



**UTT**

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

**GOBIERNO DE BAJA CALIFORNIA**

**TEMA:**

Sentencias de Block

**PRESENTADO POR:**

Padilla Virgen Jorge Luis

**GRUPO:**

9B

**MATERIA:**

Desarrollo para Dispositivos Inteligentes

**PROFESOR:**

Ray Brunett Parra Galaviz

Tijuana, Baja California, 02 de octubre del 2024

En Kotlin, las sentencias de bloque son fundamentales para controlar el flujo del programa y organizar la lógica de manera eficiente. A continuación, se describen los elementos clave:

### 1. Sentencia if

La estructura **if** en Kotlin es más flexible que en otros lenguajes, ya que puede ser utilizada como expresión, permitiendo que devuelva un valor directamente. Esto permite simplificar el código y evitar el uso de operadores ternarios. Se pueden combinar múltiples condiciones mediante **if-else**, lo que permite una estructura de decisión más clara.

### 2. Sentencia when

**when** es la alternativa al **switch** tradicional de otros lenguajes, pero es mucho más potente y flexible. Se puede usar como expresión o sentencia y permite manejar múltiples casos de manera más simple. Además de comprobar valores exactos, admite condiciones más complejas, como rangos o tipos de datos.

### 3. Bucles: for, while y do-while

Kotlin ofrece varias formas de realizar iteraciones:

- **for** se utiliza para recorrer colecciones o rangos de manera sencilla, lo que facilita trabajar con listas, arrays o secuencias numéricas.
- **while** ejecuta un bloque de código mientras una condición se mantenga verdadera, siendo útil para repeticiones que dependen de una condición dinámica.
- **do-while** es similar a **while**, pero garantiza que el bloque de código se ejecute al menos una vez antes de comprobar la condición, lo cual es útil en escenarios donde la lógica debe ejecutarse siempre al menos una vez.

## **4. Funciones**

Las funciones en Kotlin permiten encapsular y reutilizar bloques de código. Pueden definirse con parámetros opcionales y valores predeterminados, lo que aumenta su flexibilidad. Kotlin admite también funciones de una sola línea, lo que reduce la cantidad de código necesario para operaciones simples. Además, las funciones de orden superior permiten pasar funciones como parámetros o devolverlas, lo que facilita el manejo de la lógica funcional en los programas.