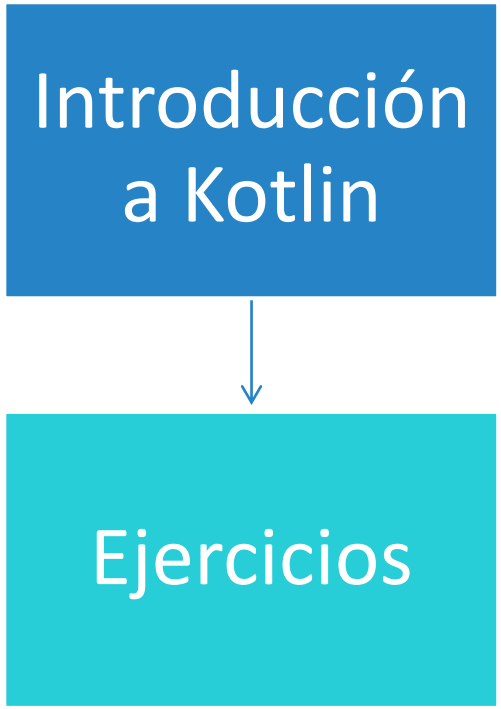
2º DAM

Programación Multimedia y Dispositivos Móviles

Programación Android

José A. Lara

Contenido



Introducción a Kotlin

Programación Android

Variables mutables e inmutables

Para utilizar variables mutables en Kotlin se utiliza la palabra reservada ***var***

```
var a:Int = 2
var b:Int = 3
var c:Int = a+b
```

```
var i=3
var j=4
var k=i+j
```

Para utilizar variables inmutables (constantes) en Kotlin se utiliza la palabra reservada ***val***

```
val a:Int = 2
val b:Int = 3
val c:Int = a+b
```

```
val e=3.1
val f=4.3
val g=e+f
```

android

Programación Android

String en Kotlin

Los objetos de tipo String tienen una serie de propiedades y métodos asociados:

- ***length:*** nos informa de la longitud de la cadena
- ***isEmpty()***: si la cadena está vacía o no
- ***contains(cadena2)***: si la cadena original contiene la cadena2
- ***substring(n1,n2)***: la subcadean desde la posición n1 a la posición n2
- ***replace(cad1,cad2)***: reemplaza la cad1 por la cad2

```
var nombre:String = "Pepe Lopez"
println(nombre.length)
var nombreVacio:Boolean = nombre.isEmpty()
println(nombreVacio)
var nombreContiene:Boolean = nombre.contains( other: "Lopez")
println(nombreContiene)
var subcadena:String = nombre.substring(1,6)
println(subcadena)
var reemplazar:String = nombre.replace( oldValue: "op", newValue: "ain")
println(reemplazar)
```

```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
10
false
true
epe L
Pepe Lainez

Process finished with exit code 0
```

android

Programación Android

String en Kotlin

```
var edad:Int = 20
var altura:Double = 1.80
var peso:Float = 80.5f
var esMujer:Boolean = true
var valorlong:Long = 123456789876543
var nombre:String = "Daniela Perez"
var descripcion:String = "Hola soy $nombre, tengo $edad años y mi altura es $altura y mi peso es $peso"
println(descripcion)
```

```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Hola soy Daniela Perez, tengo 20 años y mi altura es 1.8 y mi peso es 80.5

Process finished with exit code 0
```

android

Programación Android

String en Kotlin

```
var edad:Int = 20
var altura:Double = 1.80
var peso:Float = 80.5f
var esMujer:Boolean = true
var valorlong:Long = 123456789876543
var nombre:String = "Daniela Perez"
var descripcion:String = "Hola soy $nombre, tengo $edad años y mi altura es $altura y mi peso es $peso"
println(descripcion)
var cadenaFormato:String = String.format("Hola soy %s, tengo %d años y mi altura es %.2f y mi peso es %.2f",nombre,edad,altura,peso)
println(cadenaFormato)
```

```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
Hola soy Daniela Perez, tengo 20 años y mi altura es 1.8 y mi peso es 80.5
Hola soy Daniela Perez, tengo 20 años y mi altura es 1,80 y mi peso es 80,50

Process finished with exit code 0
```

android

Programación Android

String en Kotlin

Formato	Tipo de dato
%b	Boolean
%c	Char
%d	Integer
%e	Float en notación científica
%f	Float y Double (Agrega %**.n**f para forzar n decimales)
%o	Formato Octal
%s	Strings
%x	Formato Hexadecimal

