

01

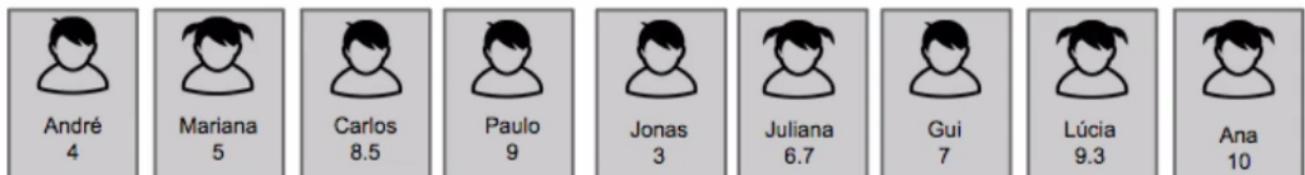
## Dividindo para conquistar

### Transcrição

Fomos capazes de intercalar dois *arrays* já ordenados. Se dividirmos as cartas de baralho - o dinheirinho falso, notas dos alunos ou outros elementos -, entre diversas pessoas e cada uma ordenar uma parte do total, intercalar é uma tarefa mais simples. É assim quando temos vários *arrays*...

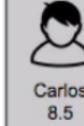


Mas o que acontece muitas vezes? Um professor chega e deixa uma pilha de provas. Depois outro professor coloca em cima uma nova pilha. Ou seja, as provas vão sendo amontoadas uma nas outras. Desta forma, não teremos dois *array* separados e organizados, um do Aniche e outro do Alberto. Na verdade, teremos uma única pilha, um só *array*, formada por dois grupos ordenados separadamente.



Em vez de termos um *array* de tamanho 4 e outro de tamanho 5, para depois criarmos uma lista de tamanho 9. O que costuma acontecer é que temos um *array* de tamanho 9, em que os quatro primeiros elementos (que seguem até o miolo) estão ordenados do menor para o maior, da mesma forma estão ordenados os próximos cinco itens. Começamos do 0 e terminamos no 9. O inicial é 0 e o termino é 9, enquanto o miolo é o 4. Então, na prática, o que acontece com frequência é que recebemos um *array* (e não dois), com duas partes ordenadas. Nosso objetivo é intercalar estes dois pedaços. Alguém chega e nos diz: "Aqui estão o meu monte de cartas e o seu. Agora encontre uma maneira de intercalar todos os itens." Nós fazemos um monte único e mandamos intercalar.

inicial: 0	
miolo: 4	
termino: 9	

								
André 4	Mariana 5	Carlos 8.5	Paulo 9	Jonas 3	Juliana 6.7	Gui 7	Lúcia 9.3	Ana 10



Aniche  
total: 4

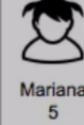
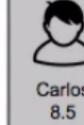
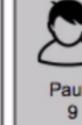


Alberto  
total: 5

Logo, tudo o que fizemos até agora com dois *arrays*, teremos que fazer algumas alterações no nosso código para trabalharmos com uma lista única.

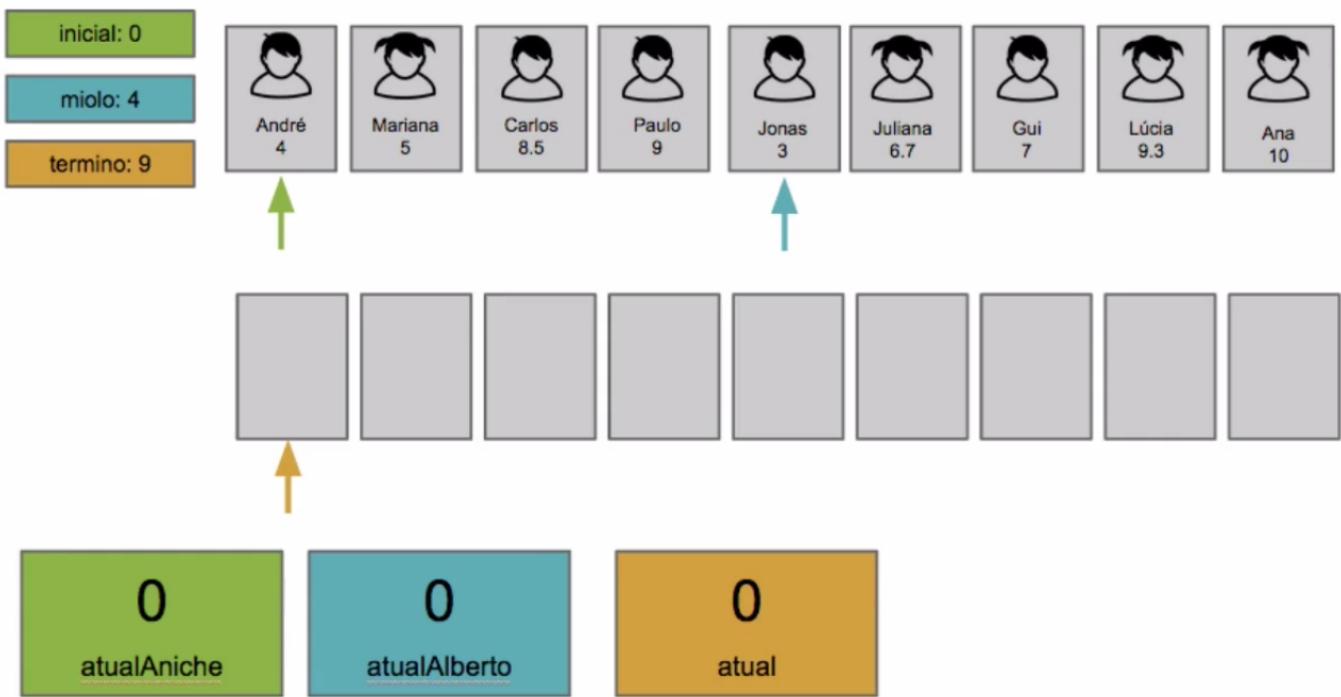
Voltando ao nosso algoritmo de intercalar dois *arrays*, nós sabemos que recebemos um único *array*. O total de elementos se refere ao `termino`.

