



INSITUTO POLITÉCNICO **NACIONAL**



Unidad Profesional Interdisciplinaria de
Ingeniería y Ciencias Sociales y Administrativas

“Tarea 1: Variables y Tipos de Datos”

Unidad de aprendizaje: Programación Móvil

Secuencia: 6NM61

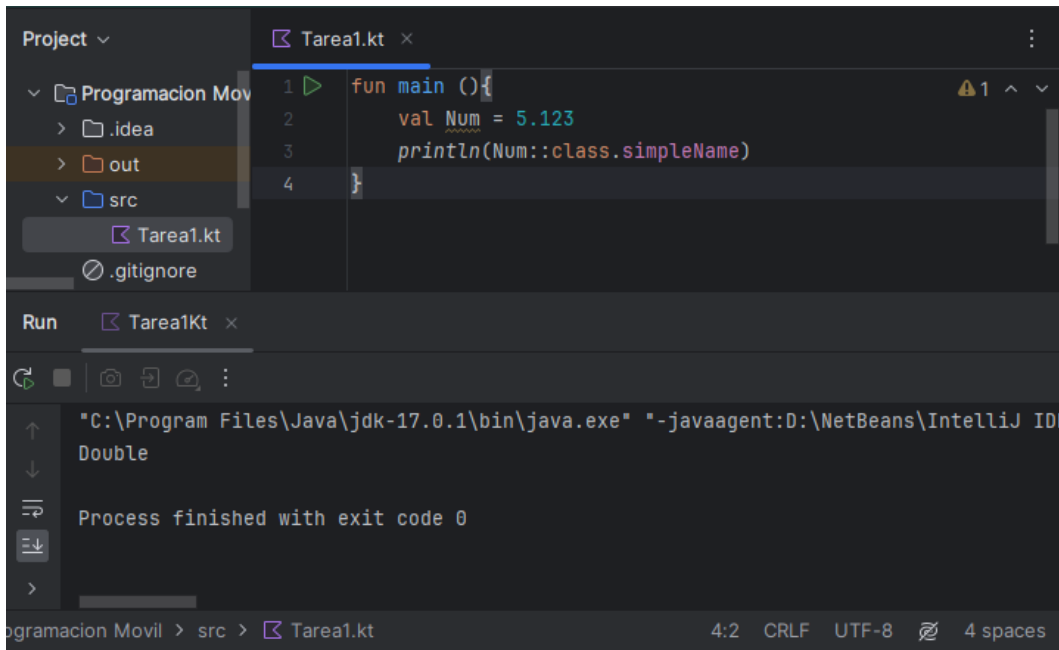
Alumno: Martínez Arellano Luis Aarón

Boleta: 2022601889

Profesor(a): Bueno Vasquez Francisco Javier

Realizar los siguientes puntos:

1. Para un número decimal, qué tipo de dato se asigna por defecto? Imprimirlo.



```
1 fun main () {  
2     val Num = 5.123  
3     println(Num::class.simpleName)  
4 }
```

