


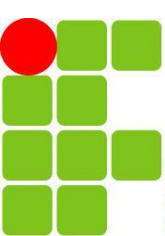
Vetores

Programação 1

Rafael Vargas Mesquita

<http://www.ci.ifes.edu.br>
<ftp://ftp.ci.ifes.edu.br/informatica/rafael/>





Declaração de Vetores

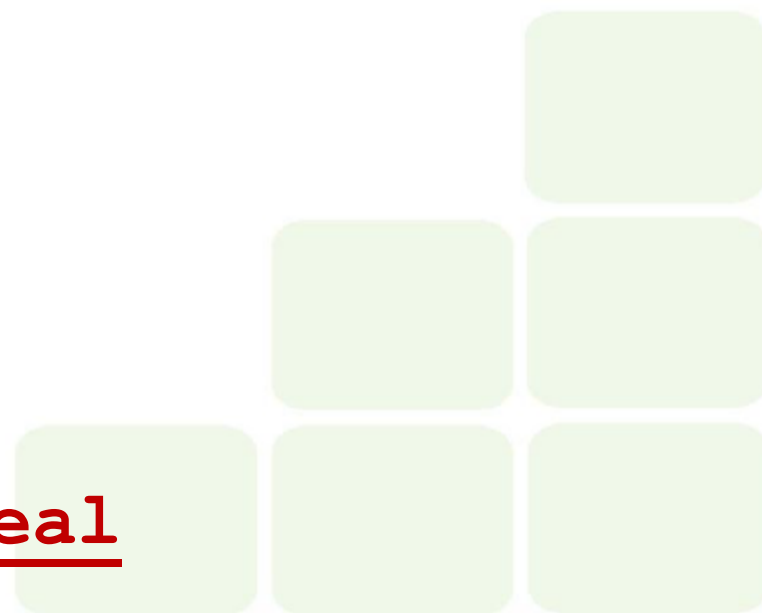
- No **Visualg 2.0** a declaração de vetores é feita da seguinte forma:

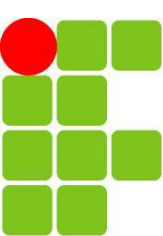
`<nome do vetor> : vetor[inicio..fim] de <tipo>`

- Onde:
 - inicio: índice inicial do vetor
 - fim: índice final do vetor

- Exemplo:

`x : vetor[1..10] de real`





Declaração de Vetores

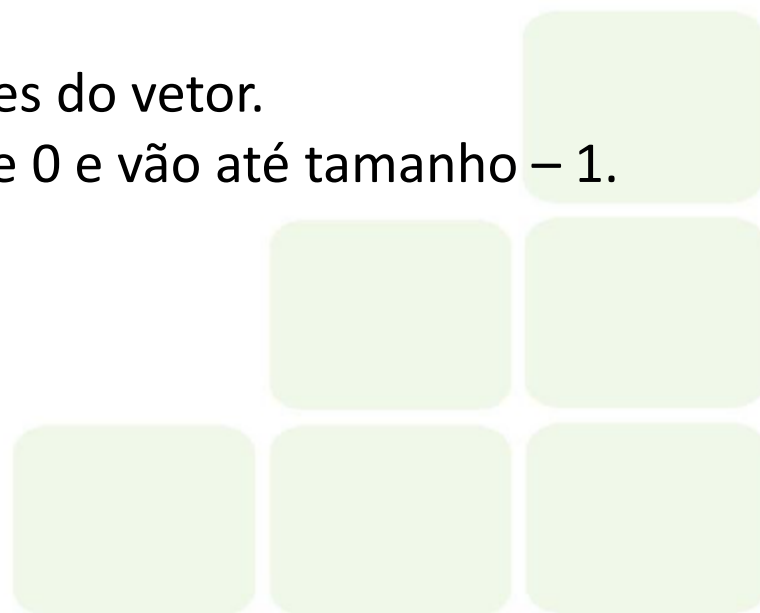
- Em **Portugol** a declaração de vetores é feita da seguinte forma:

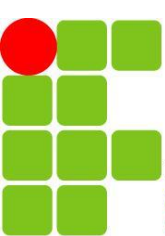
`<tipo de cada posição do vetor> <nome do vetor>[tamanho];`

- Onde:
 - tamanho: é a quantidade de posições do vetor.
 - Em Portugol os vetores começam de 0 e vão até tamanho – 1.