Programación Android

String en Kotlin

Otros métodos:

- compareTo(cadena)
- equals(cadena)
- contains(cadena)

```
var nombre = "Pepito Grillo"
var nombre2 = "Juanita Campanilla"
println(nombre.length)
println(nombre.compareTo(nombre2))
println("pepito grillo".equals(nombre))
println(nombre.contains( other: "Grillo"))
```

```
"C:\Program Files\Android\Android St
13
6
false
true

Process finished with exit code 0
```

android

Programación Android

String en Kotlin

Otros métodos:

- toLowerCase(), toUpperCase()
- subSequence(inicio,fin)
- indexOf(cadena)
- lastIndexOf(cadena)
- isEmpty()
- isBlank()

```
var nombre = "Pepito Grillo"
var nombre2 = "Juanita Campanilla"
println(nombre.toLowerCase())
println(nombre2.toUpperCase())
println(nombre.subSequence(3,8))
println(nombre2.indexOf( string: "t"))
println(nombre.lastIndexOf( string: "i"))
val vacia=" "
println(vacia.isEmpty())
println(vacia.isBlank())
```

```
"C:\Program Files\Android\Android S
pepito grillo
JUANITA CAMPANILLA
ito G
5
9
false
true

Process finished with exit code 0
```

android

Programación Android

String en Kotlin

Otros métodos:

- replace(antiguo, nuevo)
- replaceBefore(delimitador, reemplazo)
- reversed()
- slice(índices)
- split(delimitador)
- startsWith(cadena)
- substring(desde,hasta)
- trim()

```
var nombre = "Pepito Grillo"
var nombre2 = "Juanita Campanilla"
println(nombre.replace( oldValue: "i", newValue: "e"))
println(nombre2.replaceBefore( delimiter: " ", replacement: "Anita"))
println(nombre.reversed())
println(nombre2.slice( indices: 5..nombre2.length-1))
println(nombre2.split( ...delimiters: " "))
println(nombre2.startsWith( prefix: "G"))
println(nombre2.substring(3,7))
println("    Hola     ".trim())
```

```
"C:\Program Files\Android\Android
Pepeto Grello
Anita Campanilla
ollirG otipeP
ta Campanilla
[Juanita, Campanilla]
false
nita
Hola

Process finished with exit code 0
```

android

Programación Android

Constantes

Para definir una constante utilizamos las palabras ***const val,*** o bien utilizar una clase con un ***companion object***

```
const val direccion="Cesur"
fun main(){
    println(direccion)
}
```

```
"C:\Program Files\Android\Android St
Cesur

Process finished with exit code 0
```

```
fun main(){
    println(Constantes.direccion)
}

class Constantes{
    companion object{
        const val direccion="Cesur"
    }
}
```

```
"C:\Program Files\Android\Android St
Cesur

Process finished with exit code 0
```

android

Programación Android

Enum classes

Para establecer un tipo enumerado, utilizaremos ***enum class***

```
fun main(){
    val dia=DIAS.VIERNES
    println(dia)
}

enum class DIAS{
    LUNES, MARTES, MIERCOLES, JUEVES, VIERNES, SABADO, DOMINGO
}
```

```
"C:\Program Files\Android\Android S
VIERNES

Process finished with exit code 0
```

```
fun main(){
    val dia=DIAS.VIERNES.numero
    println(dia)
}

enum class DIAS (val numero:Int){
    LUNES( numero: 0), MARTES( numero: 1), MIERCOLES( numero: 2),
    JUEVES( numero: 3), VIERNES( numero: 4), SABADO( numero: 5), DOMINGO( numero: 6)
}
```

```
"C:\Program Files\Android\Android S
4

Process finished with exit code 0
```

