

RxLearning

Operadores de transformação

- Transformam os elementos de uma sequência:
Observable<T> -> Observable<U>
- **map**
- **flatMap e variantes**

map

- Mesmo comportamento do **map** aplicado em collections

```
let myInts = [1, 2, 3, 4, 5]
let myStrings = myInts.map { value in
    "\(value)"
}
// ["1", "2", "3", "4", "5"]
```

... e em um Observable

```
let myInts = Observable.of(1, 2, 3, 4, 5)
let myStrings = myInts.map { value in
    "\(value)"
}
// myStrings: Observable<String>
```

flatMap

- O que um **flatMap** faz em um array normal?

```
let arrayOfArrays = [
  [1, 2, 3],
  [4, 5, 6]
]
let flatArray = arrayOfArrays.flatMap { $0 }
//flatArray = [1, 2, 3, 4, 5, 6]
```

... e em um Observable

```
let myInts = Observable.of(1, 2, 3, 4, 5)
let flatSequence = myInts.flatMap { value -> Observable<Float> |in
  return someOtherObservable // Observable<Float>
}
// Observable<Float>
```

**SHOW ME THE
CODE**

AND STOP TALKING!

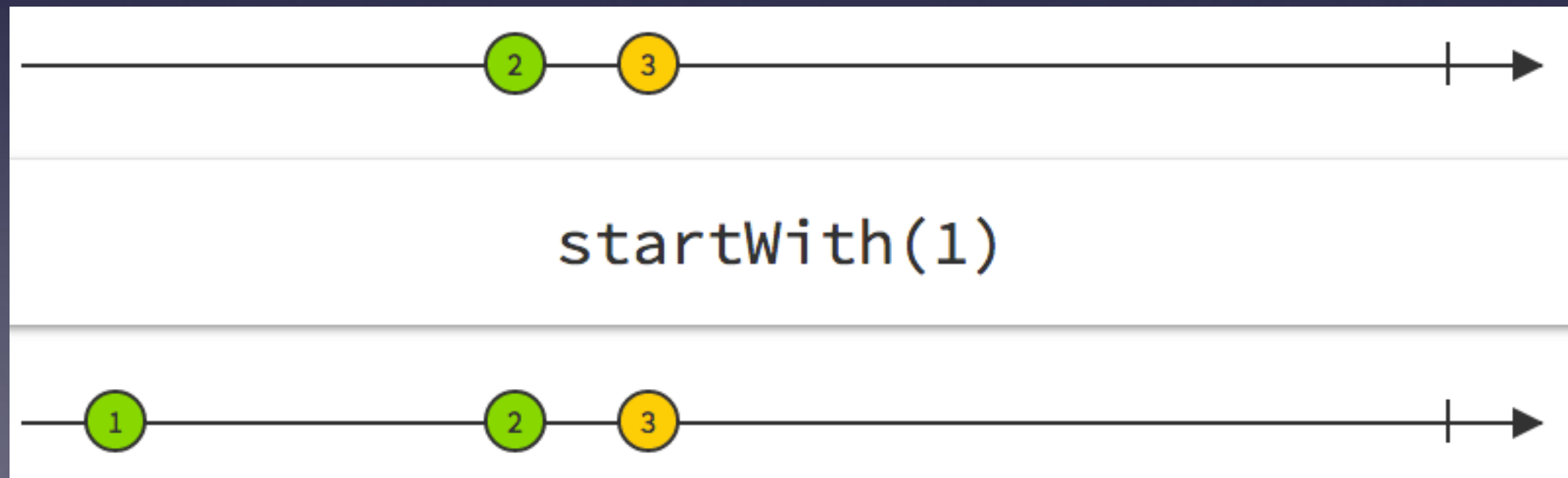
memegenerator.net

Operadores de combinação

- Combinam sequências ou elementos e sequências
 - **startWith**
 - **concat**
 - **merge**
 - **combineLatest**
 - **zip**
 - **withLatestFrom**

