



24 DE SEPTIEMBRE DE 2023



# OPERADORES

## PROGRAMACIÓN MÓVIL

EDSON ALEJANDRO CABRERA ACEVEDO

6NM61

## Variables y Tipos de Datos

Instrucciones:

La ecuación de la pendiente se obtiene por la siguiente expresión:

1. Expresarla ahora con operadores.
  - Resolver el valor de la pendiente, tomando en cuenta que: P1(4,3), P2(-3,-2)
  - Del ejercicio anterior, responder: ¿Por qué el resultado obtenido difiere del resultado esperado?
  - Encontrar una solución para que se refleje el resultado.

Anexa tu código al mismo repositorio que has trabajado, cargas tus evidencias en un formato PDF.

Problema planteado

```
const val P1_X = 4
const val P1_Y = 3
const val P2_X = -3
const val P2_Y = -2
new *
fun main(){
    val pendiente = (P2_X-P1_X)/(P2_Y-P1_Y)
    println("La pendiente es: $pendiente")
}
```

Resultado

```
"C:\Program Files\jdk-20.0.2\bin\java.exe"
La pendiente es: 1

Process finished with exit code 0
```

¿Por qué el resultado obtenido difiere del resultado esperado?

*R: Al expresar la fórmula de pendiente de esta manera y el uso de datos enteros, el resultado se ve redondeado al valor más cercano en este caso 1.*

$$m = \frac{x_2 - x_1}{y_2 - y_1} = 0.7142 \approx 1$$

## Solución al Problema

```
1 //esta clase tiene dos propiedades: 'X' y 'Y', que representan coordenadas
  new *
2 data class Punto(val x: Double, val y: Double)
3 //Pendiente (m)= //x2-x1
4 // -----
5 //           y2-y1
  new *
6 fun calcularPendiente(p1: Punto, p2: Punto): Double = (p2.y - p1.y) / (p2.x - p1.x)
7
  new *
8 fun main() {
9     //creacion de objetos
10    val punto1 = Punto(x: 4.0, y: 3.0)
11    val punto2 = Punto(x: -3.0, y: -2.0)
12    //resultado
13    val pendiente = calcularPendiente(punto1, punto2)
14    println("La pendiente es: $pendiente")
15 }
```

## Resultado

```
"C:\Program Files\jdk-20.0.2\bin\java.exe"
La pendiente es: 0.7142857142857143

Process finished with exit code 0
```

## KotlinPlayground

```
//esta clase tiene dos propiedades: 'X' y 'Y', que representan coordenadas
data class Punto(val x: Double, val y: Double)
//Pendiente (m)= //x2-x1
// -----
//           y2-y1
fun calcularPendiente(p1: Punto, p2: Punto): Double = (p2.y - p1.y) / (p2.x - p1.x)

fun main() {
    //creacion de objetos
    val punto1 = Punto(4.0, 3.0)
    val punto2 = Punto(-3.0, -2.0)
    //resultado
    val pendiente = calcularPendiente(punto1, punto2)
    println("La pendiente es: $pendiente")
}
```

```
La pendiente es: 0.7142857142857143
```

eyJ2ZXJzaW9uIjoiaS45LjEwIiwicGxhdGZvcml  
iOiJqYXZlIiwiaXJncyI6IiIsIm5vbmVNYXJrZX  
JzIjp0cnVlLCJ0aGVtZSI6Im1kZWElLCJjb2RRI  
joilY9lc3RhIGNsYXNlIHRpZW51IGRvcyBwcm9w  
aWVkaWYWRlczogJ1gnIHkgJ1knLCBxdWUgcmlVwcmV  
zZW50YW4gY29vcmlRbmFkYXNcbmRhdGEgY2xhc3  
MUgHVudG8odmFsIHg6IERvdWJsZSwgdG9sIHk6I  
ERvdWJsZS1cbi8vUGVudGZlbnRlIChtKT07glY94  
MuKIknngxXG4vLyAgICAgICAgICAgICAgICAtLS0  
tLS1cbi8vICAgICAgICAgICAgICAgICAgICB5MuKIkn  
kxXG5mdW4gY2FsY3VsYXJQZW5kaWVudGUocDE6I  
FB1bnRvLCBwMjogUHVudG8p0iEBE3VibUGUGPSAo  
cDIueSATtHAXLnkpcIC8gKHAyLnggLSBwMS4KVx  
uXG5mdW4gBWfPbipgIHtcbiAgICAvL2NyZWZjaW  
9uIGRIIG9iamV0b3NcbiAgICB2YWwgcHvudG8xI  
D0gUHVudG8oNC4wLCAZLjApXG4gICAgdmFsIHBB  
bnRvMIA9IFB1bnRvKC0zLjAsIC06YlJApXG4gICA  
gly9yZXTNtIHRHRCG9cbzIjAgICB2YWwgcGZlbnR  
lRlID0gY2FsY3VsYXJQZW5kaWVudGUocHvudG8xL

```

HEADER: ALGORITHM & TOKEN TYPE

{
  "version": "1.9.10",
  "platform": "java",
  "args": "",
  "noneMarkers": true,
  "theme": "idea",
  "code": "///esta clase tiene dos propiedades: 'X' y
'Y', que representan coordenadas\ndata class Punto(val
x: Double, val y: Double)\n//Pendiente (m)= //x2-x1\n//
-----\n//          y2-y1\nfun
calcularPendiente(p1: Punto, p2: Punto): Double = (p2.y
- p1.y) / (p2.x - p1.x)\n\nfun main() {\n    //creacion
de objetos\n    val punto1 = Punto(4.0, 3.0)\n    val
punto2 = Punto(-3.0, -2.0)\n    //resultado\n    val
pendiente = calcularPendiente(punto1, punto2)\n    println(\\\"La pendiente es: $pendiente\\\")\n  }
}

PAYLOAD: DATA

{}

VERIFY SIGNATURE

```