 cadena.forEach { letra ->
86     print("$letra")
87 }
88 println()
89
90 val arreglo = arrayListOf("Hugo", "Paco", "Luis")
91 for (nombre in arreglo){
92     println(nombre)
93 }
94
95 arreglo.forEachIndexed { indice, valor ->
96     println("$indice = $valor")
97 }
98 }
```

EstructuraDatosAplicadas > src > IfElseWhen.kt > main

ED EstructuraDatosAplicadas master

Current File

72:12 CRLF UTF-8 4 spaces

IfElseWhen.kt

Run IfElseWhenKt

```
C:\Users\luffy\.jdfs\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.1.1\lib\idea_rt.jar=595
If else anidado
Escribe un nombre: erick
Hola desconocido :)
=== when ===
escribe tu edad: 23
Es un joven
Es un joven
while
1
2
3
4
5
6
7
8
9
10
do while
12345678910
for
12345678910
for con incremento variable
13579
```

EstructuraDatosAplicadas > src > IfElseWhen.kt > main

EstructuraDatosAplicadas master

Current File

Run IfElseWhenKt

IfElseWhen.kt

Run IfElseWhenKt

do while

12345678910

for

12345678910

for con incremento variable

13579

for descendente

10987654321

for descendente con decrementos

10741

forEach con rango it

12345

forEach con rango renombrado

12345

Jose De Jesus

Hugo

Paco

Luis

0 = Hugo

1 = Paco

2 = Luis

Process finished with exit code 0

EstructuraDatosAplicadas > src > IfElseWhen.kt > main 72:12 CRLF UTF-8 4 spaces

EstructuraDatosAplicadas master

Current File

Array.kt

Array.kt

fun main() { new \*

// For con step

println("for con step")

for (par in 2 .. 20 step 2) {

print("\$par, ")

}

// For con downTo

println("\n\nfor con downTo")

for (valor in 20 >= downTo 2 step 2) {

print("\$valor, ")

}

// For con variable it

println("\n\nfor con variable it")

(2 .. 25).forEach { it: Int ->

print("\$it ")

}

// For con rango renombrado

println("\n\nfor con rango renombrado")

(2 .. 25).forEach { numero ->

print("\$numero ")

}

// Recorriendo una cadena letra por letra

println("\n\nRecorriendo cadena:")

val oracion = "Hola amigos como la ven"

oracion.forEach { letra ->

print("\$letra ")

}

