

- Caso 1: $x = v[\text{meio}]$
O algoritmo retorna verdadeiro, funcionando corretamente.
- Caso 2: $x < v[\text{meio}]$
Se $x < v[\text{meio}]$ o valor x estará entre as posições *inicio* e *meio-1*, inclusive. A cada chamada recursiva a quantidade de posições candidatas se restringirá à meio-1 posições de $v[\text{inicio}]$ a $v[\text{meio}-1]$. Na última chamada, a posição inicial será $v[\text{inicio}]$ e a final será $v[\text{inicio}]$ ou $v[\text{inicio}+1]$.
Se a posição final for $v[\text{inicio}]$ e o elemento existir no vetor $v[\text{inicio}] = v[\text{meio}]$, então o algoritmo retorna verdadeiro, funcionando corretamente.
Se a posição final for $v[\text{inicio}+1]$ e o elemento existir no vetor $v[\text{inicio}] = v[\text{meio}]$, se $x = v[\text{inicio}]$ o algoritmo retorna verdadeiro, funcionando corretamente. Caso contrário, $x = v[\text{fim}] > v[\text{meio}]$, portanto provaremos essa situação no caso 3.
- Caso 3: $x > v[\text{meio}]$
Se $x > v[\text{meio}]$ o valor x estará entre as posições *meio+1* e *fim*, inclusive. A cada chamada recursiva a quantidade de posições candidatas se restringirá à meio-1 posições de $v[\text{meio}+1]$ a $v[\text{fim}]$.
Na última chamada recursiva a posição inicial será $v[\text{fim}]$ e a final também será $v[\text{fim}]$, então $v[\text{meio}] = v[\text{fim}]$, se o elemento existir no vetor o algoritmo retorna verdadeiro, funcionando corretamente.