inicial: 0	
miolo: 4	
termino: 9	

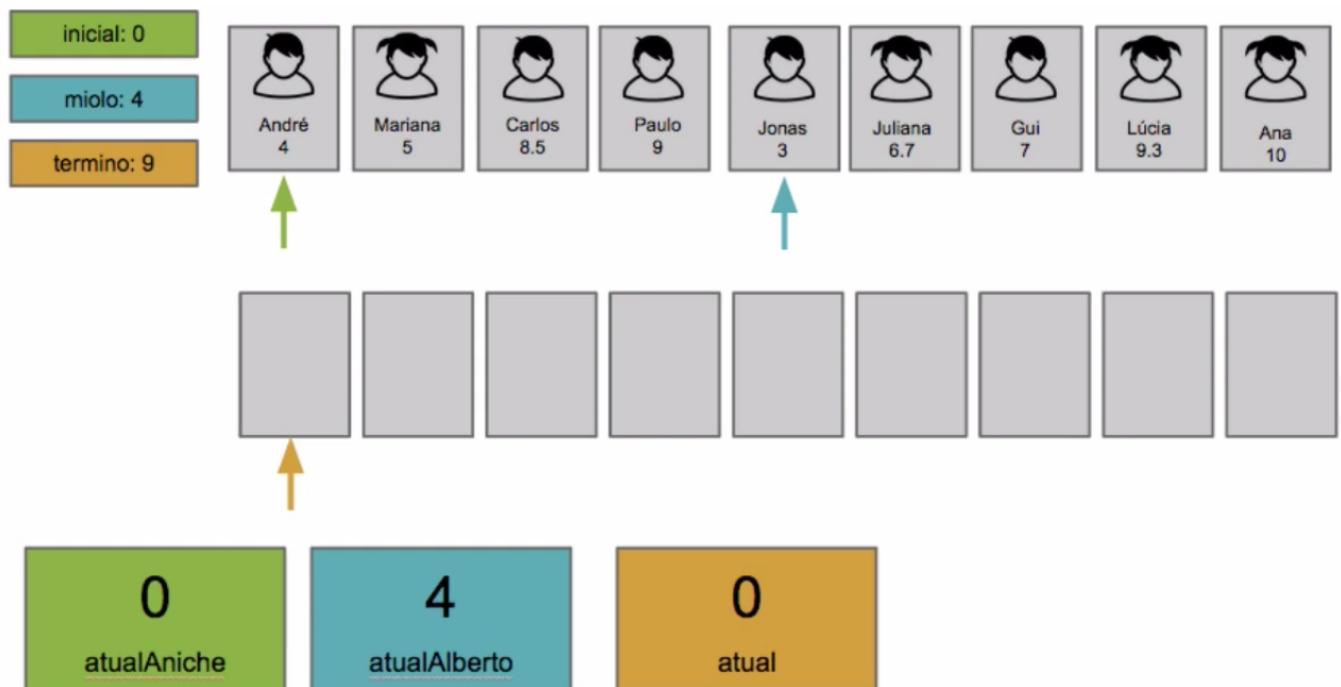
								
André 4	Mariana 5	Carlos 8.5	Paulo 9	Jonas 3	Juliana 6.7	Gui 7	Lúcia 9.3	Ana 10

O primeiro elemento é o 0. Então como nós sabemos qual é a primeira parte e qual é a segunda? Com a variável `miolo`.

Na prática, o que temos agora é algo muito parecido com o que fizemos antes. Temos o `atualAniche` que começa com o valor do `inicial` e o `atualAlberto` que começa com o valor do `miolo`. O valor do `termino` é 9, o tamanho do *array*. A variável `atual` também como era antes, começa com 0.



A melhor maneira para começarmos é pelo `atualAlberto`, que inicia no `miolo`. Isto significa que começaremos pelo valor 4. Vamos rodar o nosso algoritmo.



Iremos comparar as notas do André, que está no `atualAniche`, e do Jonas, que está no `atualAlberto`. Qual dos dois é o menor? Você irá perceber que é o mesmo algoritmo utilizado antes!

O menor elemento é o Jonas, então iremos movê-lo para o `array` geral, somamos +1 nas variáveis `atualAlberto` e `atual`.