

Taller Intermedio de Kotlin

Taller Intermedio de Kotlin – Práctica y Lógica.....¡Error! Marcador no definido.

1. Conversor de palabras clave.....	1
2. Clasificador de edades con when.....	1
3. Generador de resumen de texto.....	2
4. Sistema de registro con List mutable.....	2
5. Control de inventario con Map mutable.....	2
6. Operaciones con Set mutable e inmutable	3
7. Analizador de datos con bucles anidados	3
8. Generador de reportes con funciones de orden superior	3
9. Simulador de login.....	4
10. Aplicación de lógica combinada	4

1. Conversor de palabras clave

Objetivo: Practicar variables, entrada del usuario, y métodos de strings.

Instrucción:

- Solicita una frase al usuario, y crea un algoritmo que:
 - Cuente cuántas palabras contiene.
 - Diga cuántas comienzan con vocal.
- Convierta cada palabra a minúscula, la recorte a los primeros 4 caracteres, y las almacene en una lista.

Conceptos: split(), startsWith(), toLowerCase(), substring(), List, variables.

2. Clasificador de edades con when

Objetivo: Practicar estructuras condicionales when, constantes y funciones.

Instrucción:

- Define constantes para los rangos de edad: niño (0-12), adolescente (13-17), adulto (18-64), mayor (65+).
- Crea una función clasificarEdad(edad: Int) que retorne una cadena con la categoría.
- Prueba con diferentes edades usando when.

Conceptos: const val, when, funciones, control de flujo.

3. Generador de resumen de texto

Objetivo: Uso intensivo de métodos de strings.

Instrucción:

- Dado un párrafo de texto:
- Elimina los signos de puntuación.
- Reemplaza palabras mayores a 6 letras con "[LARGA]".
- Cuenta la cantidad de veces que aparece una palabra clave que se pasa por parámetro.

Conceptos: replace(), split(), filter, count, funciones con parámetros.

4. Sistema de registro con List mutable

Objetivo: Practicar listas mutables, funciones y bucles.

Instrucción:

- Simula un sistema donde se registren nombres en una lista.
- Debe tener funciones para:
- Agregar nombres (sin duplicados).
- Mostrar todos los nombres ordenados alfabéticamente.
- Buscar si un nombre existe (ignorar mayúsculas).

Conceptos: mutableListOf(), add(), contains(), sort(), bucles.

5. Control de inventario con Map mutable

Objetivo: Aplicar lógica con diccionarios (mapas).

Instrucción:

- Crea un inventario como MutableMap<String, Int>, donde:
- Las claves son nombres de productos.
- Los valores son cantidades.
- Debe permitir:
- Agregar productos (nuevo o actualizar cantidad).
- Remover productos agotados.
- Mostrar el producto con mayor y menor cantidad.

Conceptos: MutableMap, put(), remove(), entries, maxBy, minBy.

6. Operaciones con Set mutable e immutable

Objetivo: Comprender la diferencia entre mutabilidad en conjuntos.

Instrucción:

- Define un Set immutable con frutas favoritas.
- Define un MutableSet para frutas disponibles.
- Realiza operaciones: unión, intersección y diferencia.
- Agrega una nueva fruta al MutableSet y verifica si está en el Set immutable.

Conceptos: setOf(), mutableSetOf(), operadores de conjunto (intersect, union, etc.)

7. Analizador de datos con bucles anidados

Objetivo: Uso avanzado de bucles con estructuras.

Instrucción:

- Crea una lista de oraciones.
- Para cada oración:
- Imprime cuántas palabras tiene.
- Calcula cuántas letras tiene la palabra más larga.
- Si alguna palabra es palíndromo (se lee igual al revés), márcala.

Conceptos: for, forEach, lógica, reversed(), condiciones.

8. Generador de reportes con funciones de orden superior

Objetivo: Introducción a funciones lambda y filter/map/reduce.

Instrucción:

- Dado un List<Int> de calificaciones:
- Filtra las notas aprobadas (> 60).
- Calcula promedio, máxima, mínima.
- Agrupa en categorías: Excelente, Bueno, Regular, Malo con when.

Conceptos: filter, map, reduce, funciones de orden superior, when.

9. Simulador de login

Objetivo: Control de flujo, uso de estructuras y validación.

Instrucción:

- Crea una función que valide usuario y contraseña con una Map.
- Permitir 3 intentos.
- La contraseña debe tener al menos 6 caracteres y un número.

Conceptos: bucles, condicionales, Map, validación de strings.

10. Aplicación de lógica combinada

Objetivo: Aplicar múltiples conceptos en un solo reto.

Instrucción:

- Desarrolla un sistema para gestionar libros:
- Cada libro tiene título, autor y número de páginas.
- Almacena en una lista de mapas.
- Permite: agregar, listar por autor, buscar por título, mostrar promedio de páginas.

Conceptos: listas de mapas (List<Map<String, Any>>), funciones, bucles, lógica.