android

Programación Android

Funciones

Para definir una función utilizamos la palabra ***fun***, después se definen los parámetros y finalmente el tipo de datos que devuelve (si devuelve algo)

```
fun main(){
    println(esPar( numero: 8))
}

fun esPar(numero:Int):Boolean{
    return numero%2==0
}
```

```
"C:\Program Files\Android\Android St
true

Process finished with exit code 0
```

android

Programación Android

Funciones

También se puede definir así:

```
fun main(){
    val a=5
    val b=3
    println("$a - $b es igual a ${sub(a,b)}")
}

fun sub(a:Int, b:Int) = a-b
```

```
"/Applications/Android Studio.app/C
5 - 3 es igual a 2

Process finished with exit code 0
```

android

Programación Android

Funciones

infix

```
fun main(){
    val a=-5
    println(a.enableAbs( enable: true))
    println(a enableAbs(true) )
}

infix fun Int.enableAbs(enable: Boolean) = if (enable) abs(this) else this
```

```
"/Applications/Android Studio.app/
5
5

Process finished with exit code 0
```

android

Programación Android

Parámetros en array

```
fun main(){
    showPersons( ...personas: "Pepe","Juan")
    showPersons( ...personas: "Luis","Paco","Alberto")
}

fun showPersons(vararg personas:String){
    for(num in 0..personas.size-1) {
        println(personas[num])
    }
}
```

```
fun showPersons(vararg personas:String){
    (0..personas.size-1).forEach { num ->
        println(personas[num])
    }
}
```

```
fun showPersons(vararg personas:String){
    for (persona in personas){
        println(persona)
    }
}
```

```
fun showPersons(vararg personas:String){
    (personas.indices).forEach { num ->
        println(personas[num])
    }
}
```

```
"/Applications/Android Studio.app/
Pepe
Juan
Luis
Paco
Alberto

Process finished with exit code 0
```

android

Programación Android

Control de flujo - if

```
val a=1
val b=2
var max:Int
if (a>b){
    max=a
} else {
    max=b
}
println(max)
```

```
val a=1
val b=2
var max:Int
max=if (a>b){
    a
} else {
    b
}
println(max)
```

```
val a=1
val b=2
var max:Int
max=if (a>b) a else b
println(max)
```

```
"/Applications/Android Studio.app/C
2

Process finished with exit code 0
```

android

Programación Android

Listas mutables e inmutables

```
//var listaMutable = mutableListOf<String>("Jose","Maria","Pedro","Eva")
var listaMutable = mutableListOf("Jose","Maria","Pedro","Eva")
listaMutable.add("Juan")
listaMutable.removeAt( index: 2)
//var listaInmutable = listOf<String>("Coche","Moto","Bici")
var listaInmutable = listOf("Coche","Moto","Bici")
var tamanho:Int = listaInmutable.size
println("Tamaño de la lista inmutable: $tamanho")
tamanho=listaMutable.size
println("Tamaño de la lista mutable: $tamanho")
```

```
"C:\Program Files\Android\Android St
Tamaño de la lista inmutable: 3
Tamaño de la lista mutable: 4

Process finished with exit code 0
```

android

Programación Android

Listas inmutables

```
fun main(){
    val listaFruta= listOf("Manzana","Pera","Uva","Fresa")
    println(listaFruta.get((0..listaFruta.size-1).random()))
    println("Index de Uva es ${listaFruta.indexOf("Uva")}")
}
```

```
"/Applications/Android Studio.app/(
Fresa
Index de Uva es 2

Process finished with exit code 0
```

android

Programación Android

Listas mutables

```
fun main(){
    val listaMutable = mutableListOf("Juan","Jose","Maria","Rosa")
    println(listaMutable)
    listaMutable.add("Irene")
    println("Después de add: $listaMutable")
    listaMutable.removeAt( index: 2)
    println("Después de remove: $listaMutable")
    listaMutable.set(2,"Inés")
    println("Después de set: $listaMutable")
}
```

```
"C:\Program Files\Android\Android Studio\jre\bin\j
[Juan, Jose, Maria, Rosa]
Después de add: [Juan, Jose, Maria, Rosa, Irene]
Después de remove: [Juan, Jose, Rosa, Irene]
Después de set: [Juan, Jose, Inés, Irene]

Process finished with exit code 0
```