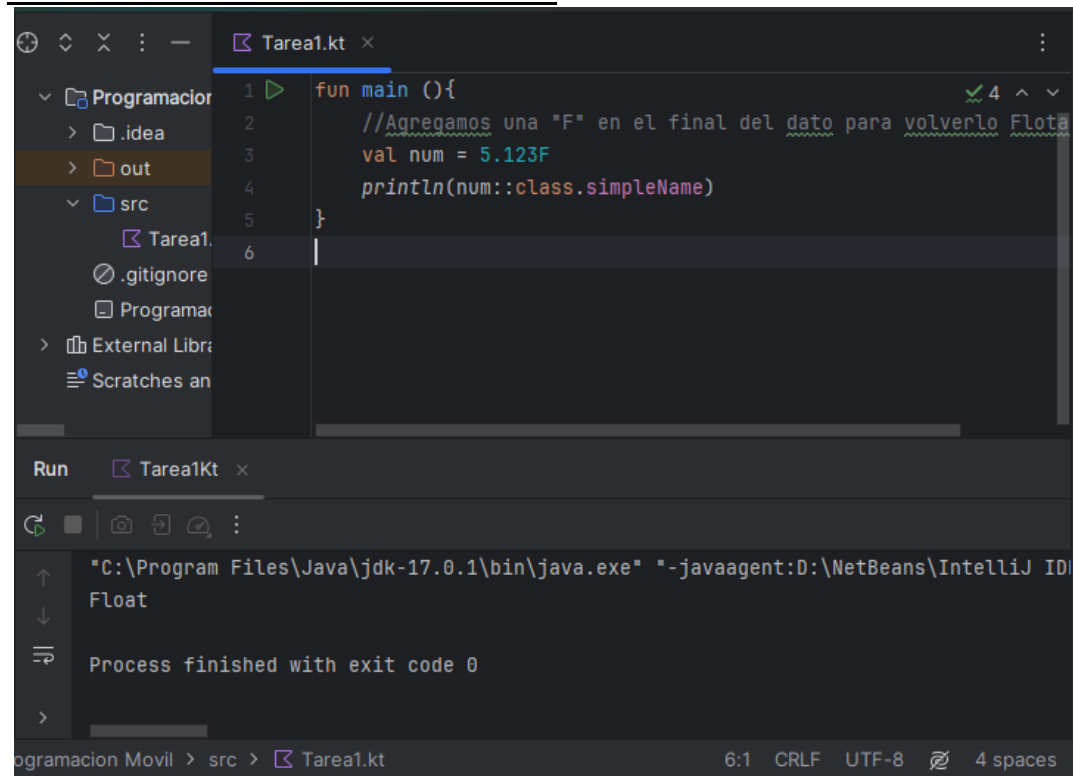
Run Tarea1Kt

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:D:\NetBeans\IntelliJ IDI
Double
Process finished with exit code 0

Programacion Movil > src > Tarea1.kt 4:2 CRLF UTF-8 4 spaces

Respuesta: Double

2. Volver flotante a la variable decimal.



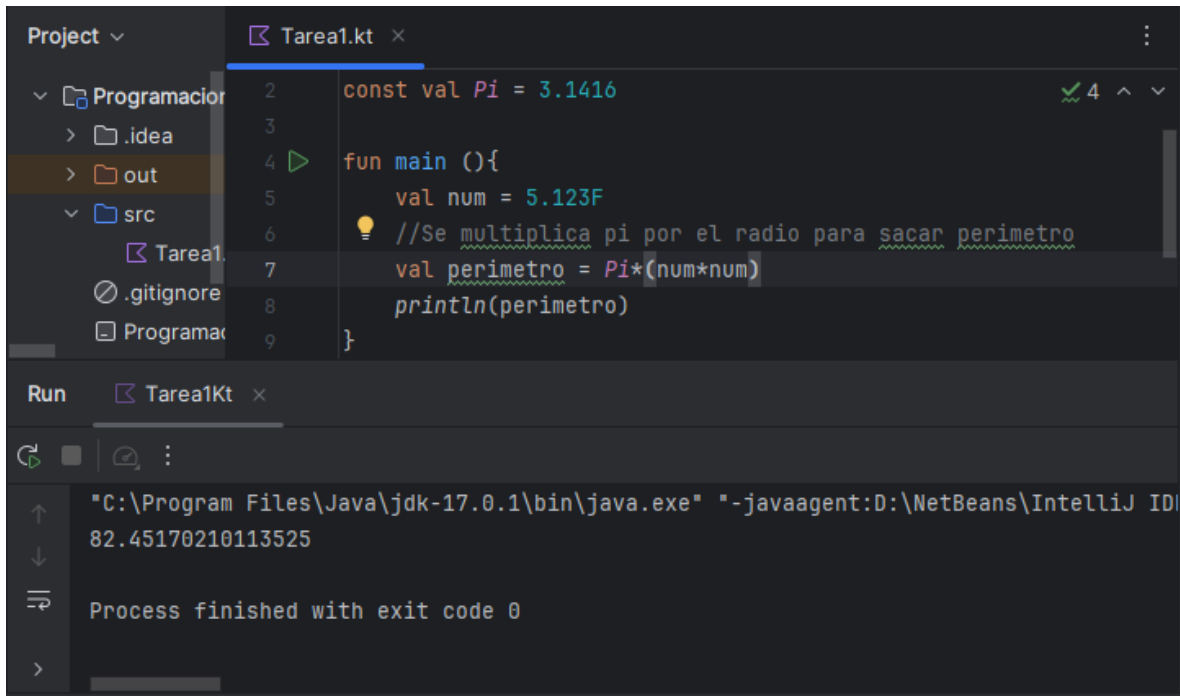
```
1 fun main () {  
2     //Agregamos una "F" en el final del dato para volverlo Flota  
3     val num = 5.123F  
4     println(num::class.simpleName)  
5 }  
6
```

Run Tarea1Kt

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:D:\NetBeans\IntelliJ IDI
Float
Process finished with exit code 0