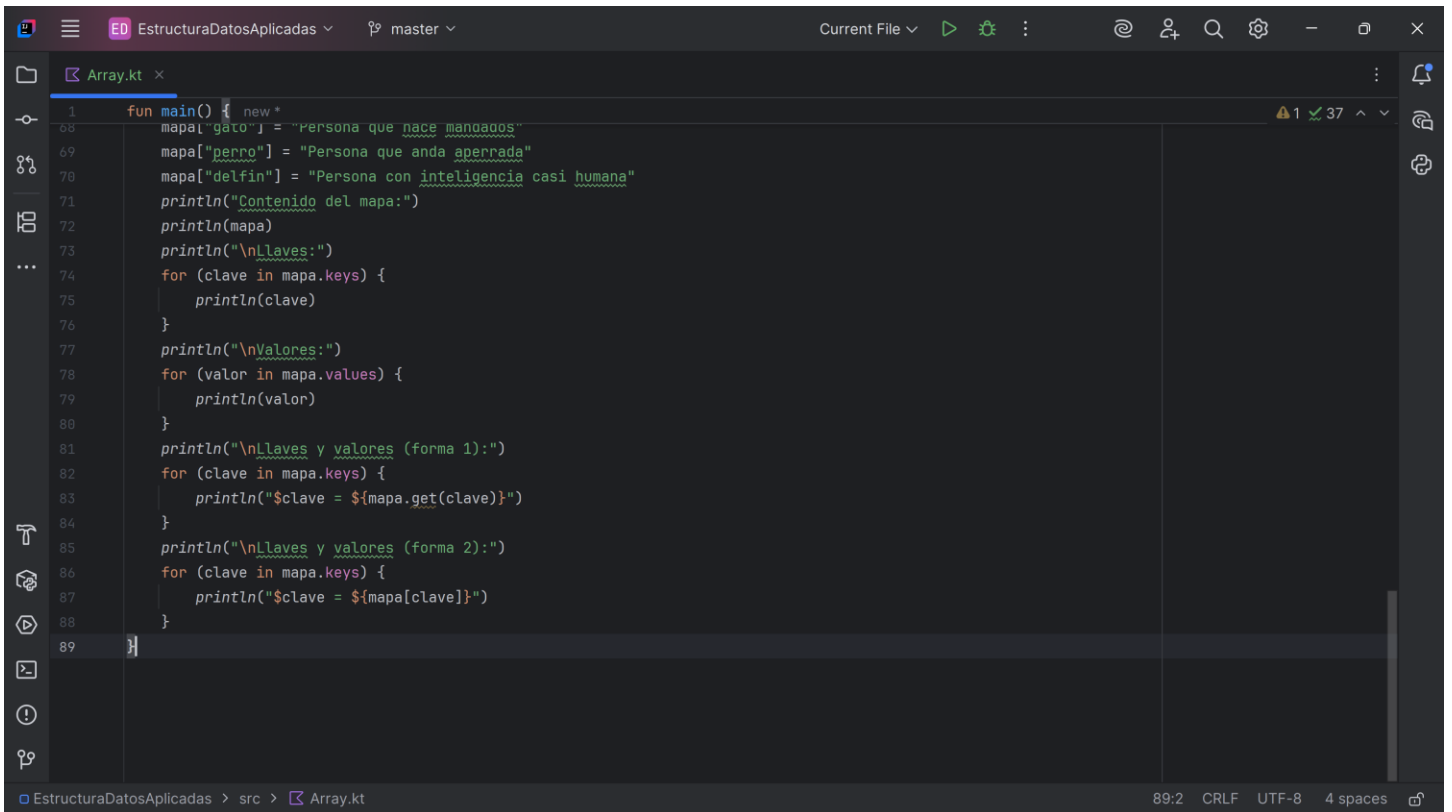
// lista simple

EstructuraDatosAplicadas > src > Array.kt 89:2 CRLF UTF-8 4 spaces



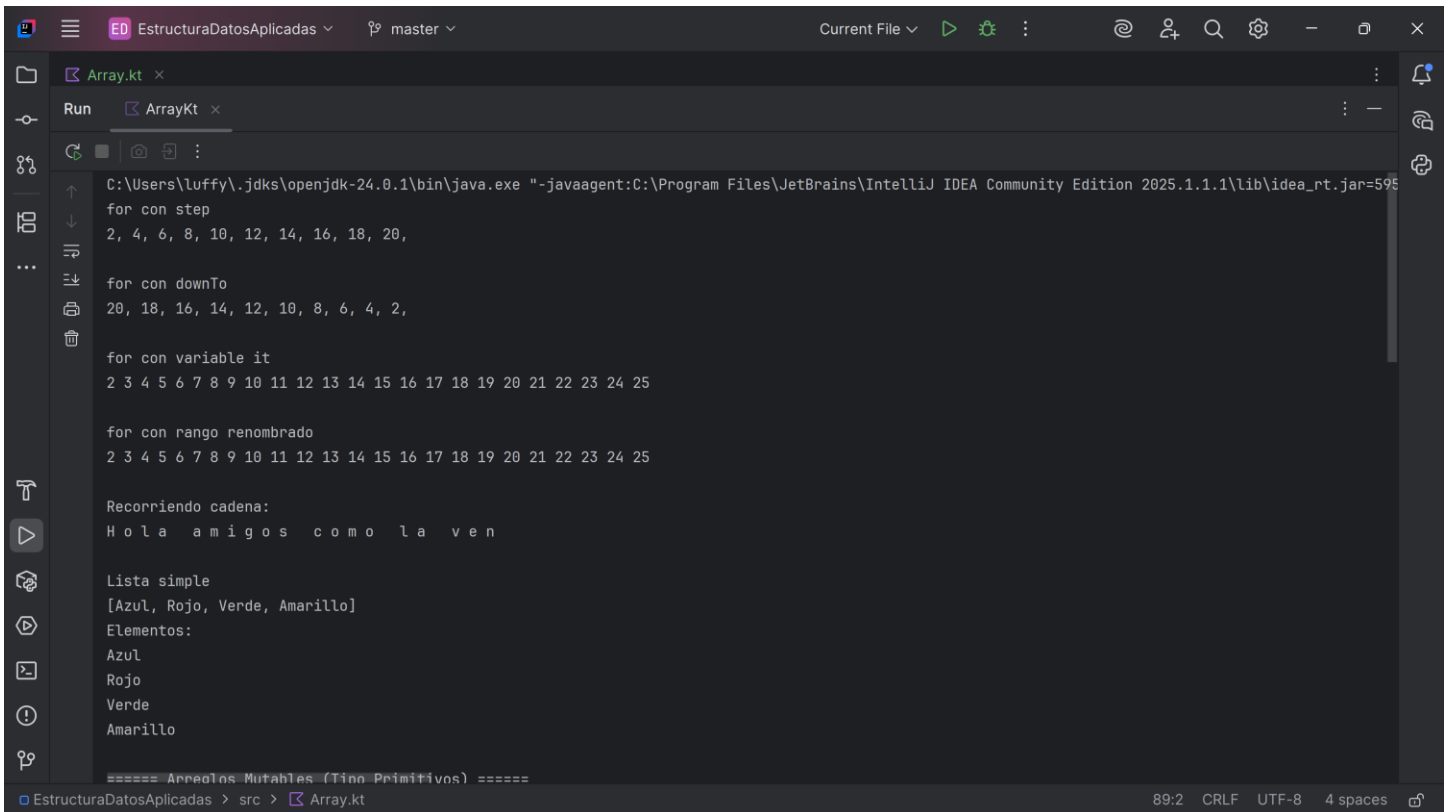
```
1 fun main() { new *
28 // Lista simple
29 println("\n\nLista simple")
30 val lista = listOf("Azul", "Rojo", "Verde", "Amarillo")
31 println(lista)
32 println("Elementos:")
33 for (n in lista) {
34     println(n)
35 }
36 // Arreglos mutables (tipo primitivos)
37 println("\n\n==== Arreglos Mutables (Tipo Primitivos) =====")
38 val arrInt = intArrayOf(54, -12, 89, 90, 3)
39 for (n in arrInt) {
40     println(n)
41 }
42 arrInt[1] = 1000
43 println("Una forma de recorrer el arreglo:")
44 for (i in 0..arrInt.size - 1) {
45     println("$i = ${arrInt[i]}")
46 }
47 println("Otra forma de recorrer el arreglo:")
48 (0..arrInt.size - 1).forEach { i ->
49     println("$i = ${arrInt[i]}")
50 }
51 // Lista mutable de Strings
52 println("\n\n==== Listas Mutables =====")
53 val mutable = mutableListOf("Hugo", "Paco", "Luis")
54 for (nombre in mutable) {
```