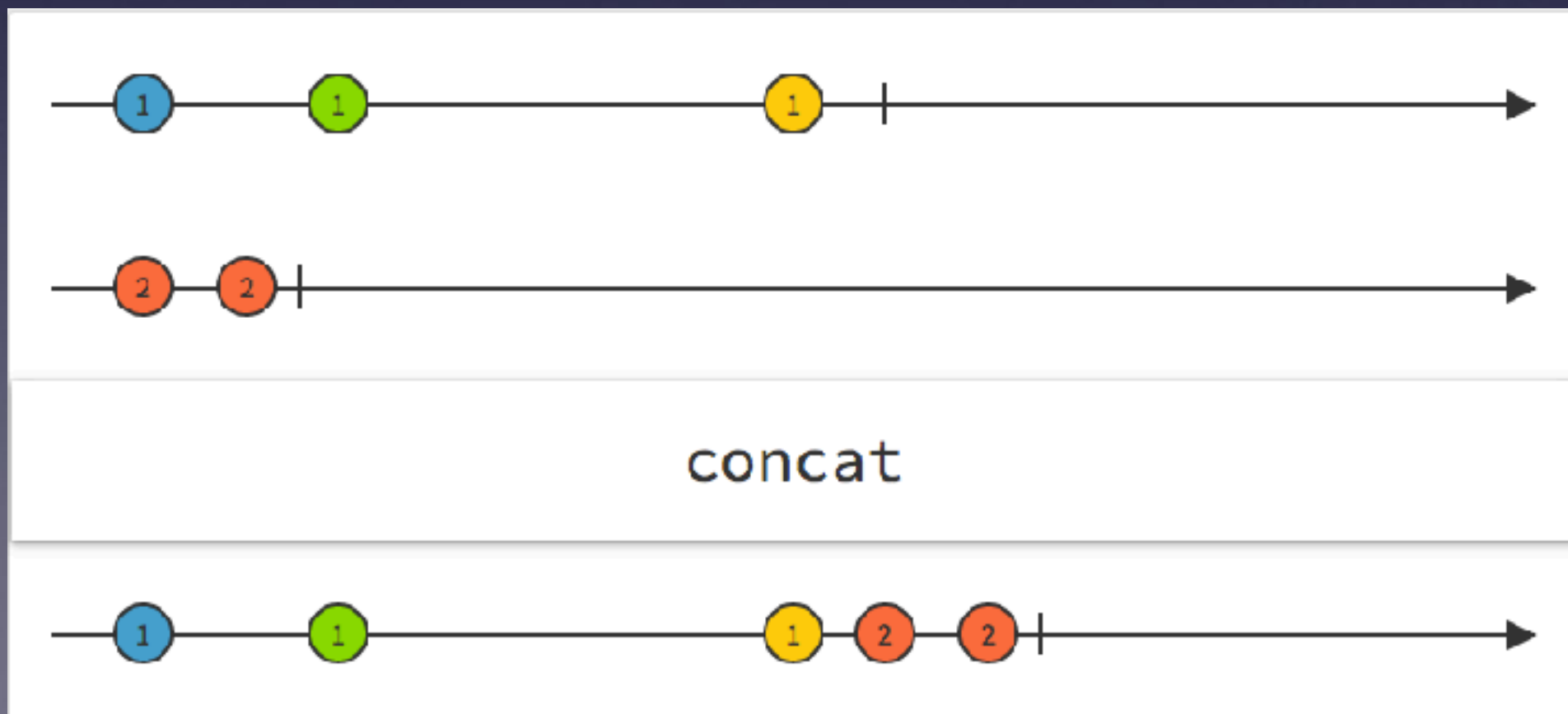
startWith

- Emite como primeiro elemento da sequência um valor desejado.
- O valor precisa ser do mesmo tipo que o tipo emitido pela sequência



