

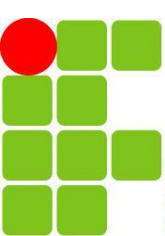
INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cachoeiro de Itapemirim

Funções e Procedimentos

Programação 1

Rafael Vargas Mesquita

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

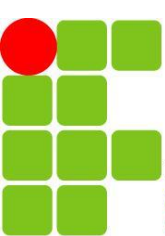


Declaração de Funções

- No Portugal a declaração de funções é feita da seguinte forma:

```
<tipo do retorno> <nome da função> (<tipo> <variável> ){  
    <sequência de comandos>  
}
```

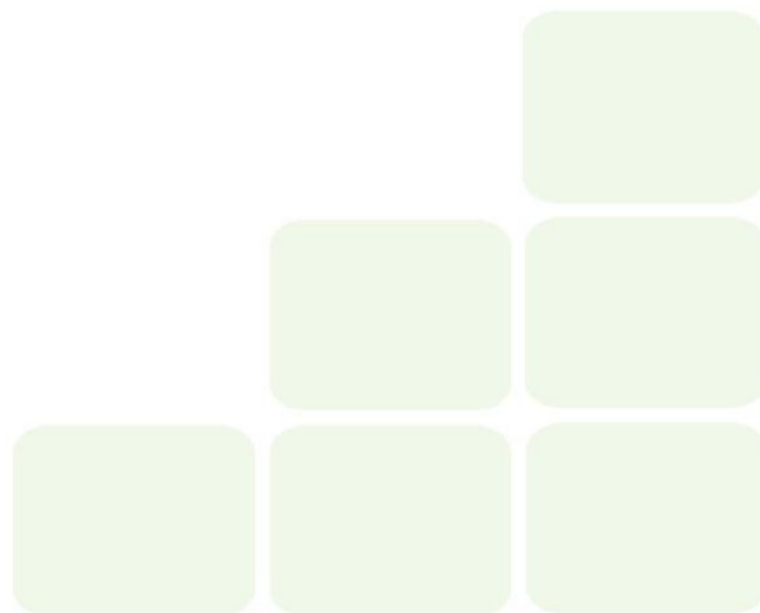


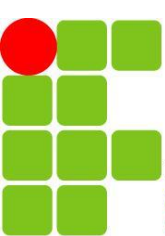


Declaração de Funções

- Em C a declaração de funções é bastante semelhante obedecendo ao seguinte formato:

```
<tipo do retorno> <nome da função> (<tipo> <variável> ){  
    <sequência de comandos>  
}
```

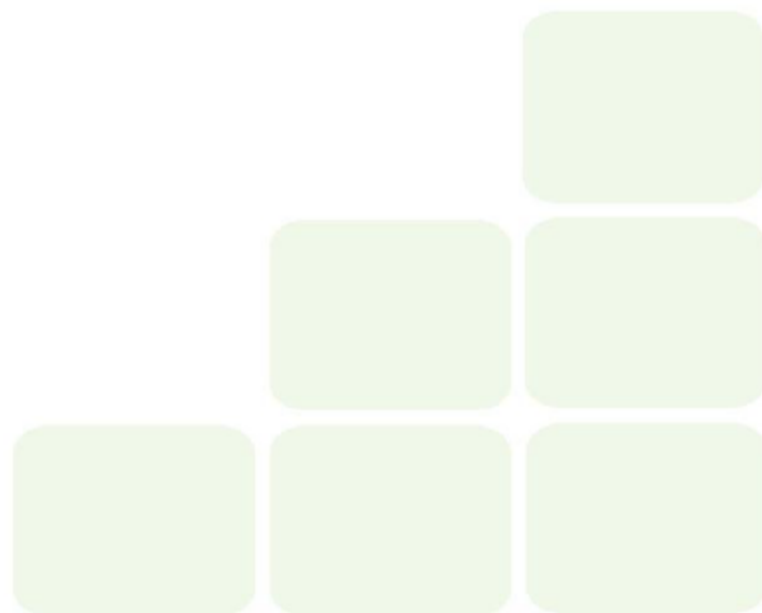


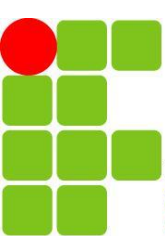


Declaração de Funções

- Exemplo:

```
/* Calcula o quadrado de x */  
int calcular_quadrado(int x) {  
    return (x*x) ;  
}
```