android

Programación Android

Algunos métodos más

```
val lista=listOf("Jose","Pepe","Maria","Irene")
println("lista: ${lista}")
println("sort: ${lista.sorted()}")
println("reverse: ${lista.reversed()}")
println("sorted + reverse: ${lista.sorted().reversed()}")
println("indexOf Pepe: ${lista.indexOf("Pepe")}")
```

```
"C:\Program Files\Android\Android Studio\jre
lista: [Jose, Pepe, Maria, Irene]
sort: [Irene, Jose, Maria, Pepe]
reverse: [Irene, Maria, Pepe, Jose]
sorted + reverse: [Pepe, Maria, Jose, Irene]
indexOf: 1

Process finished with exit code 0
```

android

Programación Android

Control de flujo - for

```
//var listaMutable = mutableListOf<String>("Jose","Maria","Pedro","Eva")
var listaMutable = mutableListOf("Jose","Maria","Pedro","Eva")
//var listaInmutable = listOf<String>("Coche","Moto","Bici")
var listaInmutable = listOf("Coche","Moto","Bici")
for (nombre in listaMutable){
    println(nombre)
}
```

```
"C:\Program Files\Android\Android S
Jose
Maria
Pedro
Eva

Process finished with exit code 0
```

android

Programación Android

Control de flujo - for

```
for (i in 1..10 step 2){
    println(i)
}
val lista=listOf("Jose","Pepe","Maria","Irene")
for (i in 0..lista.size-1){
    println("$i = ${lista.get(i)}")
}
for (nombre in lista){
    println(nombre)
}
```

```
"C:\Program Files\Android\Android
1
3
5
7
9
0 = Jose
1 = Pepe
2 = Maria
3 = Irene
Jose
Pepe
Maria
Irene

Process finished with exit code 0
```

android

Programación Android

Control de flujo - foreach

```
val lista=listOf("Jose","Pepe","Maria","Irene")
lista.forEach {
    name -> println(name)
}

(1..5).forEach {  it: Int
    println(it)
}
```

```
"C:\Program Files\Android\Android
Jose
Pepe
Maria
Irene
1
2
3
4
5

Process finished with exit code 0
```

android

Programación Android

Control de flujo - for

```
//var listaMutable = mutableListOf<String>("Jose","Maria","Pedro","Eva")
var listaMutable = mutableListOf("Jose","Maria","Pedro","Eva")
//var listaInmutable = listOf<String>("Coche","Moto","Bici")
var listaInmutable = listOf("Coche","Moto","Bici")
for ((index,value) in listaMutable.withIndex()){
    println("Nombre: $value con indice: $index")
}
```

```
"C:\Program Files\Android\Android S
Nombre: Jose con indice: 0
Nombre: Maria con indice: 1
Nombre: Pedro con indice: 2
Nombre: Eva con indice: 3

Process finished with exit code 0
```

Programación Android

Control de flujo - for

```
//var listaInmutable = listOf<String>("Coche","Moto","Bici")
var listaInmutable = listOf("Coche","Moto","Bici")

listaInmutable.forEach{ elementos ->
    println(elementos)
}

listaInmutable.forEachIndexed { index, elemento ->
    println("Elemento: $elemento con índice: $index")
}
```

```
"C:\Program Files\Android\Android S
Coche
Moto
Bici
Elemento: Coche con índice: 0
Elemento: Moto con índice: 1
Elemento: Bici con índice: 2

Process finished with exit code 0
```

android

Programación Android

Control de flujo - when

```
val x=2

when (x){
    1 -> {println("Uno")}
    2 -> {println("Dos")}
    3,4 -> {println("Mayor que dos")}
    else -> {println("Mayor que cuatro o menor que 1")}
}
```

```
"C:\Program Files\Android\Android S
Dos

Process finished with exit code 0
```

android

Programación Android

Control de flujo – while – do while

```
var i=1
while(i<=5){
    println("Valor: $i")
    i++
}
```

```
"/Applications/A
Valor: 1
Valor: 2
Valor: 3
Valor: 4
Valor: 5

Process finished
```

```
var sum=0
var entrada:String
do{
    print("Introduce un número:")
    entrada = readLine()!! //el valor NO es null
    sum+=entrada.toInt()
}while(entrada!="0")
println("Suma= $sum")
```

```
"/Applications/Android
Introduce un número:5
Introduce un número:6
Introduce un número:9
Introduce un número:8
Introduce un número:0
Suma= 28

Process finished with e
```