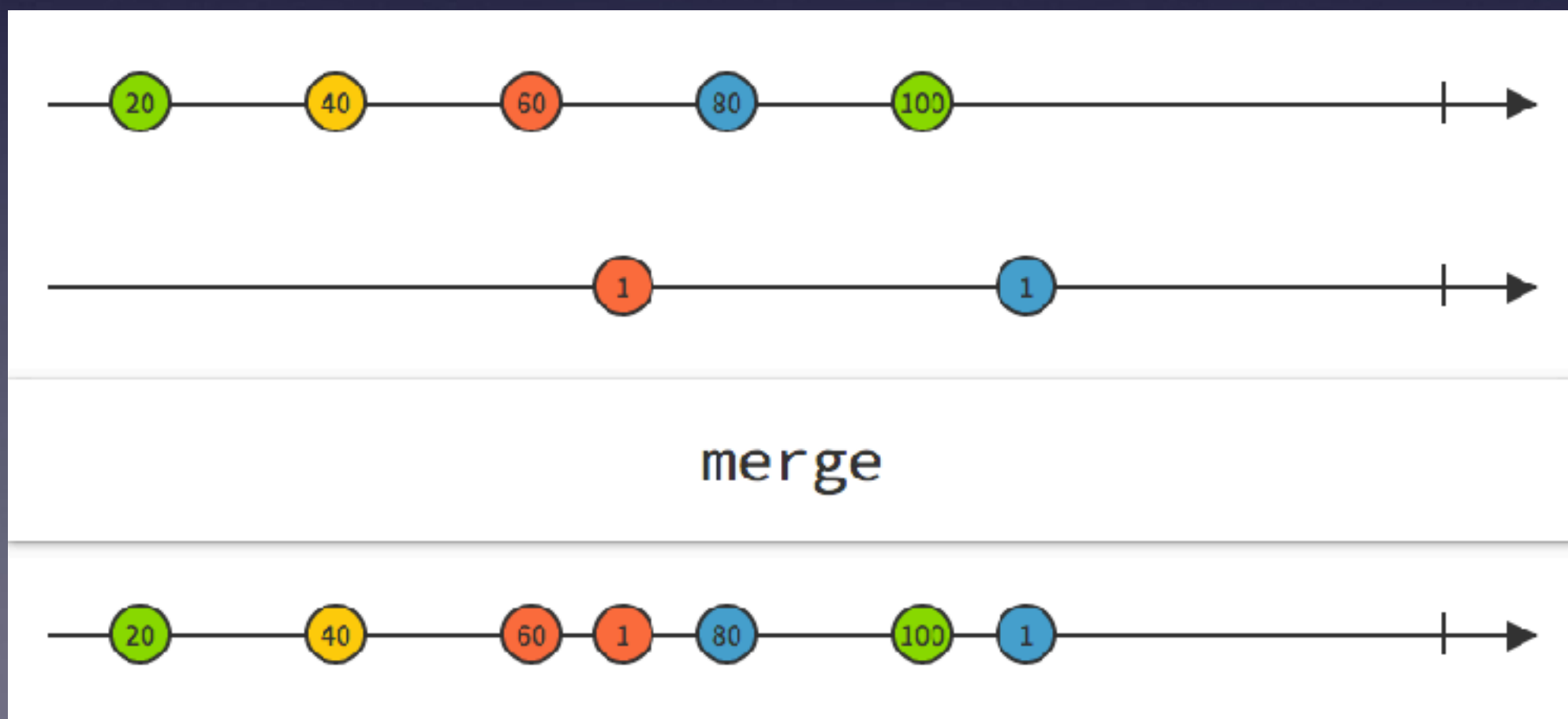
concat

- Concatena os elementos de duas sequências.
- Ambas devem possuir o mesmo tipo
- Primeira sequência deve completar para que a segunda comece



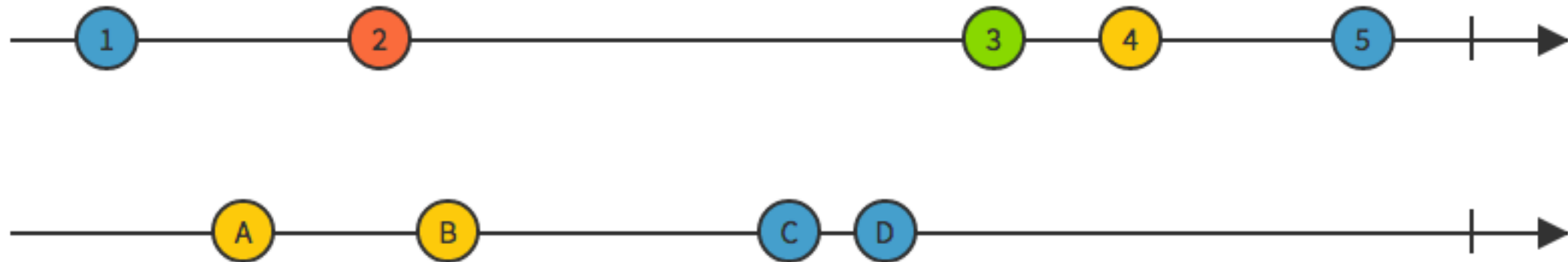
merge

- Observa elementos das sequências e emite os elementos assim que cada sequência emite
- Sequências devem possuir o mesmo tipo



combineLatest

- Observa elementos das sequências e combina as últimas emissões das mesmas
- Sequências podem possuir tipos distintos

