



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co



# Kotlin

## ¿Qué es Kotlin?

Kotlin es un lenguaje de programación estático de código abierto que admite la programación funcional y orientada a objetos.

Google es compatible oficialmente con Kotlin para el desarrollo de Android, lo cual significa que la documentación y las herramientas de Android se diseñaron para admitir Kotlin. Puedes encontrar la documentación oficial de Kotlin en <https://kotlinlang.org>. Si deseas obtener acceso a libros recomendados, consulta [Libros](#).

El lenguaje por excelencia para Android es Java, sin embargo, Kotlin es la respuesta a las solicitudes de muchos desarrolladores de la comunidad de Android como complemento del lenguaje de Java. Esto quiere decir, básicamente que podemos desarrollar nuestras aplicaciones tanto en Java como en Kotlin o en ambos, si queremos !!!

## Características

1. Moderno y conciso: Es una sintaxis más simple que la de Java, permitiendo reducción de código, y una curva de aprendizaje mucho más rápida como para los que ya vienen de Java o para los que están iniciando desde cero.
2. Interoperatividad con Java: Como lo indiqué en la parte de arriba ↑ se integran sin problema, dado que ambos corren en la máquina virtual de Java.
3. Código más seguro: Con @Nullable y @NonNull incluidos en su sistema de tipos, Kotlin lo ayuda a evitar NullPointerExceptions. Las aplicaciones de Android que usan Kotlin tienen un 20 % menos de probabilidades de fallar.
4. Concurrencia estructurada: Las corrutinas de Kotlin agilizan la programación asíncrona, haciendo que tareas comunes como llamadas de red y actualizaciones de bases de datos sean simples y eficientes.

## ¿Se debe saber Java para aprender Kotlin?

Los dos lenguajes son independientes, con varias diferencias, sin embargo, aun que su sintaxis es diferente, tienen similitudes a nivel de estructura y de los flujos a la hora de desarrollar aplicaciones. Por lo que aun que tener bases en Java es importante, no es necesario a la hora de aprender Kotlin.

## Editores de código Online

Si bien es cierto, que podemos iniciar directamente en Android Studio o Visual Studio Code, para esta primera fase daremos inicio con editores de código online.

Seleccionen el de su preferencia:

- [Kotlin Playground: Edit, Run, Share Kotlin Code Online](#)

- <https://replit.com/>
- [Online Kotlin Compiler](#)

Variables y tipos de datos

- Número
  - Enteros

Tipo	Valor mínimo	Valor máximo
Byte	-128	127
Short	-32768	32767
Int	-2,147,483,648 (-2 <sup>31</sup> )	2,147,483,647 (2 <sup>31</sup> - 1)
Long	-9,223,372,036,854,775,808 (-2 <sup>63</sup> )	9,223,372,036,854,775,807 (2 <sup>63</sup> - 1)

Kotlin

SolutionsDocsCommunityTeachPlay

1.8.21JVMProgram arguments

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    val uno = 1 // Int
    val tresbillones = 3000000000 // Long
    val unlong = 1L // Long
    val unbyte: Byte = 1
```

- Reales

Tipo	Bits de exponente	Dígitos decimales
Float	8	6-7
Double	11	15-16

Kotlin

SolutionsDocsCommunityTeachPlay

1.8.21JVMProgram arguments

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    val a = 2.7182818284 // Double
    val efloat = 2.7182818284f // Float, valor actual es 2.7182817
```

- Booleanos

El tipo **Boolean** representa objetos booleanos que pueden tener dos valores: true y false.

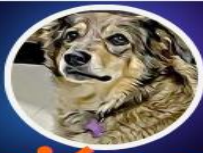
Boolean tiene una contraparte anulable **Boolean?** que también tiene el null valor.

Las operaciones integradas en booleanos incluyen:

||– disyunción ( OR lógico )

&&– conjunción ( AND lógico )

!– negación ( NO lógico )



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    val mytrue : Boolean = true
    val myfalse : Boolean = false
    ! val boolNull : Boolean ? = null

    println(mytrue || myfalse)
    println(mytrue && myfalse)
    println(! mytrue)

}

true
false
false
```

- Caracteres - Char

Los caracteres están representados por el tipo Char. Los caracteres literales van entre comillas simples: '1'.

- Cadena de caracteres - String

Las cadenas en Kotlin están representadas por el tipo String. Generalmente, un valor de cadena es una secuencia de caracteres entre comillas dobles ( " ):

```
val str = "abcd 123"
```

```
Kotlin Solutions Docs Community Teach Play
1.8.21 JVM Program arguments

/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    val str = "abcd"
    println(str.uppercase()) // Crear e imprimir un nuevo objeto de String
    println(str) // La cadena original sigue siendo la misma

}

ABCD
abcd
```



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

Para concatenar cadenas, utilice el **+** operador. Esto también funciona para concatenar cadenas con valores de otros tipos, siempre que el primer elemento de la expresión sea una cadena:

```
Kotlin Solutions Docs Community Tea

1.8.21 JVM Program arguments

/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    val s = "abc" + 1
    println(s + "def") //

}
```

- Arreglos – Array

Las matrices en Kotlin están representadas por la clase **Array**.

Para crear una matriz, use la función **arrayOf()** y pásele los valores de los elementos, de modo que **arrayOf(1, 2, 3)** crea una matriz **[1, 2, 3]**.