Programación Android

Control de flujo – return - break

```
(1..5).forEach { it: Int
    if (it==3){
        return@forEach
    }
    println(it)
}

for (i in 1..5){
    if (i==3){
        break
    }
    println(i)
}
```

```
"C:\Program Files\Android\Android
1
2
4
5
1
2

Process finished with exit code 0
```

Programación Android

Extension functions

```
fun main(){
    val nombre="Hola Kotlin"
    println(nombre.borrarPrimerYUltimoCaracter())
}

fun String.borrarPrimerYUltimoCaracter():String=this.substring(1,this.length-1)
```

```
"/Applications/Android Studio.app/
ola Kotli

Process finished with exit code 0
```

android

Programación Android

Nulabilidad

El símbolo ? permite que una variable tenga un valor null
El símbolo !! indica que el valor de esa variable NO es null

```
val nombre:String?=null
println(nombre!!.length)
```

```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/
Exception in thread "main" java.lang.NullPointerException
    at com.example.pruebakotlin.PruebaKt.main(Prueba.kt:5)
    at com.example.pruebakotlin.PruebaKt.main(Prueba.kt)

Process finished with exit code 1
```

```
val nombre:String?=null
println(nombre?.length)
```

```
"/Applications/Android Studio.app/C
null

Process finished with exit code 0
```

android

Programación Android

Nulabilidad

El operador elvis ?: reemplaza el valor de una variable si su valor es null

```
var elvis = nullStrGlobal ?: "Kotlin"
println("Yo programo en $elvis")
```

```
"C:\Program Files\Android\Android S
Yo programo en Kotlin

Process finished with exit code 0
```

```
var elvis = nullStrGlobal ?: "Kotlin"
elvis = "Java"
println("Yo programo en $elvis")
```

```
"C:\Program Files\Android\Android
Yo programo en Java

Process finished with exit code 0
```

android

Programación Android

Nulabilidad

El operador elvis ?: reemplaza el valor de una variable si su valor es null

```
val nombre:String?=null
val c=nombre?.length ?: "Leo".length
println(c)
```

```
"/Applications/Android Studio.app/
3

Process finished with exit code 0
```

```
val nombre:String?=null
nombre?.let{ it: String
    print(it)
}
```

```
"/Applications/Android Studio.app/C

Process finished with exit code 0
```

android

Programación Android

readLine

```
println("Introduce un número:")
val num = readLine()
val numero = num!!.toInt()
println("Número = $numero")
```

```
"C:\Program Files\Android\Android S
Introduce un número:
78
Número = 78

Process finished with exit code 0
```

Programación Android

Operadores matemáticos – Smart Cast

Los operadores clásicos son +, -, *, /, %

```
"C:\Program Files\Android\Android
a + b = 5
a - b = -1
a * b = 6
a / b = 0
a % b = 2
Es un número
6

Process finished with exit code 0
```

```
var a:Int = 2
var b:Int = 3
println("a + b = ${a+b}")
println("a - b = ${a-b}")
println("a * b = ${a*b}")
println("a / b = ${a/b}")
println("a % b = ${a%b}")

var obj:Any = 3
if (obj is Int){
    println("Es un número")
    println(obj.toString().toInt()*2)
} else {
    println("No es un número")
}
```

android

Programación Android

try – catch - finally

```
var ob:Any="Kotlin"
var obj:Any = 3
if (obj is Int){
    println("Es un número")
    println(obj.toString().toInt()*2)
} else {
    println("No es un número")
}

try{
    println(ob.toString().toInt() * 8)
    println("try")
} catch (e: Exception){
    println(e)
    println("Error en catch")
} finally {
    println("Try finalizado")
}
```

```
6
java.lang.NumberFormatException: For input string: "Kotlin"
Error en catch
Try finalizado

Process finished with exit code 0
```

