

ESQUEMA DE MEMÓRIA :

1. Estado inicial:

- a. `arr[] = {100, 70, 90, 90, 10, 40, 30}`
- b. `limInf = 0`
- c. `limSup = 6`
- d. Recursão ainda não começou.

2. Após a primeira chamada de Quicksort:

- a. `Pivô = 30`.
- b. Array após a partição: `{10, 30, 90, 90, 100, 40, 70}`.
- c. `ind = 1`.

3. Durante a recursão:

- a. A próxima chamada recursiva vai processar a sublista antes e depois do pivô.
- b. Para cada chamada recursiva, a memória contém uma nova instância das variáveis da função, como `limInf`, `limSup`, e `pivo`.

Como que fica na prática:

Eu posso desenhar a pilha de execução de cada ciclo da recursão. Por exemplo:

1. Chamada inicial: `quickSort(arr, 0, 6)`

- a. Estado do array: `{10, 30, 90, 90, 100, 40, 70}`
- b. `Pivô: 30`
- c. Novas chamadas: `quickSort(arr, 0, 0)` e `quickSort(arr, 2, 6)`.

2. Em `quickSort(arr, 2, 6)`:

- a. `Pivô: 70`.
- b. Estado do array: `{10, 30, 40, 70, 100, 90, 90}`.

