

Práctica Guiada

Hola mundo Kotlin

Asignatura: Aplicaciones Móviles

Tiempo planificado en el sílabo: 2 horas

Puntos a la calificación del parcial: 1 punto

Estudiante: Mero Cepeda Brando Rafael

Curso: "B"

Lectura previa:

Kotlin: <https://kotlinlang.org>

Kotlin playground: <https://tinyurl.com/u437ttu5>

Requerimientos de software:

Android Studio: <https://developer.android.com/studio>

Kotlin: Está incluido en la instalación de Android Studio.

1. TEMA: Desarrollo de la primera App móvil

En los codelabs de este programa, compilarás una app de Dice Roller para Android. Cuando el usuario "lance el dado", se generará un resultado aleatorio. Ese resultado tendrá en cuenta la cantidad de caras que tiene el dado. Por ejemplo, para un dado de 6 caras, solo saldrán valores del 1 al 6.

2. CÓDIGO FUENTE BASE:

Sin código fuente base.

3. OBJETIVOS:

- Crear un proyecto móvil Android nativo con Kotlin

4. ACTIVIDADES POR DESARROLLAR:

- **Seguir los pasos de estas actividades:**
 1. <https://developer.android.com/codelabs/basic-android-kotlin-training-create-dice-roller-in-kotlin>
 2. <https://developer.android.com/codelabs/basic-android-kotlin-training-create-dice-roller-app-with-button>
 3. <https://developer.android.com/codelabs/basic-android-kotlin-training-conditional-dice-roll-behavior>
 4. <https://developer.android.com/codelabs/basic-android-kotlin-training-dice-roller-images>
 - Pegar a continuación capturas de pantalla de la(s) aplicación(es) resultado de los Codelab:
- Actividad 1**

Playground de Kotlin

Prueba Kotlin y practica lo que aprendiste hasta ahora. Escribe tu código en la siguiente ventana y haz clic en el botón para ejecutarlo.

```
fun main() {
    val myFirstDice = Dice(6)
    println("¡Tus dados de ${myFirstDice.numSides} caras sacaron ${myFirstDice.roll()}!")

    val mySecondDice = Dice(20)
    println("¡Tus dados de ${mySecondDice.numSides} caras sacaron ${mySecondDice.roll()}!")
}

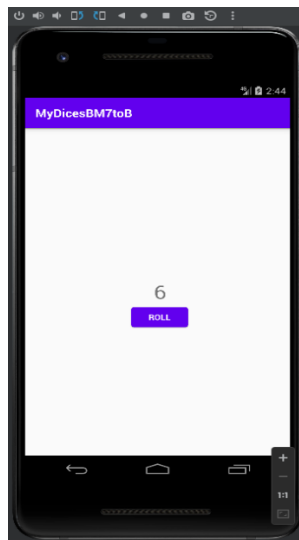
class Dice (val numSides: Int) {

    fun roll(): Int {
        return (1..numSides).random()
    }
}
```

¡Tus dados de 6 caras sacaron 5!
¡Tus dados de 20 caras sacaron 16!

Target platform: JVM Running on Kotlin v. 1.7.10

Actividad 2



```
8  /* Maria Cepeda Brando Rafael */
9
10 /* código estudiado, tomado de la página https://developer.android.com/codelabs
11 * para entender el manejo de androidStudio
12 */
13 class MainActivity : AppCompatActivity() {
14     override fun onCreate(savedInstanceState: Bundle?) {
15         super.onCreate(savedInstanceState)
16         setContentView(R.layout.activity_main)
17
18         val rollButton: Button = findViewById(R.id.button)
19         rollButton.setOnClickListener { rollDice() }
20     }
21     /**
22      * Tira los dados y actualiza la pantalla con el resultado.
23      */
24     private fun rollDice() {
25         // Crea un nuevo objeto Dice con 6 lados y hazlo rodar
26         val dice = Dice( numSides: 6)
27         val diceRoll = dice.diceRoll()
28         // Actualizar la pantalla con la tirada de dados
29         val resultTextView: TextView = findViewById(R.id.textView)
30         resultTextView.text = diceRoll.toString()
31     }
32 }
33
34 //Clase Dado "Dice" en la que se le da la función de rodar el dado
35 class Dice(private val numSides: Int) {
36     /*
37      * método que permite crear un rango de números aleatorios que se encuentre en el rango de números
38      * de lados y retorne un número al azar.
39      */
40     fun diceRoll(): Int {
41         return (1 ≤ .. ≤ numSides).random()
42     }
43 }
```

