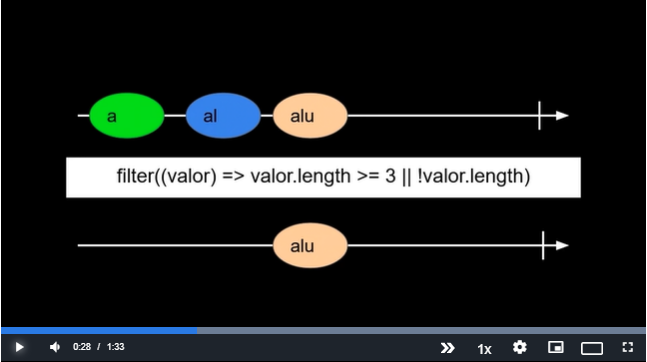
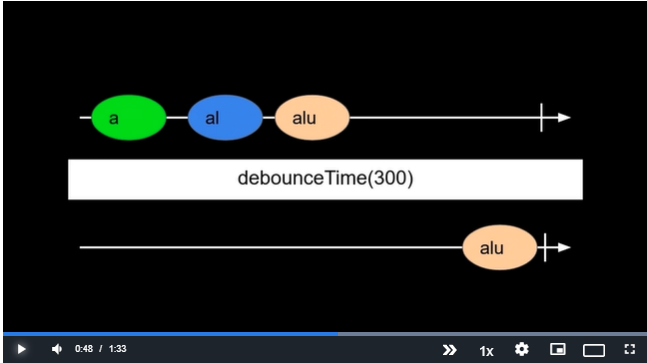
[00:00] Vamos agora fixar melhor esses novos operadores que colocamos na nossa caixa de ferramentas do RxJS, utilizando o Gráfico de Marbles. Temos, primeiro, o filtro onde colocamos nossa condição filter((valor) => valor.length >= 3 || !valor.length) e quando digitamos a letra "a", ela não atende o critério e não passa para a frente; "al" também não passa no critério e não é passada para o próximo operador. Quando digitamos "alu", esse termo tem mais que 3 caracteres e passa no critério.

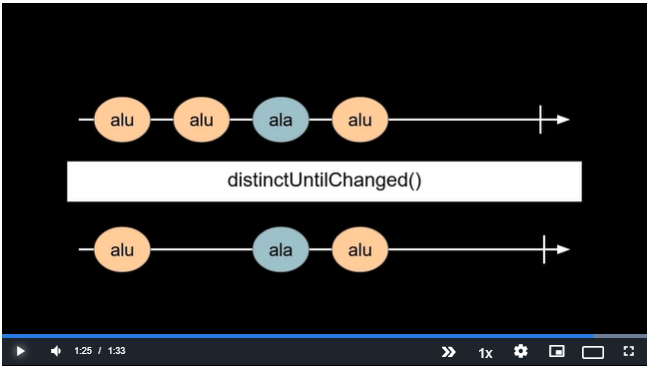


[00:29] No debounceTime(300), a questão não é o que passa no critério e sim quando. Temos a letra "a", depois a letra "l" e, por fim, o termo "alu". O termo "alu" será passado para frente, mas, depois de um tempo que, nesse caso, foi definido como 300 milissegundos.



[00:48] Por fim, temos o distinctUntilChanged(), onde, no exemplo, temos o termo "alu", que é passado para frente do fluxo. Quando temos o termo "alu" novamente, o operador filtra esse valor e não passa para frente, aí, o usuário digita o termo "ala".

[01:10] O "ala" é passado para o próximo fluxo. O termo "alu" aparece, de novo, que é passado para frente, já que nesse exemplo podemos reparar que o operador trabalha, apenas, com o último valor e não com o fluxo inteiro.



[01:25] Temos agora todos os operadores fixados, tanto pelo código como pelo gráfico.