Programacion Movil > src > Tarea1.kt 6:1 CRLF UTF-8 4 spaces

3. Declarar una variable const val con el valor de PI y multiplicarla por 2 veces nuestra variable decimal para sacar el perímetro de un círculo.



```
Project ▾ Tarea1.kt ×
  ▾ Programacion
    > .idea
    > out
    ▾ src
      Tarea1.kt
      .gitignore
      Programacion.kt

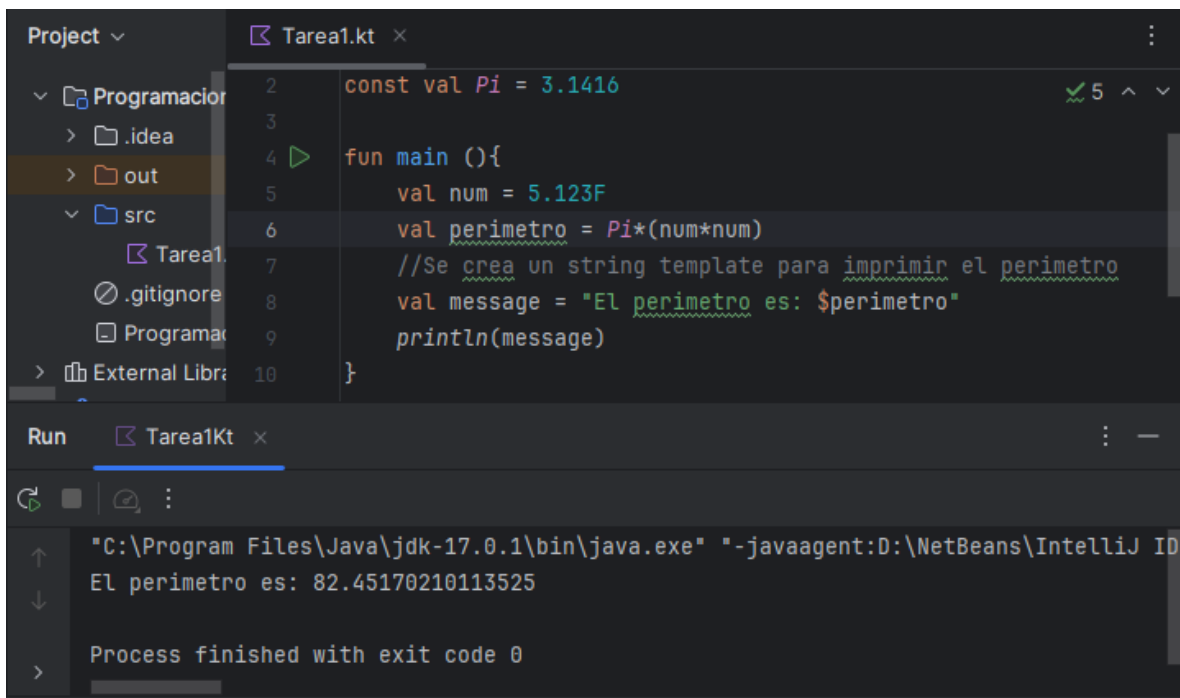
2  const val Pi = 3.1416
3
4  fun main (){
5      val num = 5.123F
6      //Se multiplica pi por el radio para sacar perimetro
7      val perimetro = Pi*(num*num)
8      println(perimetro)
9  }
```

Run Tarea1Kt ×

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:D:\NetBeans\IntelliJ IDI
82.45170210113525

Process finished with exit code 0

4. Para el ejemplo anterior, utilizar un String Template para imprimir el texto El perímetro del círculo es: [resultado] siendo resultado nuestra variable perímetro.



```
Project ▾ Tarea1.kt ×
  ▾ Programacion
    > .idea
    > out
    ▾ src
      Tarea1.kt
      .gitignore
      Programacion.kt
    > External Libr...

2  const val Pi = 3.1416
3
4  fun main (){
5      val num = 5.123F
6      val perimetro = Pi*(num*num)
7      //Se crea un string template para imprimir el perimetro
8      val message = "El perimetro es: $perimetro"
9      println(message)
10 }
```

Run Tarea1Kt ×

"C:\Program Files\Java\jdk-17.0.1\bin\java.exe" "-javaagent:D:\NetBeans\IntelliJ IDI
El perimetro es: 82.45170210113525