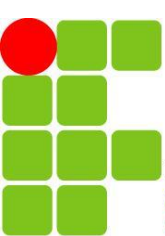


Local de Implementação

- Em C as funções devem ser declaradas antes da função *main*, obedecendo assim o seguinte formato:

```
<declaração de funções e procedimentos>  
  
int main() {  
    <sequência de comandos>  
}
```



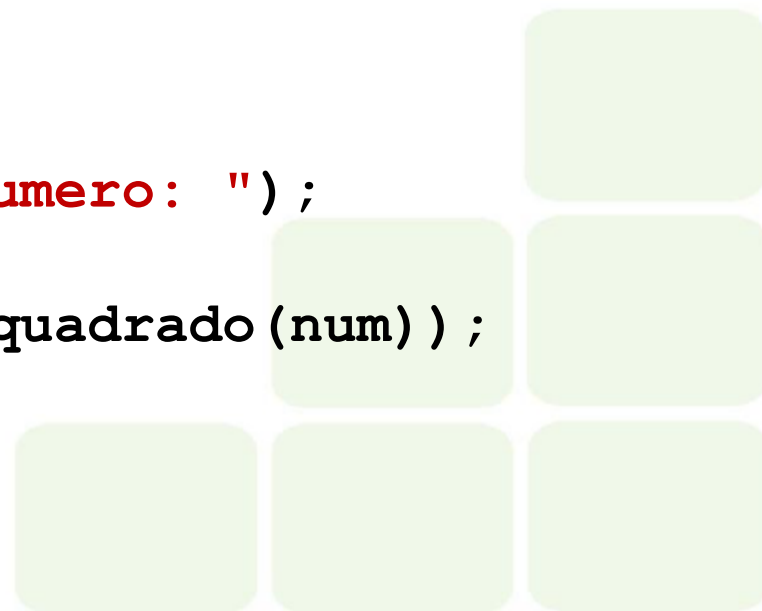


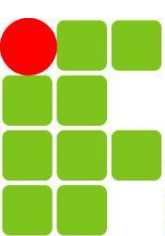
Local de Implementação

- Exemplo:

```
#include <stdio.h>
/* Calcula o quadrado de x */
int calcular_quadrado(int x) {
    return(x*x);
}

int main () {
    int num;
    printf ("Entre com um numero: ");
    scanf ("%i", &num);
    printf ("%i", calcular_quadrado(num));
    return(0);
}
```





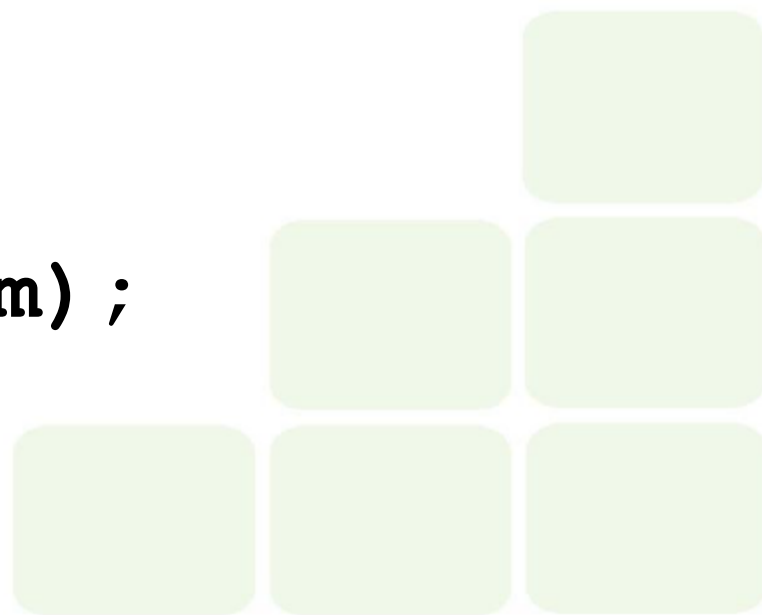
Uso de Função

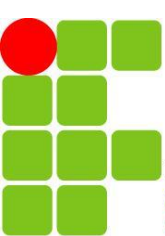
- Para utilizar uma função em C é bastante simples, basta utilizar a seguinte sintaxe:

`<nome da função>(<parâmetros de entrada>);`

- Exemplo:

`calcular_quadrado(num);`

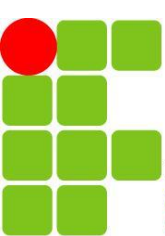




Procedimentos

- A linguagem C não trabalha o conceito de procedimentos, mas existe o tipo de retorno void, que em C significa que nada será retornado.
- Uma função que retorna void pode ser declarada da seguinte forma:

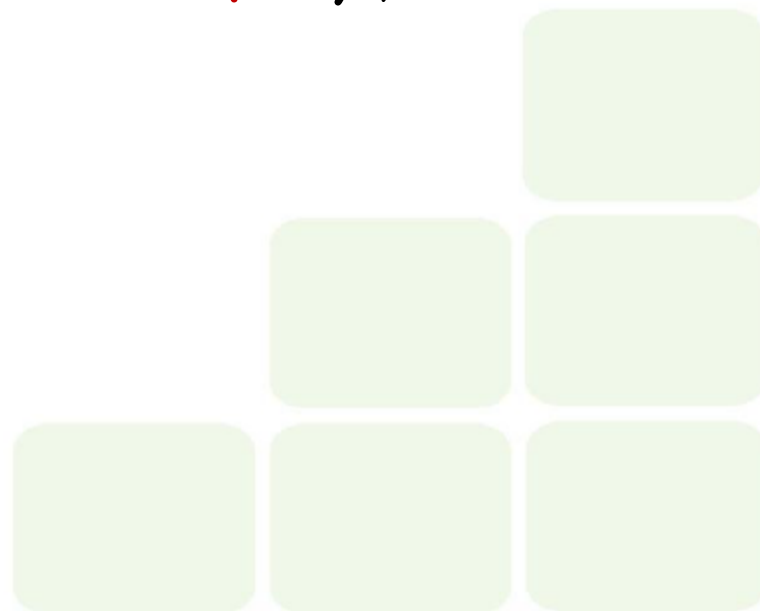
```
void <nome da função>(<tipo da variável> <variável>)
```

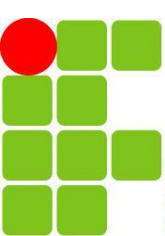



Procedimentos

- Exemplo:

```
void mensagem () {  
    printf ("Ola! Eu estou vivo.\n");  
}
```





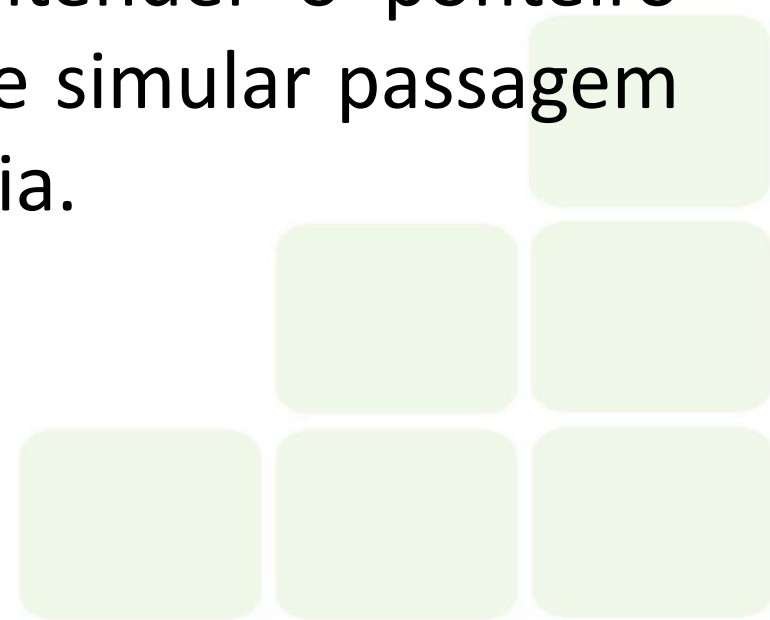
Passagem de Parâmetros por Valor

- Tanto em C como em Portugol a forma de passagem de parâmetros padrão é a por valor, que é o tipo de passagem de parâmetros que NÃO permite a alteração de variáveis no interior de funções.
- Exemplo:

```
int calcular_quadrado (int x) {  
    return x*x;  
}  
int main () {  
    int num;  
    ...  
    printf ("%i", calcular_quadrado (num) ) ;  
}
```
- X é passado por valor, sendo assim, quando a função calcular_quadrado é chamada recebendo num como parâmetro, ela não altera o valor de num.

Passagem de Parâmetros por Referência

- Em C não existe passagem de parâmetros por referência, para utilizar esse recurso são utilizados “ponteiros”.
- Nesse momento vamos entender o ponteiro apenas como uma forma de simular passagem de parâmetros por referência.



