



**Transformations** é um recurso do **Kotlin** para **transformar coleções de um tipo em outro**, ele faz isso através da **própria função de Map do Kotlin**. O mais interessante nesse caso é o **"Associate"** que basicamente é fazer a **associação dos elementos de uma lista a algum valor único** (um mapa onde elemento é o valor e o valor único é a chave).

Então fazer a **associação** é um **jeito simples de criar um mapa** definindo a **chave** e o **valor**:

```
val pedidos = listOf(
    Pedido( numero: 1, BigDecimal( val: "67")),
    Pedido( numero: 2, BigDecimal( val: "99")),
    Pedido( numero: 3, BigDecimal( val: "765"))
)

val pedidosMapeados = pedidos.associate { pedido: Pedido -> Pair(pedido.numero, pedido) }

val pedidosMapeados = pedidos.associateBy { pedido -> pedido.numero }
```

Um método de associação para criação de um mapa baseado em condições é o **associateWith**. Basicamente a ideia é **criar um mapa onde os valores atendem determinada condição**:

```
val pedidosFreteGratis = pedidos.associateWith { pedido -> pedido.valor >= BigDecimal( val: "160") }
```

**\*\*Não é**

**uma boa prática usar booleano como chave em um associate, pois eles não agrupam mais que um valor \*\***

Então a melhor maneira de agrupar elementos por uma condição é utilizar o **groupBy()**, dessa maneira eu vou ter uma **coleção como valor** de uma **chave booleana**:

```
val pedidos = listOf(
    Pedido( numero: 1, BigDecimal( val: "67")),
    Pedido( numero: 2, BigDecimal( val: "99")),
    Pedido( numero: 3, BigDecimal( val: "765"))
)

val pedidosFreteGratis = pedidos.groupBy { pedido -> pedido.valor > BigDecimal( val: "77") }
```

MainKt x

"C:\Program Files\Java\jdk-15\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community

{false=[Pedido(numero=1, valor=67)], true=[Pedido(numero=2, valor=99), Pedido(numero=3, valor=765)]}