```
Kotlin Solutions Docs Community

1.8.21 JVM Program arguments

/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

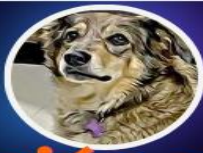
    val asc = arrayOf (1, 2, 3)
    asc . forEach {println(it)}

}

1
2
3
```

## TIPS ¡!

1. **Var** : Hace referencia a variables mutables, es decir que pueden cambiar.
2. **Val** : Hace referencia a variables inmutables, es decir que son constantes, una vez se declaran no se pueden cambiar.
3. Cuando el nombre de una variable se compone de dos palabras, la primera palabra arranca en minúscula y la segunda en mayúscula para que el código sea mas entendible y organizado ejemplo:



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

```
var numeroEntero: Int=24
```

## Casting

En kotlin no se le puede hacer **casting** a los datos, es decir aun tipo de dato **INT** pasarle un tipo de dato **Byte**, dado que son diferentes, por lo que se realizaría lo siguiente teniendo en cuenta que en Kotlin todos los datos se consideran objetos:

```
1.8.21 ▾ JVM ▾ Program arguments Copy link <> Share code Run

/**
 * Demostraciones para la guía 1. introducción a Kotlin
 */
fun main() {

    var numeroA: Byte=12
    var edad: Int=numeroA // Error!
    var edad: Int=numeroA.toInt() //Estamos convirtiendo en tipo de dato Byte a entero

}
```

## Concepto de diferencia de tipos

Básicamente lo que nos dice el concepto de diferencia de tipos es que, nosotros como desarrolladores no estamos obligados a declarar el tipo de datos a utilizar en Kotlin:

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin
 */
fun main() {

    var numeroA: Byte=12
    var numeroA:12 |

    var nombre: String ="Eliana"
    var nombre="Eliana"

}
```

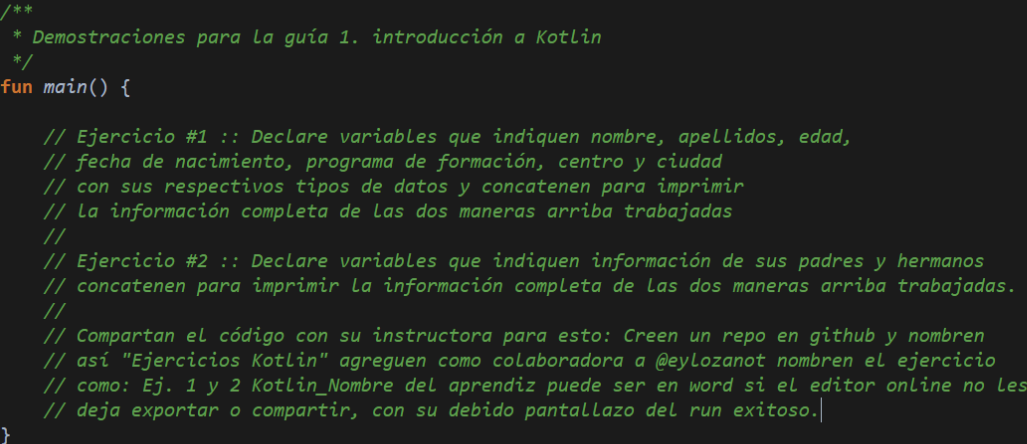
Kotlin está tan fuertemente equipado que el determina a que tipo de dato se le asigna el dato de la variable, sin embargo, declarar el tipo de datos por estructura, por mantenimiento del código mismo y por organización es lo ideal.



Podemos realizarlo de las siguientes dos formas:

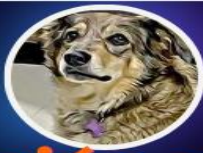
Hola su nombre es: Eliana, su edad es: 31 años y pertenece al centro: CADPH de la ciudad de Garzón  
 Hola su nombre es: Eliana, su edad es 31 años y pertenece al centro: CADPH de la ciudad de Garzón

**\*\*\* Ejercicios en clase ¡! \*\*\***



Básicamente nos estamos refiriendo a suma, resta, multiplicación, división y módulo.

