



Sentencias de Block

Alumnos:

Canales Bernal Manuel Alejandro

Materia:

Wearables IDSM

Grado y Grupo:

9A

Docente:

Ray Brunett Parra Galaviz

Fecha:

2 de Octubre del 2024

Resumen: Sentencias de Bloque en Kotlin

Kotlin es un lenguaje de programación moderno que promueve la claridad y la concisión en el código. Entre sus características más destacadas se encuentran las sentencias de bloque, que permiten controlar el flujo de ejecución del programa. Estas se dividen principalmente en tres categorías: condicionales, bucles y funciones.

1. Condicionales (IF)

La sentencia 'if' en Kotlin permite ejecutar bloques de código basados en condiciones booleanas. Se puede utilizar de manera simple o en forma de expresión, donde 'if' devuelve un valor. La sintaxis básica es:

```
```kotlin
if (condicion) {
 // código a ejecutar si la condición es verdadera
} else {
 // código a ejecutar si la condición es falsa
}
```
```

Kotlin también permite el uso de 'when' como una alternativa más expresiva a las sentencias 'if-else', ideal para múltiples condiciones:

```
```kotlin
when (valor) {
 condicion1 -> { /* código */ }
 condicion2 -> { /* código */ }
 else -> { /* código por defecto */ }
}
```
```

2. Bucles (LOOPS)

Kotlin ofrece varias estructuras de bucles para iterar sobre colecciones o ejecutar un bloque de código repetidamente:

- ****for****: Itera sobre rangos o colecciones.

```
```kotlin
for (item in collection) {
 // código a ejecutar para cada elemento
}
```
```

- ****while****: Ejecuta un bloque de código mientras la condición sea verdadera.

```
```kotlin
while (condicion) {
 // código a ejecutar
}
```
```

- ****do while****: Similar a 'while', pero garantiza que el bloque se ejecute al menos una vez.

```
```kotlin
do {
 // código a ejecutar
} while (condicion)
```
```

3. Funciones

Las funciones en Kotlin son bloques de código reutilizables que se pueden definir utilizando la palabra clave 'fun'. Pueden aceptar parámetros y devolver un valor:

```
```kotlin
fun nombreFuncion(parametro: Tipo): TipoDeRetorno {
 // código
 return valor
}
```
```

Las funciones son esenciales para organizar el código y mejorar la legibilidad, permitiendo descomponer tareas complejas en partes más manejables.