



A hierarquia de coleções do **Kotlin** tem distinção entre coleções **mutáveis** e **imutáveis**:



Quando trabalhando com **coleções mutáveis** é interessante **proteger o acesso e modificação** que podem ser feitos **nelas** com **encapsulamento**, tornando a **collection privada** e **retornando uma cópia dela** para leitura:

```
class bancoDeNomes(){
    private val dados = mutableListOf<String>()

    val nomes: Collection<String> get() = dados.toList()

    fun salvar(nome:String){
        dados.add(nome)
    }
}
```



**Set** é uma coleção de elementos únicos, ou seja, um conjunto de objetos que não se repetem, **Set** não é ordenado. Por **não ter ordem**, **não é possível acessar elementos do Set através de um índice**.

A principal diferença do **Kotlin** com relação a **Set** são as operações **"infix"** (elas não exigem **"."** ou **sinhal**) que você pode fazer com dois conjuntos de **Set**, como **unir eles(soma-los)**, **subtrai-los** e fazer a **intersecção** (Manter os elementos que são comuns em ambos os conjuntos):

```
val conjunto1 = setOf("Lauro", "Marcio", "Yourmungard", "Jorel")
val conjunto2 = setOf("Lauro", "Beltrona", "Julia", "Jorel")

conjunto1 union conjunto2
conjunto1 subtract conjunto2
conjunto1.intersect(conjunto2)
```