Ejemplos de operadores aritméticos SUMA  
El resultado de la suma de  $8 + 10 = 18$



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

## RESTA

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    println("Ejemplos de operadores aritméticos RESTA")
    var num1=8
    var num2 = 10
    var res=0

    res=num2-num1

    println("El resultado de la resta de $num2 - $num1 = $res")
}
```

Ejemplos de operadores aritméticos RESTA  
El resultado de la resta de 10 - 8 = 2

## MULTIPLICACIÓN

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

    println("Ejemplos de operadores aritméticos MULTIPLICACIÓN")
    var num1=8
    var num2 =18
    var res=0

    res=num1*num2

    println("El resultado de la multiplicación de $num1 * $num2 = $res")
}
```

Ejemplos de operadores aritméticos MULTIPLICACIÓN  
El resultado de la multiplicación de 8 \* 18 = 144

## DIVISIÓN

```
/**
 * Demostraciones para La guía 1. introducción a Kotlin
 */
fun main() {

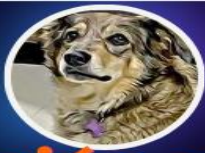
    println("Ejemplos de operadores aritméticos DIVISIÓN")
    var num1: Double =8.0
    var num2:Double =18.0
    var res: Double=0.0

    res=num2/num1

    println("El resultado de la división de $num2 / $num1 = $res")
}
```

Ejemplos de operadores aritméticos DIVISIÓN  
El resultado de la división de 18.0 / 8.0 = 2.25





# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

## MÓDULO

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin
 */
fun main() {

    println("Ejemplos de operadores aritméticos MÓDULO")
    var num1=8
    var num2=18
    var res=0

    res=num1%num2

    println("El resultado del módulo de $num1 % $num2 = $res")
}
```

Ejemplos de operadores aritméticos MÓDULO  
El resultado del módulo de 8 % 18 = 8

## PROMEDIO

```
/**
 * Demostraciones para la guía 1. introducción a Kotlin
 */
fun main() {

    println("Ejemplos de operadores aritméticos PROMEDIO")
    var num1: Double =8.0
    var num2: Double =18.0
    var num3=3.0

    var prom=(num1 + num2 + num3)/3

    println("El promedio de \nnum1=$num1 \nnum2=$num2 \nnum3=$num3 \nes $prom")
    // Donde \n nos permite hacer saltos de línea
}
```

Ejemplos de operadores aritméticos PROMEDIO  
El promedio de  
num1=8.0  
num2=18.0  
num3=3.0  
es 9.666666666666666

## Operadores unarios

++ (Aumentos de 1 en 1) o -- (Resta de 1 en 1)

Prefijo: Aumentar antes de la variable  
Postfijo: Aumenta después de utilizar la variable.

```
fun main() {

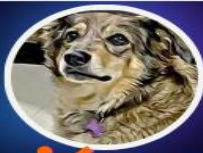
    println("Ejemplos de operadores UNARIOS")

    // ++ --
    var num1:Int=1

    println("num1 por defecto $num1")
    println("num1 con prefijo ${++num1}")
    println("num1 despues del incremento $num1")
}
```

Ejemplos de operadores UNARIOS  
num1 por defecto 1  
num1 con prefijo 2  
num1 despues del incremento 2





# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

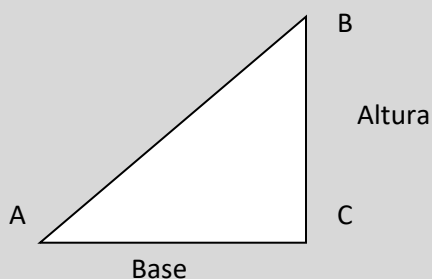
```
fun main() {  
  
    println("Ejemplos de operadores UNARIOS")  
  
    // ++ --  
    var num1:Int=1  
  
    println("num1 por defecto $num1")  
    println("num1 con postfijo ${num1++}")  
    println("num1 despues del incremento $num1")  
}
```

```
Ejemplos de operadores UNARIOS  
num1 por defecto 1  
num1 con postfijo 1  
num1 despues del incremento 2
```

**\*\*\* Ejercicios en clase ¡! \*\*\***

## Ejercicio 3

Hallar e imprimir el área de un triángulo



$$\text{Área} = (\text{Base} * \text{Altura}) / 2$$

Base= 5  
Altura = 7

## Ejercicio 4

Calcular e imprimir el salario quincenal de un empleado a partir de sus horas trabajadas y de su salario por hora.

Total= horas trabajadas \* Salario hora

Hrs. Trabajadas = 48  
Salario por hora = 14000

## Ejercicio 5

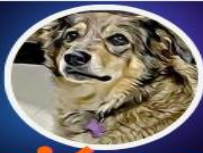
Dada una cantidad de horas (8 horas), calcular e imprimir su resultado, equivalente en minutos y segundos.

1 hora = 60 minutos  
1 minuto = 3600 segundos

## Ejercicio 6

Calcular e imprimir el promedio de notas del estudiante Juan Pérez

Matemáticas =9.5  
Biología = 9.0  
Física = 8.5



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA  
eylozano@sena.edu.co

Ética = 9.5  
Español = 8.7

// Comparta el código con su instructora mediante el repositorio anteriormente creado, nombre los ejercicios así: Ej 3-4-5-6\_nombre del aprendiz y lo demás como se indicó para el ejercicio 1 y 2.

## Operadores relacionales

Devuelven valores booleanos: True o False, posterior a una comparación, permitiendo definir la relación entre 2 o más expresiones.