android

Programación Android

Safe y unsafe cast

```
var ob:Any="Kotlin"
var obj:Any = 3
if (obj is Int){...} else {...}

try{...} catch (e: Exception){
    println(e)
    println("Error en catch")
} finally {...}

ob=true
val unsafeStr: String = ob as String
println(unsafeStr)
val safeStr= ob as? String
println(safeStr)
```

```
6
java.lang.NumberFormatException: For input string: "Kotlin"
Error en catch
Try finalizado
Exception in thread "main" java.lang.ClassCastException Create breakpoint : class java.lang.Boolean
    at com.jlara.prueba01.PruebaKt.main(prueba.kt:33)
    at com.jlara.prueba01.PruebaKt.main(prueba.kt)

Process finished with exit code 1
```

android

Programación Android

Safe y unsafe cast

```
println("Error en catch")
} finally {...}

ob=true
/*val unsafeStr: String = ob as String
println(unsafeStr)*/
val safeStr= ob as? String
println(safeStr)
```

```
6
java.lang.NumberFormatException: For input string: "Kotlin"
Error en catch
Try finalizado
null

Process finished with exit code 0
```

android

Programación Android

throw

```
fun main(){
    val job="Diseñador"
    chequearTipo(job)
}

private fun chequearTipo(value: Any) {
    if (value is String){
        println("String válido")
    } else {
        throw TypeCastException("No es un String")
    }
}
```

```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
String válido

Process finished with exit code 0
```

android

Programación Android

throw

```
fun main(){
    val job="Diseñador"
    chequearTipo(job)
    val b=10
    chequearTipo(b)
}

private fun chequearTipo(value: Any) {
    if (value is String){
        println("String válido")
    } else {
        throw TypeCastException("No es un String")
    }
}
```

```
"C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
String válido
Exception in thread "main" kotlin.TypeCastException: No es un String
    at com.jlara.prueba01.Prueba2Kt.chequearTipo(prueba2.kt:14)
    at com.jlara.prueba01.Prueba2Kt.main(prueba2.kt:7)
    at com.jlara.prueba01.Prueba2Kt.main(prueba2.kt)

Process finished with exit code 1
```

android

Programación Android

Lazy - Lateinit

Lateinit permite declarar una variable (solo var) que no va a tener ningún valor inicialmente.

```
private lateinit var lateinitGlobal:String

fun main(){
    val job="Diseñador"
    chequearTipo(job)
    val b=10
    chequearTipo(b)
    if (asignarValor()){
        println(lateinitGlobal)
    }
}

private fun asignarValor(): Boolean{
    lateinitGlobal="glateinitKotlin"
    return lateinitGlobal.isNotEmpty()
}
```

```
Tarea finalizada exitosamente
glateinitKotlin

Process finished with exit code 0
```

android

Programación Android

Lazy - Lateinit

Lazy (solo con val) asigna un valor a la variable cuando es invocada, y ya no se le vuelve a asignar ni se puede cambiar su valor.

```
private lateinit var lateinitGlobal:String
private val lazyGlobal: Any by lazy { "gLazyKotlin"}
fun main(){
    val job="Diseñador"
    chequearTipo(job)
    val b=10
    chequearTipo(b)
    if (asignarValor()){
        println(lateinitGlobal)
    }
    println(lazyGlobal)
}

private fun asignarValor(): Boolean{
    lateinitGlobal="glateinitKotlin"
    return lateinitGlobal.isNotEmpty()
}
```

```
Tarea finalizada exitosamente
glateinitKotlin
gLazyKotlin

Process finished with exit code 0
```