- Exemplo:

`real x[10]`





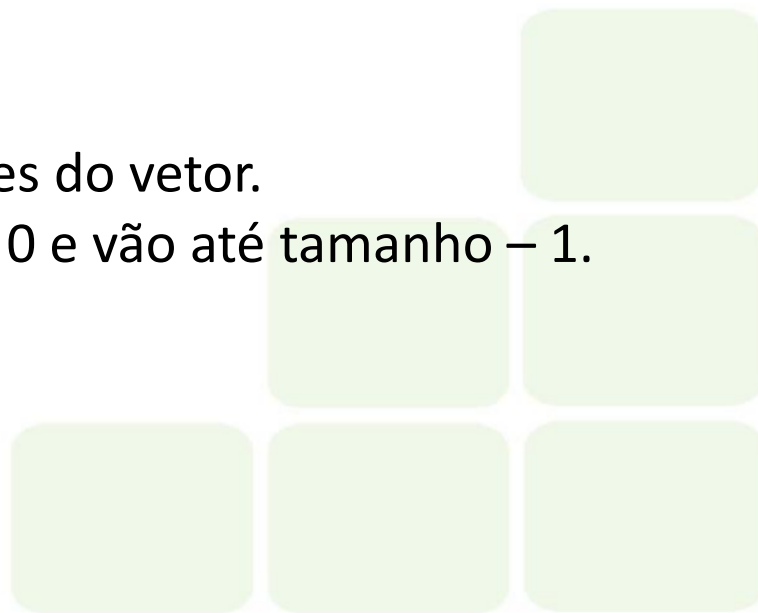
Declaração de Vetores

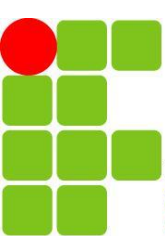
- Em **C** a declaração de vetores é bastante semelhante, obedecendo ao seguinte formato:

`<tipo de cada posição do vetor> <nome do vetor>[tamanho];`

- Onde:
 - tamanho: é a quantidade de posições do vetor.
 - Em C todos os vetores começam de 0 e vão até tamanho – 1.
- Exemplo:

```
float x[10];
```





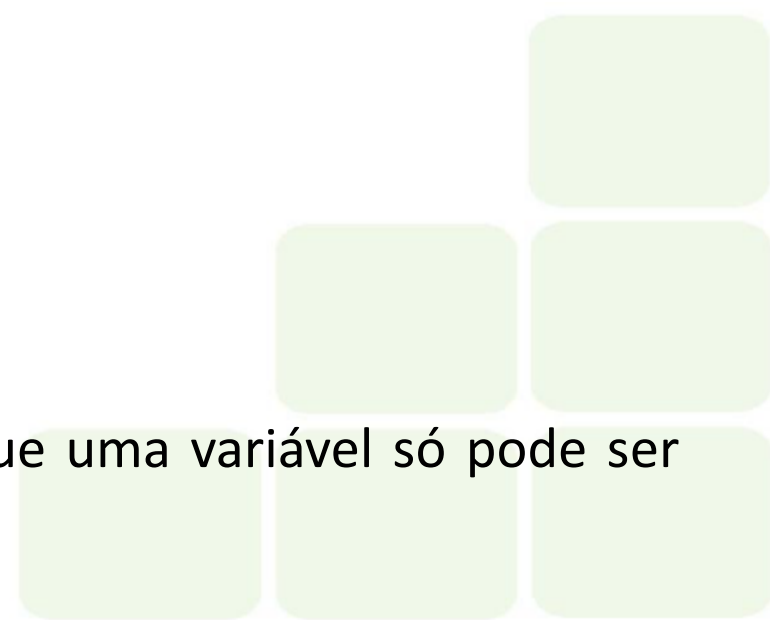
Local de Declaração

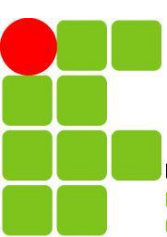
- Em **C**, os vetores, como qualquer outra variável, podem ser declarados em qualquer função, não existindo uma sessão para definição de variáveis:

<Declaração de funções>

```
int main() {  
    <Declaração de variáveis>  
    <Sequência de comandos>  
    <Declaração de variáveis>  
    <Sequência de comandos>  
}
```

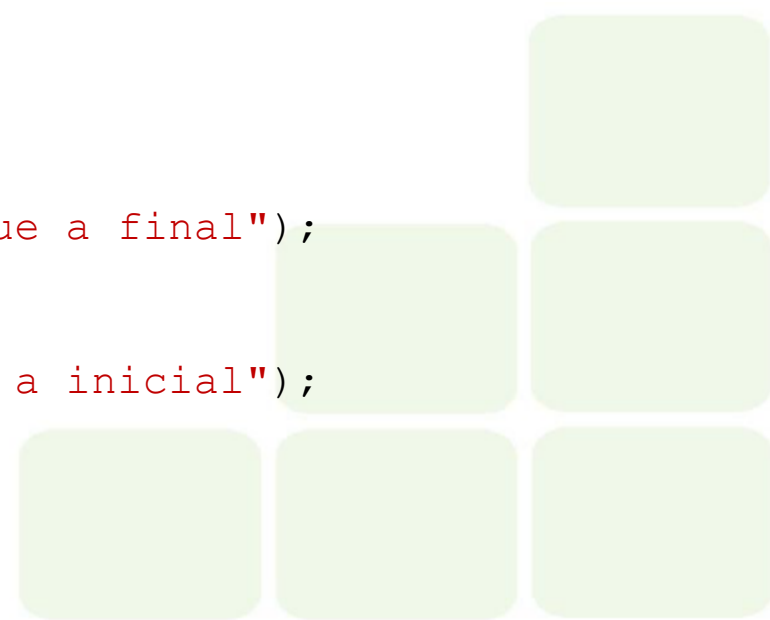
- Obs.: Deve ser obedecida a restrição que uma variável só pode ser utilizada depois de declarada.





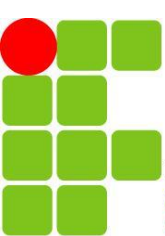
Local de Declaração

```
int main () {  
    int num[2];  
    printf ("Digite 2 numeros: ");  
    scanf ("%i", &num[0]);  
    scanf ("%i", &num[1]);  
    int num2[2];  
    printf ("Digite 2 numeros: ");  
    scanf ("%i", &num2[0]);  
    scanf ("%i", &num2[1]);  
    num[0] = num[0] + num2[0];  
    num[1] = num[1] + num2[1];  
    printf ("\n\n");  
    if (num[0] > num[1]) {  
        printf("Soma inicial maior que a final");  
    }  
    else {  
        printf("Soma final maior que a inicial");  
    }  
    return 1;  
}
```



Passando Vetores como Parâmetros

- Em **C** a passagem de uma variável do tipo vetor como parâmetro para uma função pode ser realizada.
- Importante:
 - Um vetor só pode ser passado como parâmetro por referência;
 - Consequência: A função chamada sempre pode alterar os valores do vetor passado como parâmetro pela função chamadora.
 - O nome do vetor é o seu endereço;