```
*/  
fun main() {  
  
    // < - menor que          - 2 < 4  
    // > - mayor que          - 6 > 2  
    // <= - menos o igual que - 2<=3  
    // >= - mayor o igual que - 3>=2  
    // != - diferente de      - 2!=8  "juan" != "oscar"  
    // == - igual que         - 2 ==2  "oscar" == "oscar"  
  
    var num1 = 8  
    var num2 = 5  
    var res=num1 > num2  
  
    println("$num1 es mayor que $num2 = $res")  
  
    8 es mayor que 5 = true
```

## Operadores lógicos

Nos permite combinar expresiones y compararlas en términos lógicos.

```
fun main() {  
  
    // ! - Negación          !true (daria como res. falso)  
    // && - y (Conjunción)    (2<5) && (8>5) = True - Es true si ambos son True,  
    // si uno de los dos fuera false daria como resultado false  
    // || - o (Disyunción)   (2<5) || (5>8) = True - Es true si al menos uno  
    // es true  
  
    var num1 = 2  
    var num2 = 4  
    var res=(num1==num2) && (num1<5)  
  
    println("($num1==$num2) && ($num1<5)= $res")  
  
    (2==4) && (2<5)= false
```

\*\*\* Ejercicios en clase ¡! \*\*\*

**Ejercicio 7:** Plante al menos dos ejercicios con los operadores relacionales indicados arriba, escójalos a su consideración, determine e imprima.

**Ejercicio 8:** Plante al menos dos ejercicios con los operadores lógicos indicados arriba, escójalos a su consideración, determine e imprima.



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

eylozano@sena.edu.co

// Comparta el código con su instructora mediante el repositorio anteriormente creado, nombre los ejercicios así: Ej 7 y 8\_nombre del aprendiz y lo demás como se indicó para el ejercicio 1 y 2.

## Rango de instancias o alcance

Nos será de utilidad cuando se esté trabajando con arreglos o listas cuando queramos definir un rango o intervalo.

```
fun main() {  
  
    // a..b ..Define el intervalo entre los valores de a y b  
    // x in a..b Valida si x se encuentra dentro del intervalo  
    // x !in a..b Valida si x NO se encuentra dentro del intervalo  
  
    var res=2 in 1..5  
  
    println("2 está en el intervalo de 1..5? =$res")  
    res=22 in 1..5  
    println("22 está en el intervalo de 1..5? =$res")  
    res=15 !in 1..5  
    println("15 NO está en el intervalo de 1..5? =$res")  
}
```

```
2 está en el intervalo de 1..5? =true  
22 está en el intervalo de 1..5? =false  
15 NO está en el intervalo de 1..5? =true
```

## \*\*\* Ejercicios en clase ¡! \*\*\*

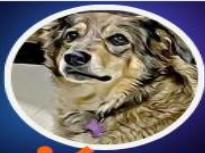
**Ejercicio 9:** Haciendo uso de ***x in a..b***, determinar e imprimir

- 3 está en el intervalo de 1 a 10?
- 10 está en el intervalo de 15 a 30?
- 7 está en el intervalo de 5 a 10?
- 2 está en el intervalo de 1 a 3?
- 5 está en el intervalo de 6 a 10?
- 'c' está en el intervalo de 'a'..'f'
- 'h' está en el intervalo de 'a'..'f'

**Ejercicio 10:** Haciendo uso de ***x !in a..b***, determinar e imprimir

- 5 está en el intervalo de 1 a 10?
- 16 está en el intervalo de 15 a 30?
- 2 está en el intervalo de 5 a 10?
- 3 está en el intervalo de 1 a 3?
- 7 está en el intervalo de 6 a 10?
- 'g' está en el intervalo de 'a'..'f'
- 'e' está en el intervalo de 'a'..'f'

// Comparta el código con su instructora mediante el repositorio anteriormente creado, nombre los ejercicios así: Ej 9 y 10\_nombre del aprendiz y lo demás como se indicó para el ejercicio 1 y 2.



# Introducción a KOTLIN

ELIANA YINETH  
LOZANO TRIANA

[eylozano@sena.edu.co](mailto:eylozano@sena.edu.co)