```
55     println(nombre)
56 }
57 mutable.add("José")
58 println("Después de add: $mutable")
59 mutable.removeAt(index = 1)
60 println("Después de removeAt(1): $mutable")
61 mutable.remove(element = "Luis")
62 println("Después de remove(\"Luis\"): $mutable")
63 mutable[0] = "Christian"
64 println("Después de set(0, \"Christian\"): $mutable")
65 // Mapas o diccionario
66 println("\n\n==== Mapas o Diccionario de Datos =====")
67 val mapa = mutableMapOf<String, String>()
68 mapa["gato"] = "Persona que hace mandados"
69 mapa["perro"] = "Persona que anda aperrada"
70 mapa["delfin"] = "Persona con inteligencia casi humana"
71 println("Contenido del mapa:")
72 println(mapa)
73 println("\n\nLlaves:")
74 for (clave in mapa.keys) {
75     println(clave)
76 }
77 println("\n\nValores:")
78 for (valor in mapa.values) {
79     println(valor)
80 }
81 println("\n\nLlaves y valores (forma 1):")
```



```
1 fun main() { new +
68     mapa["gato"] = "Persona que nace mandados"
69     mapa["perro"] = "Persona que anda aperrada"
70     mapa["delfin"] = "Persona con inteligencia casi humana"
71     println("Contenido del mapa:")
72     println(mapa)
73     println("\nLlaves:")
74     for (clave in mapa.keys) {
75         println(clave)
76     }
77     println("\nValores:")
78     for (valor in mapa.values) {
79         println(valor)
80     }
81     println("\nLlaves y valores (forma 1):")
82     for (clave in mapa.keys) {
83         println("$clave = ${mapa.get(clave)}")
84     }
85     println("\nLlaves y valores (forma 2):")
86     for (clave in mapa.keys) {
87         println("$clave = ${mapa[clave]}")
88     }
89 }
```

EstructuraDatosAplicadas > src > Array.kt 89:2 CRLF UTF-8 4 spaces



```
C:\Users\luffy\.jdk\openjdk-24.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.1.1.1\lib\idea_rt.jar=595
for con step
2, 4, 6, 8, 10, 12, 14, 16, 18, 20,

for con downTo
20, 18, 16, 14, 12, 10, 8, 6, 4, 2,

for con variable it
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

for con rango renombrado
2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Recorriendo cadena:
H o l a   a m i g o s   c o m o   l a   v e n

Lista simple
[Azul, Rojo, Verde, Amarillo]
Elementos:
Azul
Rojo
Verde
Amarillo

===== Arreglos Mutables (Tipo Primitivos) =====
```

EstructuraDatosAplicadas > src > Array.kt 89:2 CRLF UTF-8 4 spaces

```
==== Arreglos Mutables (Tipo Primitivos) ====
54
-12
89
90
3
Una forma de recorrer el arreglo:
0 = 54
1 = 1000
2 = 89
3 = 90
4 = 3
Otra forma de recorrer el arreglo:
0 = 54
1 = 1000
2 = 89
3 = 90
4 = 3

==== Listas Mutables ====
Hugo
Paco
Luis
Después de add: [Hugo, Paco, Luis, José]
Después de remove("Luis"): [Hugo, Paco, José]
```

```
Después de remove("Luis"): [Hugo, José]
Después de set(0, "Christian"): [Christian, José]

==== Mapas o Diccionario de Datos ====
Contenido del mapa:
{gato=Persona que hace mandados, perro=Persona que anda aperrada, delfin=Persona con inteligencia casi humana}

Llaves:
gato
perro
delfin

Valores:
Persona que hace mandados
Persona que anda aperrada
Persona con inteligencia casi humana

Llaves y valores (forma 1):
gato = Persona que hace mandados
perro = Persona que anda aperrada
delfin = Persona con inteligencia casi humana

Llaves y valores (forma 2):
gato = Persona que hace mandados
perro = Persona que anda aperrada
```

ED EstructuraDatosAplicadas master

Current File

Array.kt

Run ArrayKt

↑

↓

↕

↕

🗑

```
{gato=Persona que hace mandados, perro=Persona que anda aperrada, delfin=Persona con inteligencia casi humana}

Llaves:
gato
perro
delfin

Valores:
Persona que hace mandados
Persona que anda aperrada
Persona con inteligencia casi humana

Llaves y valores (forma 1):
gato = Persona que hace mandados
perro = Persona que anda aperrada
delfin = Persona con inteligencia casi humana

Llaves y valores (forma 2):
gato = Persona que hace mandados
perro = Persona que anda aperrada
delfin = Persona con inteligencia casi humana

Process finished with exit code 0
```

EstructuraDatosAplicadas > src > Array.kt 89:2 CRLF UTF-8 4 spaces