

Investigación sobre `readln()` y `random()` en Kotlin

Carlos Adrian Vizarreta Checya

Agosto 2025

Índice

1. Introducción	2
2. Herramienta utilizada: IntelliJ IDEA	2
3. Lectura de datos con <code>readln()</code>	2
4. Generación de valores aleatorios con <code>random()</code>	2
4.1. Aleatorio en un rango	2
4.2. Selección aleatoria en listas	3
4.3. Usando <code>Random</code> de Kotlin	3
5. Ejemplo práctico	3
6. Conclusiones	3

1. Introducción

En el presente informe se investigan las funciones `readln()` y `random()` en el lenguaje de programación Kotlin, utilizando el entorno de desarrollo IntelliJ IDEA. Estas funciones permiten la interacción con el usuario y la generación de valores aleatorios, lo cual resulta esencial en aplicaciones dinámicas y educativas.

2. Herramienta utilizada: IntelliJ IDEA

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) ampliamente utilizado para programar en Kotlin, ya que ofrece:

- Autocompletado de código.
- Ejecución rápida de programas.
- Integración con librerías estándar.

3. Lectura de datos con `readln()`

La función `readln()` permite leer datos desde la entrada estándar (teclado). Siempre devuelve un texto de tipo `String`, el cual puede convertirse a otros tipos de datos.

Ejemplo de uso en Kotlin:

```
fun main() {  
    println("Ingrese su nombre:")  
    val nombre = readln()  
    println("Hola, $nombre!")  
}
```

Si se requiere otro tipo de dato, se puede transformar:

```
val numero = readln().toInt()  
val decimal = readln().toDouble()
```

4. Generación de valores aleatorios con `random()`

Kotlin permite obtener valores aleatorios de distintas formas:

4.1. Aleatorio en un rango

```
fun main() {  
    val numero = (1..10).random()  
    println("Número aleatorio: $numero")  
}
```

4.2. Selección aleatoria en listas

```
val colores = listOf("Rojo", "Verde", "Azul")
val colorAleatorio = colores.random()
println("Color aleatorio: $colorAleatorio")
```

4.3. Usando Random de Kotlin

```
import kotlin.random.Random

fun main() {
    val entero = Random.nextInt(100)          // 0 a 99
    val decimal = Random.nextDouble(0.0, 1.0)
    println("Entero: $entero, Decimal: $decimal")
}
```

5. Ejemplo práctico

Un pequeño juego que combina lectura y valores aleatorios:

```
fun main() {
    println("Adivina un n mero entre 1 y 5:")
    val eleccion = readln().toInt()
    val secreto = (1..5).random()

    if (eleccion == secreto) {
        println(" Correcto ! El n mero era $secreto")
    } else {
        println("Fallaste. El n mero era $secreto")
    }
}
```

6. Conclusiones

- La función `readln()` es sencilla y útil para la interacción con el usuario.
- La función `random()` ofrece distintas maneras de generar datos impredecibles.
- IntelliJ IDEA facilita el trabajo con Kotlin gracias a su integración y herramientas de depuración.

Referencias

- Documentación oficial de Kotlin: <https://kotlinlang.org/docs/home.html>
- IntelliJ IDEA: <https://www.jetbrains.com/idea/>