Algoritmo utilizado:

```
int procurar(int *array, int tamArray, int procurado){

    for(int i=0; i<tamArray; i++){
        if(array[i] == procurado){
            return i+1;
        }
    }

    return NULL;
}
```

Analise de Finitude

Para a analise de finitude do algoritmo, deve-se observar se existe alguma situação na qual as estruturas de repetição não encontram uma saída. Para a estrutura utilizada no algoritmo, podemos demonstrar facilmente que não existe caso na qual a estrutura se mantenha infinitamente, pois o “i” é sempre incrementado a condição de existência da estrutura implica em “i” ser menor do que o tamanho do Array. Assim, em algum momento o “i” sempre passará este número e então, a estrutura de repetição se encerrará.

Estrutura de repetição utilizada:

```
for(int i=0; i<tamArray; i++){
    if(array[i] == procurado){
        return i+1;
    }
}
```

Analise de Corretude

Invariante Principal: A cada iteração, o laço para e retorna à posição(i+1) caso Array[i] possua o número procurado. Caso contrário, o laço continua.

Propriedades necessárias:

- Na primeira iteração (i=0), caso Array[0] contenha o número procurado, o laço irá ser encerrado. Caso contrário, o algoritmo segue para a próxima interação.

- Em uma iteração aleatória, caso o Array[i] contenha o número procurado, o programa não avançara para a próxima iteração. Caso o número não seja encontrado, então a próxima iteração ocorre.

- Na ultima iteração, caso o Array[i] contenha o número procurado, será retornado a posição (i+1). Caso contrário, o programa retornará NULL, mostrando assim que o número não pode ser encontrado no Array.

Analise de Complexidade de Tempo

Melhor caso

Linha	Código	Custo	# execuções
1	for(int i=0; i<tamArray; i++){	c1	1
2	if(array[i] == procurado){	c2	1
3	return i+1;	c3	1
4	}	0	0
5	}	0	0
6	return NULL;	c6	0

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

Pior caso

Linha	Código	Custo	# execuções
1	for(int i=0; i<tamArray; i++){	c1	n
2	if(array[i] == procurado){	c2	n-1
3	return i+1;	c3	0
4	}	0	n-1
5	}	0	n-1
6	return NULL;	c6	1

$$T(n) = c1*n + c2*(n-1) + c6$$

$$T(n) = c1*n + c2*n - c2 + c6$$

$$T(n) = (c1 + c2)*n + c6 - c2$$