

Listex 2 - Ex 3

João Pedro Machado Silva, BV3032477

```
int procuraValor(int arrayA[], int n, int valorProcura){
    for(int i = 1; i <= n; i++){
        if(arrayA[i-1] == valorProcura){
            return i;
        }
    }
    return -1;
}
```

● Finitude:

- No laço “for”, o valor “i” recebe o valor 0 e aumenta a cada iteração, sendo que precisa ser menor que “n”. O valor de “i”, em algum momento, chegará a “n-1”, que é a condição de parada. Ou seja, em algum momento, o contador “i” atingirá o valor “n” após “n-1” iterações. Portanto, o algoritmo para!

● Corretude:

- **Invariante de laço:** $A[1 \dots i-1]$
 - Antes da 1-ésima iteração começar, $i = 1$ e os elementos de $A[1 \dots 1-1]$ já foram acessados.
 - Antes da 2-ésima iteração começar, $i = 2$ e os elementos de $A[1 \dots 2-1]$ já foram acessados.
 - Antes da 3-ésima iteração começar, $i = 3$ e os elementos de $A[1 \dots 3-1]$ já foram acessados.
 - ...
 - Na última iteração começar, $i = n+1$ e os elementos de $A[1 \dots n]$ já foram acessados.

- Analisando o item anterior, através da Invariante de Laço, fica provado que o algoritmo funciona.

- **Complexidade de Tempo:**

- **Pior Caso:**

Linha	Custo	Vezez
2	c1	n
3	c2	n-1
7	c3	1

$$T(n) = c1(n) + c2(n-1) + c3$$

Temos uma função Linear!

- **Melhor Caso:**

Linha	Custo	Vezez
2	c1	1
3	c2	1
4	c3	1

$$T(n) = c1 + c2 + c3$$

Temos uma função Linear!