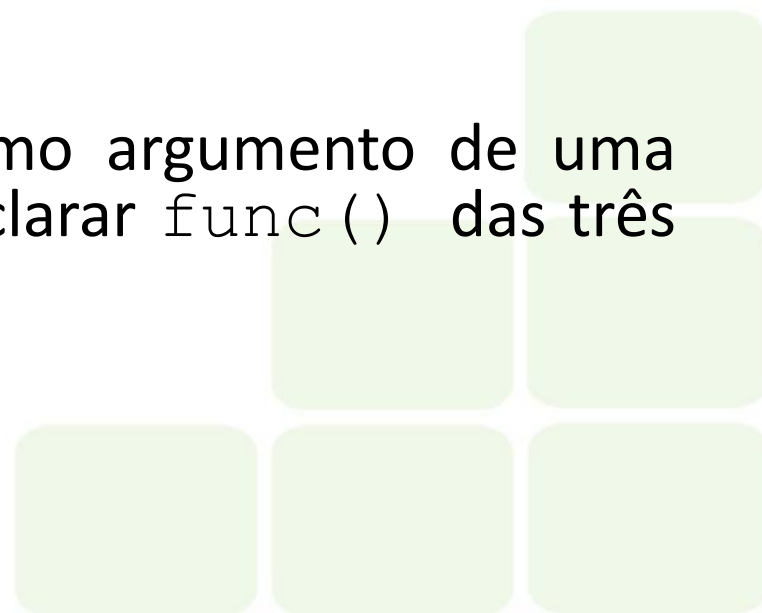
Passando Vetores como Parâmetros

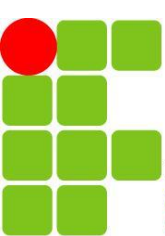
- Exemplo:
 - Quando vamos passar um vetor como argumento de uma função, podemos declarar a função de três maneiras equivalentes. Seja o vetor:

```
int vet[50];
```

- E que queiramos passá-la como argumento de uma função `func()`. Podemos declarar `func()` das três maneiras seguintes:

```
void func (int vet[50]);  
void func (int vet[]);  
void func (int *vet);
```





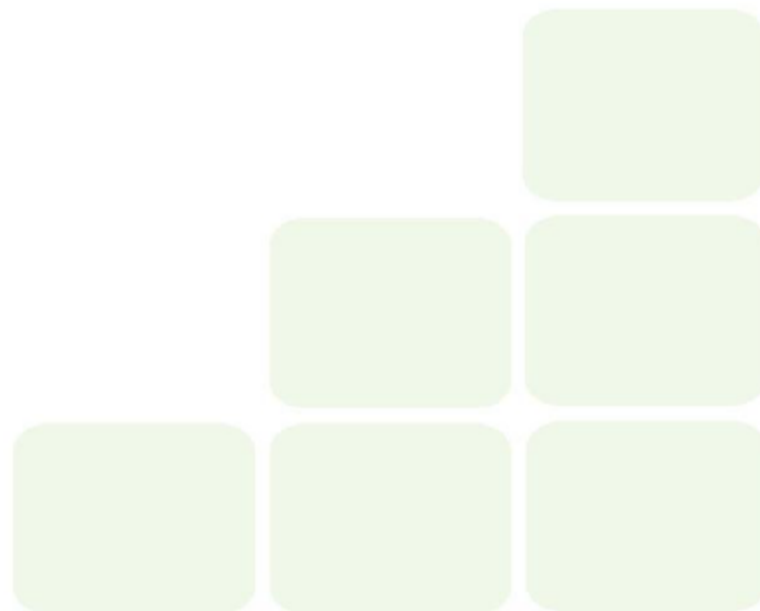
Passando Vetores como Parâmetros

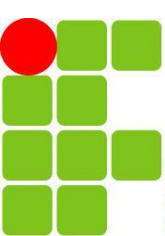
- Exemplo:

```
#include<stdio.h>
```

```
void ler_dados(int vet[10]) {  
    for(int i = 0; i <= 9; i++) {  
        scanf("%i", &vet[i]);  
    }  
}
```

```
int main() {  
    int vetor[10];  
    ler_dados(vetor);  
}
```





Bibliografia

- SANTOS, Henrique José. Curso de Linguagem C da UFMG, apostila.
- FORBELLONE, André Luiz. Lógica de Programação – A Construção de Algoritmos e Estruturas de Dados. São Paulo: MAKRON, 1993.