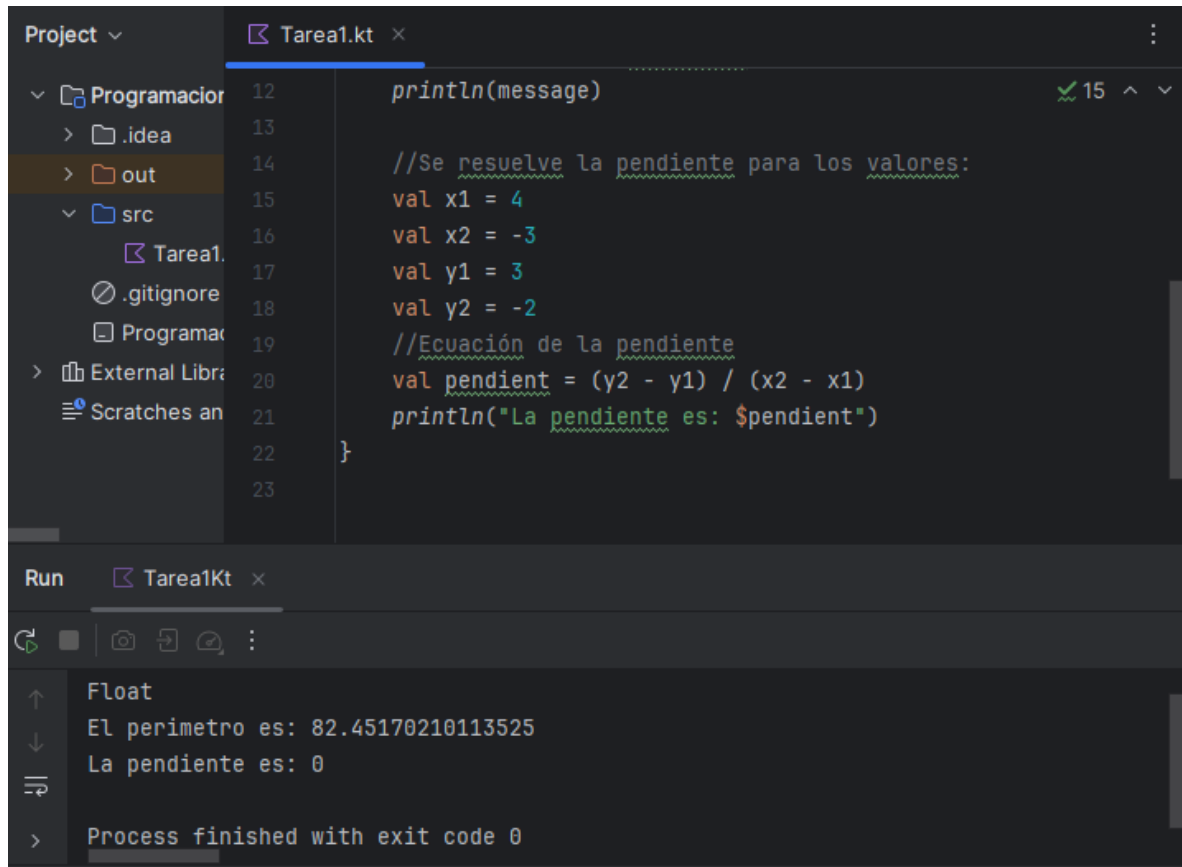
Process finished with exit code 0

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

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

5. Expresarla ahora con operadores.

6. Resolver el valor de la pendiente, tomando en cuenta que P1(4,3), P2(-3,-2)



```
Project ▾
  ▾ Programaci...
    ▾ .idea
    ▾ out
    ▾ src
      Tarea1.kt
      .gitignore
      Programaci...
    ▾ External Libr...
    ▾ Scratches an...

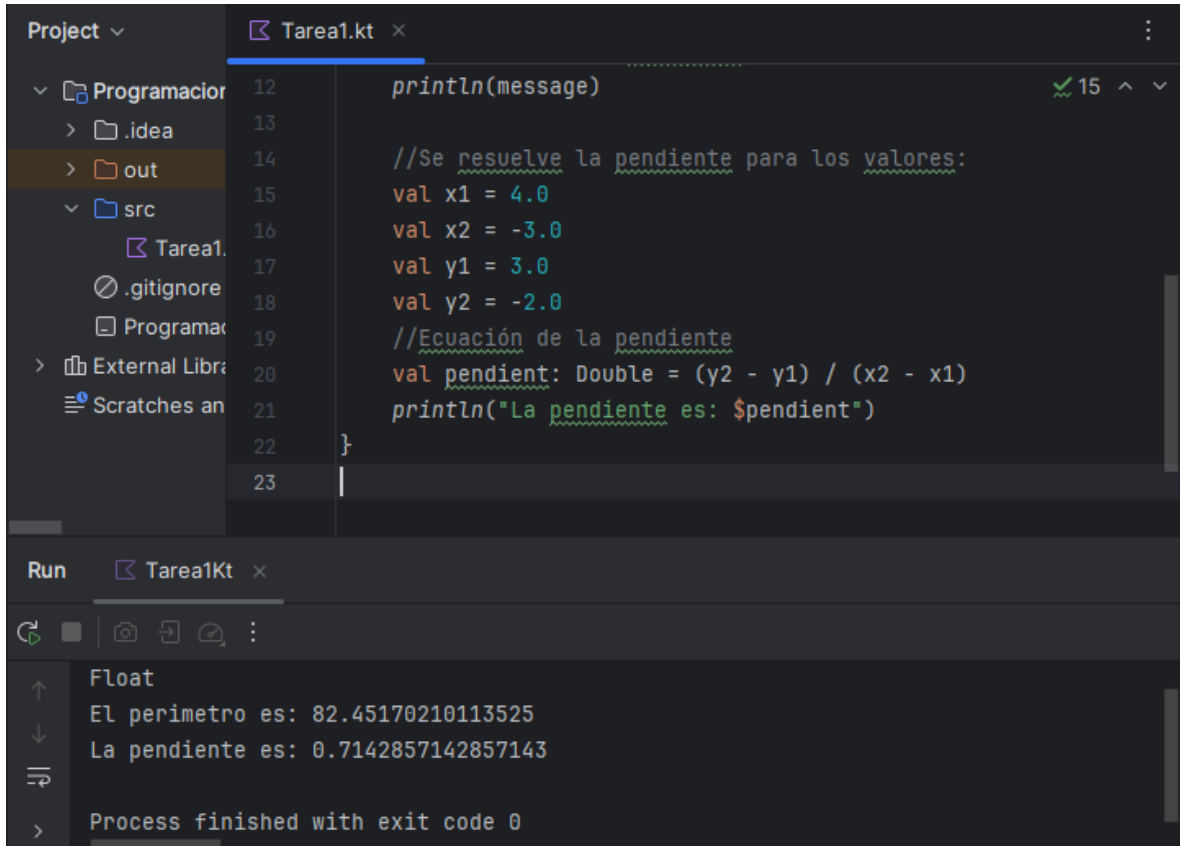
Tarea1.kt x
12 println(message)
13
14 //Se resuelve la pendiente para los valores:
15 val x1 = 4
16 val x2 = -3
17 val y1 = 3
18 val y2 = -2
19 //Ecuación de la pendiente
20 val pendiente = (y2 - y1) / (x2 - x1)
21 println("La pendiente es: $pendiente")
22 }
23

Run Tarea1Kt x
Float
El perimetro es: 82.45170210113525
La pendiente es: 0
Process finished with exit code 0
```

7. Del ejercicio anterior, responder: ¿Por qué el resultado obtenido difiere del resultado esperado?

Difiere porque en los datos que nosotros ingresamos no especificamos que deberían ser tipo float o en su caso double, por lo tanto como por defecto toma la variable entera el resultado que nos arroja es cero.

8. Encontrar una solución para que se refleje el resultado.



