

REALOCAMENTO

Pseudocódigo

```
// Variáveis para controle
TextoMemoria = ""
TamanhoMemoria = 0

// Encontrar processo e seu tamanho
Para Pos de 1 até 104 Faça
    Se Memoria[Pos] ≠ Nulo E Memoria[Pos - 1] = Nulo Então
        TextoMemoria = Memoria[Pos]
        Enquanto Memoria[Pos] = TextoMemoria Faça
            TamanhoMemoria++
            Pos++
        Se Pos > 104 Então Sair Do Loop
    Fim Enquanto
    Sair Do Loop
Fim Se
Fim Para

// Apagar a posição atual do processo
Para Pos de 0 até 104 Faça
    Se TextoMemoria = Memoria[Pos] Então
        Memoria[Pos] = Nulo
    Fim Se
Fim Para

// Realocar na nova posição
TamanhoMemoria--
Para Pos de 0 até 104 Faça
    Se Memoria[Pos] = Nulo E TamanhoMemoria >= 0 Então
        Memoria[Pos] = TextoMemoria
        TamanhoMemoria--
    Fim Se
Fim Para
```

C#

```
//Variaveis de controle
string textMemoria = "";
int tamanhoMemoria = 0;

//Pega o processo e o seu tamanho
for (int pos = 1; pos < 105; pos++)
{
    if (DadosGlobais.memoria[pos] != null && DadosGlobais.memoria[pos - 1] ==
null)
    {
        textMemoria = DadosGlobais.memoria[pos];
        while (DadosGlobais.memoria[pos] == textMemoria)
        {
            tamanhoMemoria++;
            pos++;
            if (pos > 104) { break; }
        }
        break;
    }
}
//Apaga a posição atual do processo
for (int pos = 0; pos < 105; pos++)
{
    if (textMemoria == DadosGlobais.memoria[pos])
    {
        DadosGlobais.memoria[pos] = null;
    }
}
//Realoca na nova posição
tamanhoMemoria--;
for (int pos = 0; pos < 105; pos++)
{
    if (DadosGlobais.memoria[pos] == null && tamanhoMemoria >= 0)
    {
        DadosGlobais.memoria[pos] = textMemoria;
        tamanhoMemoria--;
    }
}
```