

## Tipos de Datos en Kotlin

Kotlin es un lenguaje de programación moderno y estáticamente tipado que se ejecuta en la JVM (Java Virtual Machine). Uno de sus aspectos fundamentales es el manejo de tipos de datos, que permite definir la naturaleza de las variables y asegurar la seguridad en tiempo de compilación.

### Tipos de Datos Básicos (Primitivos)

1. **Números:**
  - **Int:** Enteros de 32 bits.
  - **Long:** Enteros de 64 bits.
  - **Short:** Enteros de 16 bits.
  - **Byte:** Enteros de 8 bits.
  - **Float:** Números de punto flotante de 32 bits.
  - **Double:** Números de punto flotante de 64 bits.
2. **Boolean:**
  - Representa valores de verdad: `true` o `false`.
3. **Char:**
  - Representa un único carácter. Se define entre comillas simples, como `'a'`.
4. **String:**
  - Representa una secuencia de caracteres, definido entre comillas dobles, como `"Hola, Kotlin"`.

### Tipos de Referencia

Kotlin ofrece varias estructuras de datos que son esenciales para el manejo de colecciones:

1. **List:** Una colección ordenada que permite elementos duplicados. Puede ser mutable (`MutableList`) o inmutable (`List`).
2. **Set:** Una colección que no permite elementos duplicados. También puede ser mutable (`MutableSet`) o inmutable (`Set`).
3. **Map:** Una colección de pares clave-valor. Al igual que las demás, puede ser mutable (`MutableMap`) o inmutable (`Map`).

### Tipos Nullable

Una de las características distintivas de Kotlin es su manejo de tipos nulos. Por defecto, las variables no pueden contener valores nulos, lo que ayuda a evitar el famoso "NullPointerException". Para permitir valores nulos, se usa el operador `?`, como en `String?`.

### Tipo Any

El tipo `Any` es el supertipo de todos los tipos en Kotlin. Cualquier objeto puede ser asignado a una variable de tipo `Any`, aunque para acceder a sus propiedades o métodos específicos, es necesario realizar un casting.

## **Ejemplos**

### **Números:**

val numeroEntero: Int = 42

val numeroLargo: Long = 100000L

val numeroCorto: Short = 1000

val numeroByte: Byte = 10

val numeroDecimal: Float = 3.14f

val numeroDouble: Double = 3.14159

### **Números:**

val esVerdadero: Boolean = true

val esFalso: Boolean = false

### **List:**

val listaInmutable: List<Int> = listOf(1, 2, 3, 4)

val listaMutable: MutableList<String> = mutableListOf("a", "b", "c")

### **Set**

val conjuntoInmutable: Set<Int> = setOf(1, 2, 3)

val conjuntoMutable: MutableSet<String> = mutableSetOf("x", "y", "z")

### **Map**

val mapaInmutable: Map<String, Int> = mapOf("uno" to 1, "dos" to 2)

val mapaMutable: MutableMap<String, String> = mutableMapOf("nombre" to "Juan", "apellido" to "Pérez")

### **Para cualquier valor**

val cualquierCosa: Any = "Puede ser cualquier tipo"