The screenshot shows an IDE with a project named 'Programacion'. The file 'Tarea1.kt' is open, showing Kotlin code. The code calculates the slope of a line given two points (x1, y1) and (x2, y2). The variables are declared as Double. The output of the program is shown in the Run console, displaying the perimeter and the slope.

```
println(message)

//Se resuelve la pendiente para los valores:
val x1 = 4.0
val x2 = -3.0
val y1 = 3.0
val y2 = -2.0
//Ecuación de la pendiente
val pendiente: Double = (y2 - y1) / (x2 - x1)
println("La pendiente es: $pendiente")
}
```

Run Tarea1Kt

Float
El perimetro es: 82.45170210113525
La pendiente es: 0.7142857142857143
Process finished with exit code 0

Para que se nos muestre el resultado correcto debemos declarar las variables con puntos decimales y la variable en que se guarde el resultado de igual forma declarar lo de tipo float o double.

Código Fuente:

```
const val PI = 3.1416

fun main () {
    val num = 5.123F
    println(num::class.simpleName)

    //Se multiplica PI por el radio para sacar el perimetro
    val perimetro = PI*(num*num)

    //Se crea un String template para imprimir el perimetro
    val message = "El perimetro es: $perimetro"
    println(message)

    //Se resuelve la pendiente para los valores:
    val x1 = 4.0
    val x2 = -3.0
    val y1 = 3.0
    val y2 = -2.0
    //Ecuación de la pendiente
    val pendiente: Double = (y2 - y1) / (x2 - x1)
    println("La pendiente es: $pendiente")
}
```