# Ejemplo de uso de .also en Kotlin

[crayon-6a9d9149849a4363944749/]

Explicación del código:

fun main() {

Esta línea marca el inicio de la función main(), que es el punto de entrada de cualquier programa en Kotlin.

val nombres = listOf("Ana", "Juan", "María", "Carlos", "Elena", "Pedro", "Sofía")

Aquí se crea una lista llamada nombres que contiene una serie de nombres como elementos. listOf() es una función de Kotlin que crea una lista inmutable.

val cantidadNombresLargos = nombres

Se declara una variable llamada cantidadNombresLargos para almacenar el resultado de las operaciones posteriores.

.map { it.length }

La función map se utiliza para transformar cada elemento de la lista. En este caso, se está aplicando una función lambda que toma cada nombre (it) y calcula su longitud (length). Esto produce una nueva lista con las longitudes de los nombres.

.also { println("Longitudes de los nombres: $it") }

.also es una función de orden superior que permite realizar una acción (en este caso, imprimir las longitudes de los nombres) sin modificar el objeto al que se aplica. La acción se define mediante una función lambda que toma el objeto anterior a .also (en este caso, la lista de longitudes) como argumento y lo imprime.

.count { it > 5 }

La función count se utiliza para contar los elementos de la lista que cumplan con una condición dada. En este caso, se cuenta cuántas longitudes de nombres son mayores que 5 (es decir, cuántos nombres tienen más de 5 letras).

println("Total de nombres largos: $cantidadNombresLargos") }

Finalmente, se imprime el resultado, que es la cantidad de nombres que tienen más de 5 letras.

En resumen, este código crea una lista de nombres, calcula las longitudes de los nombres, imprime esas longitudes y cuenta cuántos nombres tienen más de 5 letras, todo ello utilizando la función .also para imprimir las longitudes intermedias sin afectar al flujo de datos principal.