Difencia estre LIST y SET vs MAPS –

**LIST y SET ->** Tienen objetos

MAPS- > Por ser clave-valor , no agrupan objetos.

**Colección LIST**

* Por defecto tiene 10 valores , pero se le agregan 10 cuando llega a 10 .
* Orden natural de los elementos.
* Tienen un indice.
* Tiene interfaz inmutable

List **mutables** : Se puede cambiar valores de la lista  Imagen que contiene Teams

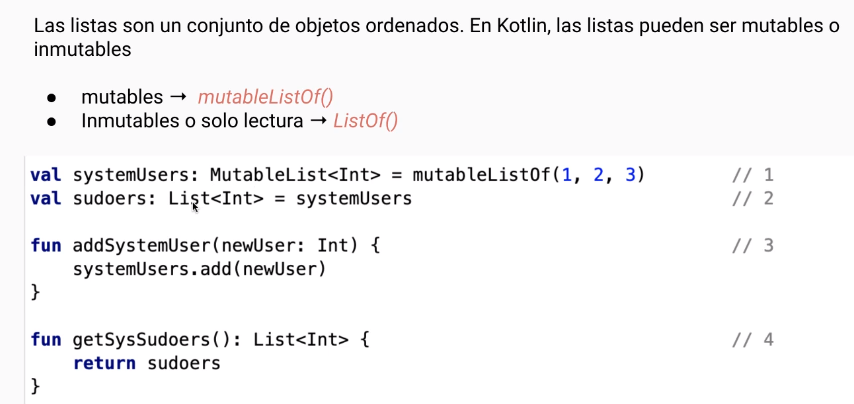
Descripción generada automáticamente

**REMOVER OBJETOS DE UNA LISTA**

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

List **inmutables** : No se pued emodificar en contenidos de la lista/objetos

* Elementos accedidos por un indice , y que no importa el orden , y no importa que esten repetidos . se usa lista
* 

Add - > se agrega algo a la lista

* Colección ->**SET**

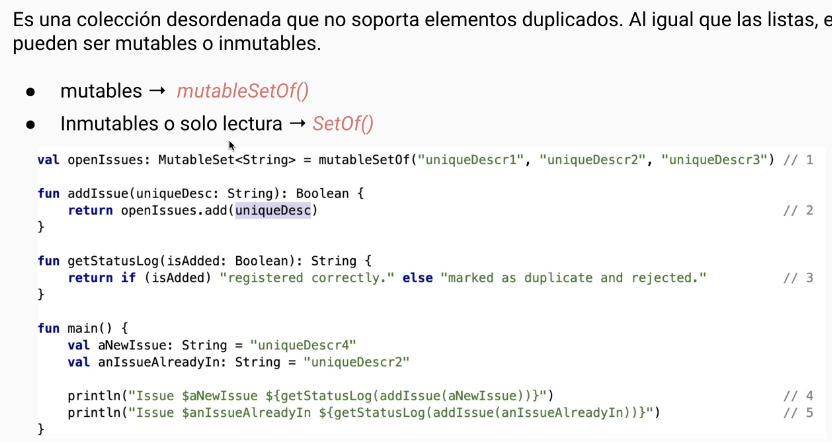
-Interfaz heradada de coleccions

- No admite elementos repetidos

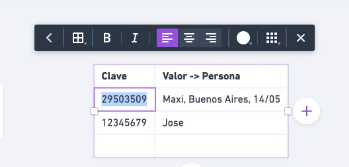
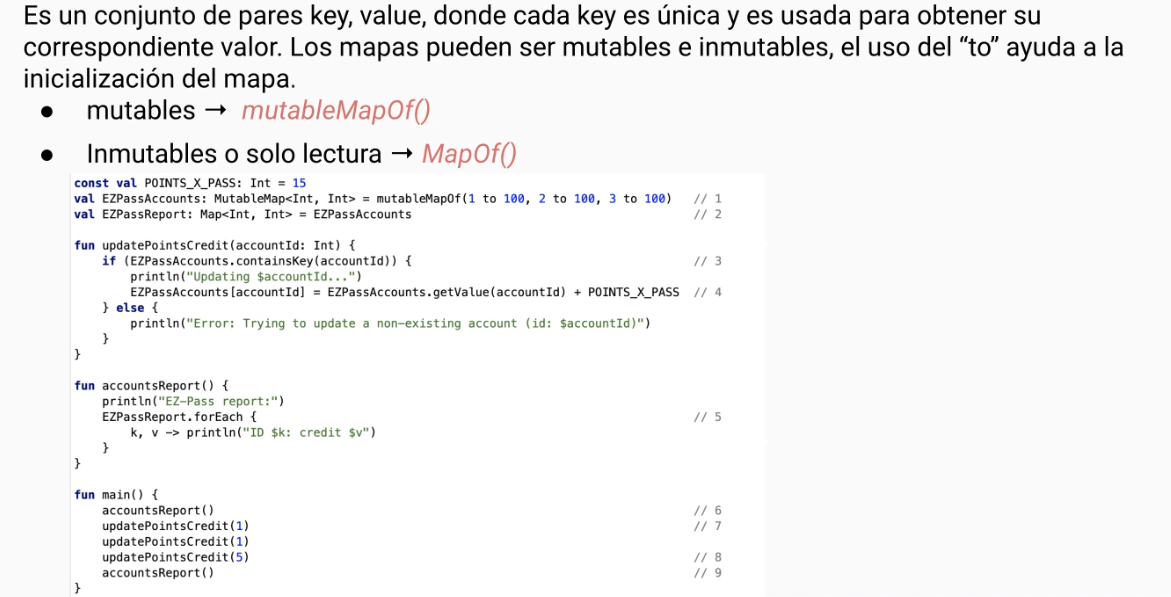
- No estan ordenados, requiere implemetacion de un comparador para ordenarlos.

- Interfaz Inmutable

- **CARACTERISTICA PRINCIPAL , SI NECESITO LISTAS QUE NO ACEPTAN DUPLICADOS**



No es una colección -> **MAP**

* **Clave - Valor**
* Clave que identifica a un objeto
* Mapa = tabla.
* Clave no se puede repetir
* La clavee no puede ser nula
* Se accede a los elementos por clave
* Intefaz Inmutable
* 
* Mapa puede representar estructuras de datos complejas. 

**Se necesita un objeto MAP para usarlos.**

**containsKey ->** pregunta al mapa si existe o no la clave

.**getValue (accountId) - >** devuelve el value de la accountId