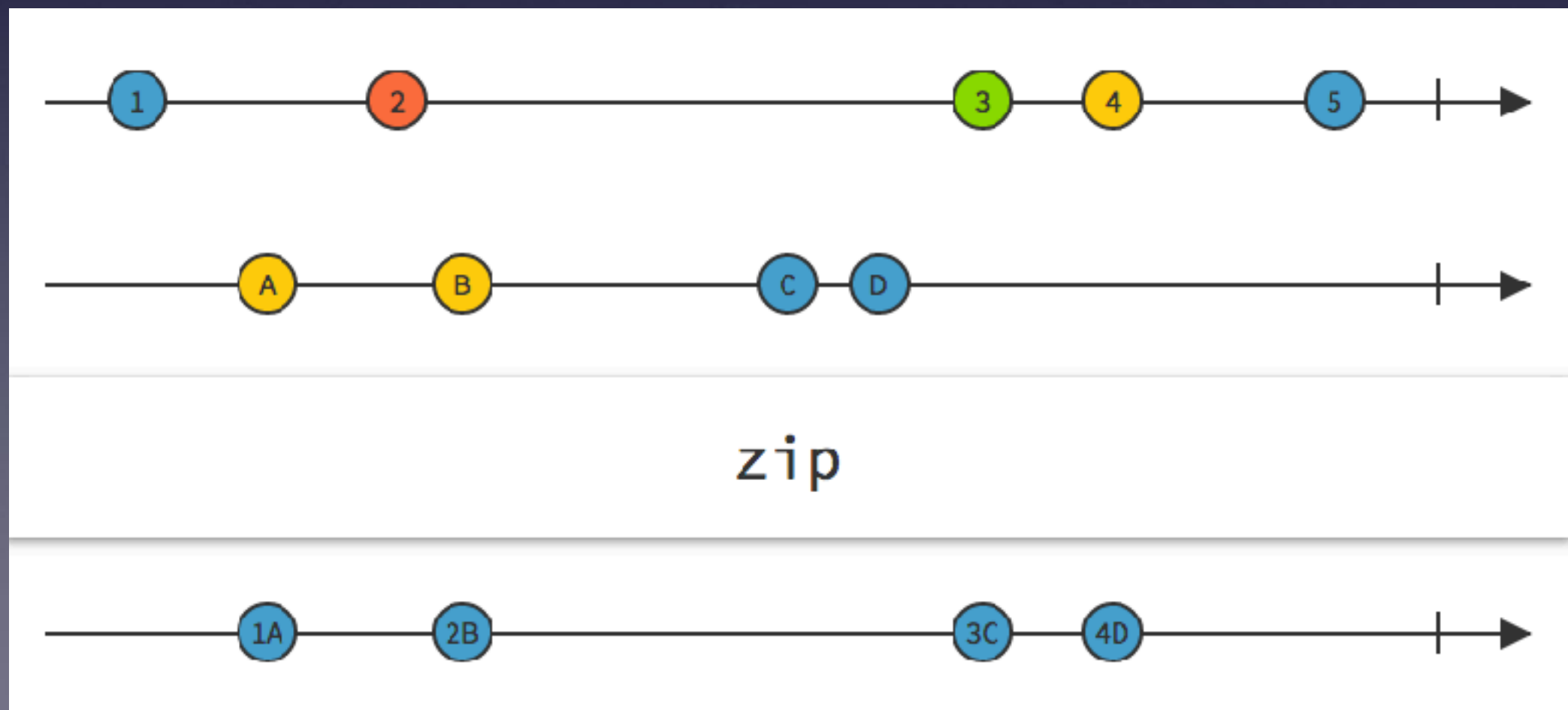


```
combineLatest((x, y) => "" + x + y)
```



zip

- Observa sequências e combina elementos de mesmo índice
- Idêntico ao zip em arrays
- Sequências podem possuir tipos distintos



withLatestFrom

- Emite o último elemento emitido por uma sequência alvo, a cada emissão de uma sequência gatilho.
- Pode ou não combinar os elementos de ambas sequências.
- Sequências podem ter tipos distintos