Passagem de Parâmetros por Referência

- No Portugol, a passagem de parâmetros por referência (em funções) é feita da seguinte forma:

```
<tipo do retorno> <nome da função> (<tipo> &<variável> ){  
    <sequência de comandos>  
}
```

Passagem de Parâmetros por Referência

- Em C precisaremos utilizar os “ponteiros”.
Façamos um exemplo de procedimento com passagem por referência em Portugol sendo transformada em passagem por valor usando ponteiros em C.

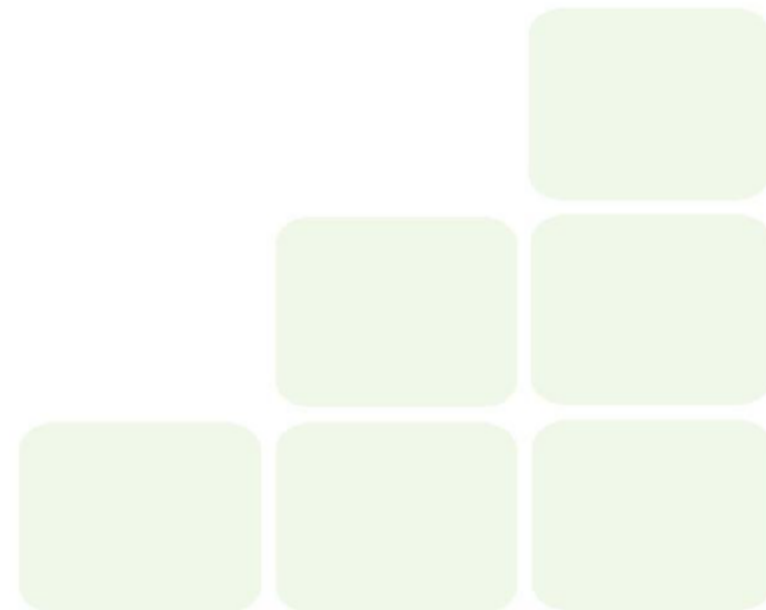


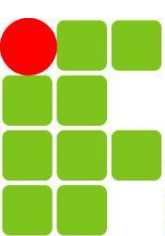
Passagem de Parâmetros por Referência

- Exemplo em Portugol:

```
void lerDados (int &num1, int &num2) {  
    num1 = 10;  
    num2 = 20;  
}
```

```
int main() {  
    int n1, n2;  
    lerDados(n1, n2);  
}
```



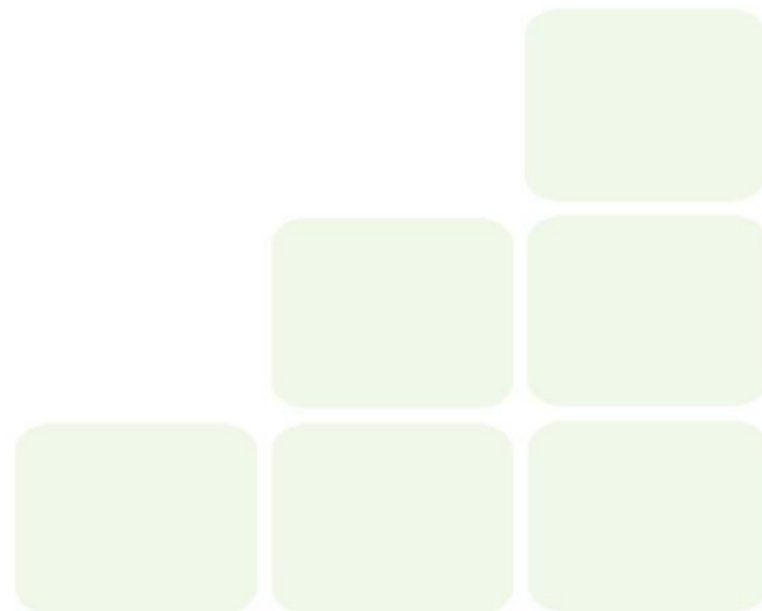


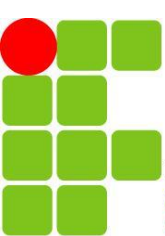
Passagem de Parâmetros por Referência

- Exemplo em C:

```
void lerDados (int *num1, int *num2) {  
    *num1 = 10;  
    *num2 = 20;  
}
```

```
int main() {  
    int n1, n2;  
    lerDados (&n1, &n2);  
}
```





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.