Actividad 3

```

fun main() {
    val myFirstDice = Dice(6)
    val rollResult = myFirstDice.roll()
    val luckyNumber = 4

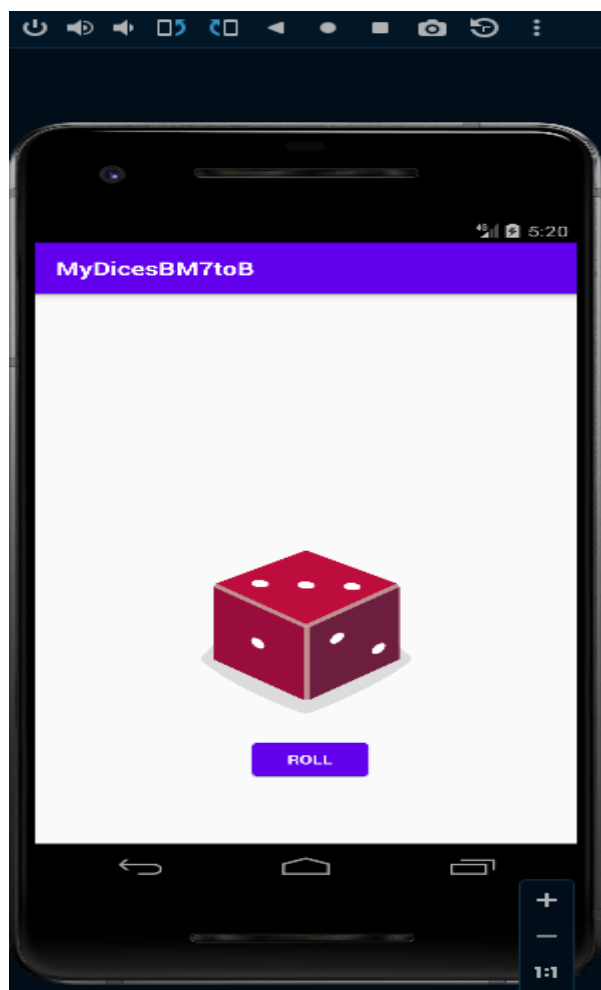
    when (rollResult) {
        luckyNumber -> println("You won! :3")
        1 -> println("So sorry! You rolled a 1. Try again! UnU")
        2 -> println("Sadly, you rolled a 2. Try again! UnU")
        3 -> println("Unfortunately, you rolled a 3. Try again! UnU")
        5 -> println("Don't cry! You rolled a 5. Try again! UnU")
        6 -> println("Apologies! You rolled a 6. Try again! UnU")
    }
}

class Dice (val numSides: Int) {
    fun roll(): Int {
        return (1..numSides).random()
    }
}

```

Unfortunately, you rolled a 3. Try again! UnU

Actividad 4



```

    val rollButton: Button = findViewById(R.id.button)
    val diceImage: ImageView = findViewById(R.id.imageView)
    diceImage.setImageResource(R.drawable.dice_5)
    rollButton.setOnClickListener { rollDice() }
}

/**
 * Tira los dados y actualiza la pantalla con el resultado.
 */
private fun rollDice() {
    // Crea un nuevo objeto Dice con 6 lados y hazla rodar
    val dice = Dice( numSides: 6)
    val diceRoll = dice.diceRoll()
    // Encuentra la ImageView en el diseño
    val diceImage: ImageView = findViewById(R.id.imageView)
    // Determine which drawable resource ID to use based on the dice roll
    val drawableResource = when (diceRoll) {
        1 → R.drawable.dice_1
        2 → R.drawable.dice_2
        3 → R.drawable.dice_3
        4 → R.drawable.dice_4
        5 → R.drawable.dice_5
        else → { R.drawable.dice_6 }
    }

    diceImage.setImageResource(drawableResource)
    diceImage.contentDescription = diceRoll.toString()
}

```

- Pegar a continuación la(s) URL(s) de la(s) aplicación(es).

<https://github.com/VzBrandonZ/PracticeGuideMovil>

Actividad adicional: <https://developer.android.com/codelabs/basic-android-kotlin-training-intro-debugging>