



## **Tipos de Datos en Kotlin**

**Alumnos:**

**Canales Bernal Manuel Alejandro**

**Materia:**

**Wearables IDSM**

**Grado y Grupo:**

**9A**

**Docente:**

**Ray Brunett Parra Galaviz**

**Fecha:**

**25 de septiembre del 2024**

## **Introducción:**

En Kotlin, los tipos de datos se dividen en dos grandes categorías: tipos de datos primitivos y tipos de datos de referencia. Sin embargo, es importante notar que en Kotlin, incluso los tipos "primitivos" son tratados como objetos, lo que permite un enfoque más uniforme al trabajar con tipos de datos.

### **Tipos Primitivos:**

Byte, Short, Int, Long, Float, Double, Char, Boolean

### **Tipos de Referencia:**

String, Array, List, Set, Map, clases y objetos definidos por el usuario

## **Tipos de Datos Primitivos**

Aunque en Kotlin todos los tipos son objetos, estos tipos primitivos se manejan de manera especial en términos de eficiencia (internamente, pueden ser optimizados para trabajar como tipos primitivos de JVM cuando sea posible):

### **Números enteros:**

**Byte:** 8 bits

**Short:** 16 bits

**Int:** 32 bits

**Long:** 64 bits

### **Números decimales (punto flotante):**

**Float:** 32 bits (precisión simple)

**Double:** 64 bits (precisión doble)

### **Otros tipos primitivos:**

**Char:** carácter Unicode (16 bits)

**Boolean:** puede tener solo dos valores: true o false

## Tipos de Datos de Referencia

Estos tipos de datos son referencias a objetos en la memoria, y no contienen los valores directamente.

**Cadenas de texto (String):** Representa secuencias de caracteres. En Kotlin, las cadenas son inmutables.

**Ejemplo:** "Hola, Kotlin"

**Arreglos (Array):** Una estructura que almacena múltiples valores del mismo tipo.

**Ejemplo:** `val nums: Array<Int> = arrayOf(1, 2, 3)`

**Colecciones:** Kotlin tiene una rica API de colecciones que incluye:

**Listas (List):** Colección ordenada de elementos (puede ser mutable o inmutable).

**Ejemplo:** `val list: List<Int> = listOf(1, 2, 3)`

**Conjuntos (Set):** Colección de elementos únicos.

**Ejemplo:** `val set: Set<Int> = setOf(1, 2, 3)`

**Mapas (Map):** Colección de pares clave-valor.

**Ejemplo:** `val map: Map<String, Int> = mapOf("uno" to 1, "dos" to 2)`

**Clases y Objetos:** Todos los tipos de datos definidos por el usuario (como clases, interfaces, objetos) son tipos de referencia.

**Ejemplo:**

```
class Persona(val nombre: String, val edad: Int)
```

## **Bibliografía:**

Programación Fácil. (s.f.). Los tipos de datos en Kotlin. Recuperado de <https://programacionfacil.org>(Programación Fácil)(Programación Fácil)

Codea App. (s.f.). Tipos de datos en Kotlin y ejemplos de algoritmos de ordenamiento. Recuperado de <https://codea.app>(Laravel Web & Mobile)

Mil y Una Formas. (s.f.). Tipos de datos en Kotlin. Recuperado de <https://milyunaformas.com>(Mil y una formas)

Nube Colectiva. (s.f.). Tipos de datos en Kotlin - Parte 1. Recuperado de <https://blog.nubecolectiva.com>(Nube Colectiva)

Android Developers. (s.f.). Kotlin en Android Studio. Recuperado de <https://developer.android.com>