android

Programación Android

Mapas mutables

```
fun main(){
    val user1:User=User( name: "Pepe", lastName: "Lopez")
    val user2:User=User( name: "Paco", lastName: "Perez")
    val user3:User=User( name: "Luis", lastName: "Sanchez")
    val usersMap= mutableMapOf<Int,User>()
    usersMap.put(1,user1)
    usersMap.put(2,user2)
    println(usersMap)
    usersMap.put(3,user3)
    usersMap.remove( key: 2)
    println(usersMap)
    println(usersMap.isEmpty())
    println(usersMap.containsKey(1))
    println(usersMap.keys)
    println(usersMap.values)
}
```

```
"/Applications/Android Studio.app/Contents/jre/jdk/Contents/Home/bin/java
{1=User(name=Pepe, lastName=Lopez), 2=User(name=Paco, lastName=Perez)}
{1=User(name=Pepe, lastName=Lopez), 3=User(name=Luis, lastName=Sanchez)}
false
true
[1, 3]
[User(name=Pepe, lastName=Lopez), User(name=Luis, lastName=Sanchez)]

Process finished with exit code 0
```

android

Programación Android

Array de nulls

```
fun main(){
    val arrayNulos = arrayOfNulls<String>( size: 3)
    arrayNulos[1]="Jose"
    println(arrayNulos[0])
    println(arrayNulos[1])
}
```

```
"C:\Program Files\Android\
null
Jose

Process finished with exit
```

android

Ejercicios básicos de programación en Kotlin

Programación Android

Ejercicios

1. Crea un programa que te pida una cantidad en miligramos para una poción mágica, el valor debe ser de tipo entero, si el valor es mayor a 100 imprime “¡Felicidades, es una buena poción!”, de lo contrario imprime “La poción es mediocre, sangre sucia inmunda”.

2. Haz un ciclo **for** y **while** que obtenga la sumatoria de los números hasta n (ejemplo: para 5 debes obtener 15 (1+2+3+4+5), para 3 debes obtener 6). Imprime el resultado así: “La suma es 15” usando formatos de String.

android

Programación Android

Ejercicios

3. Escribe un programa que te diga si un Uber puede iniciar su recorrido. Para esto se necesitan dos cosas, que el conductor esté cerca y que esté disponible, el programa te pedirá dos valores, la distancia del conductor en kilómetros y su disponibilidad, donde false = no disponible y true = disponible, según los datos que insertes imprime lo siguiente:
   - Si la distancia es menor o igual a 0.5 km y el conductor está disponible, imprime “Listo para iniciar recorrido”
   - Si la distancia es menor o igual a 0.5 km y el conductor NO está disponible, imprime, “Conductor cercano, pero no disponible”
   - Si la distancia es mayor a 0.5 km y el conductor está disponible, imprime, “Conductor disponible pero muy lejos, aplicarán tarifas más altas”
   - Si la distancia es mayor a 0.5 km y el conductor NO está disponible, imprime, “No hay conductores disponibles”

RETO OPCIONAL: Si no se cumplen las condiciones de “Listo para iniciar recorrido”, después de imprimir el mensaje, vuelve a pedir los datos de distancia y disponibilidad

android

Programación Android

Ejercicios

4. Escribe un programa que imprima “¿Cómo es el clima de hoy?”, dependiendo del número que insertes te imprima los siguientes valores
   - 1 = “Soleado”
   - 2 = “Nublado”
   - 3 = “Lluvioso”
   - 4 = “Tormentoso”
   - 5 = “Nevado”

   Si insertas cualquier otro valor debe imprimir “Pregúntale a Google”
   Recomendación: Usa When.

android

Programación Android

Ejercicios

5. Para un Array de String de títulos de películas, imprime el título de la película con el título más largo. Por ejemplo si tenemos un arreglo con los valores:
   {“Jumanji”, “Toy Story”, “Pulp Fiction”, “Batman: El caballero de la noche”, “Kill Bill”}
   Debería imprimir “Batman: El caballero de la noche”.
   RETO OPCIONAL: Imprime el índice del título más largo, para el ejemplo anterior, debería imprimir el 3, porque es el índice de “Batman: El caballero de la noche”

6. Escribe un programa que para un número N, imprima los números pares desde 1 hasta N, por ejemplo si N = 6, debe imprimir “2, 4, 6”. Hazlo con un for o un while. Si el número es menor o igual a 0, debes imprimir “Inserta un número positivo”

android