

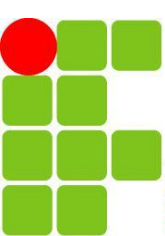
INSTITUTO FEDERAL
ESPÍRITO SANTO
Campus Cachoeiro de Itapemirim

Manipulação de Strings

Programação 1

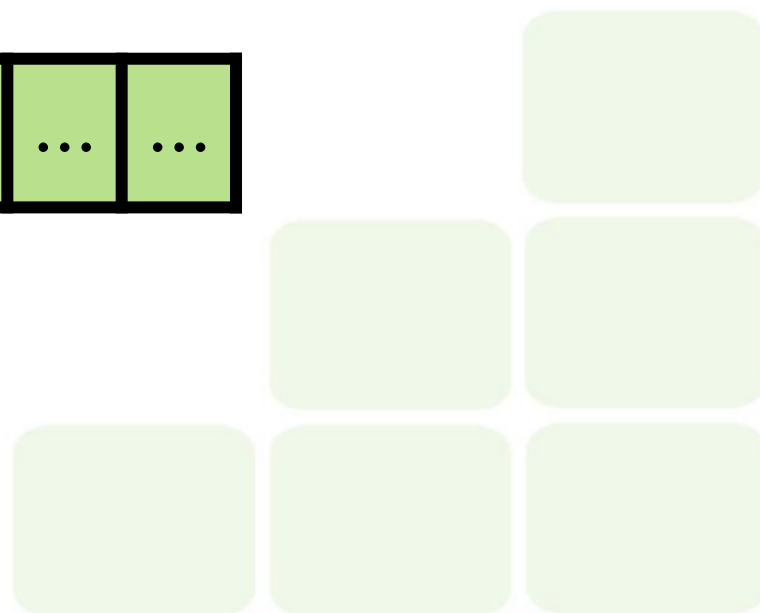
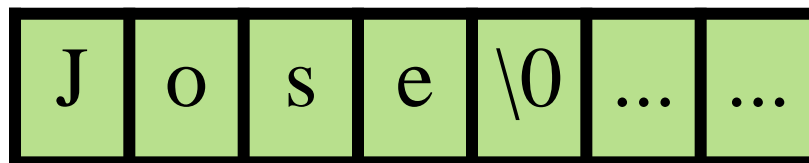
Rafael Vargas Mesquita

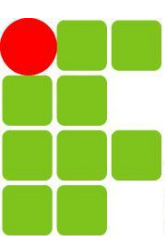
<http://www.ci.ifes.edu.br>
<ftp://ftp.ci.ifes.edu.br/informatica/rafael/>



Declaração de Strings

- São vetores de char que possuem o char '\0' definindo o término da string.
- Exemplo:



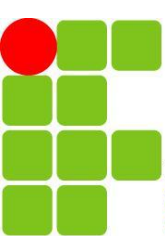


Declaração de Strings

- Como as strings são vetores de char podemos declarar strings da seguinte forma:

```
char <nome_string>[tamanho] ;
```





Uso Básico de Strings

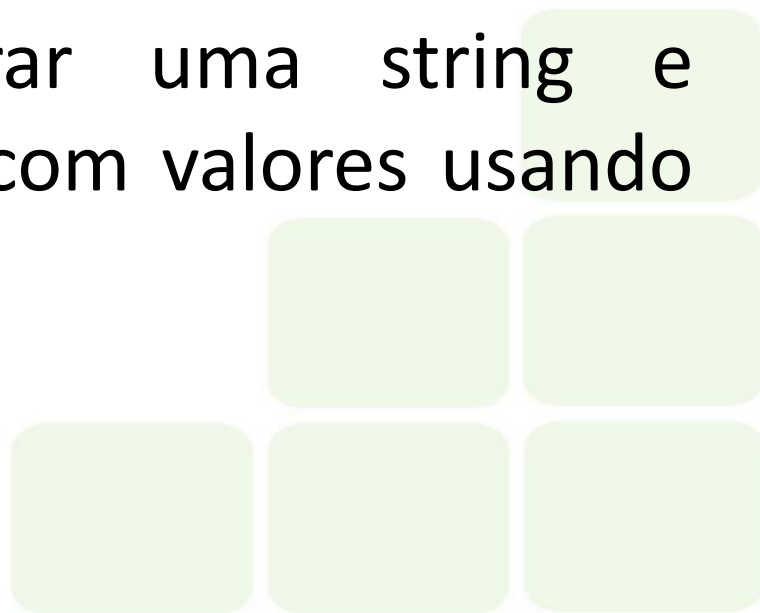
- Podemos declarar uma string e pré carregá-la com valores:

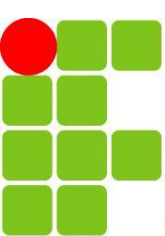
```
char str[10] = "Joao";
```

- Não é permitido declarar uma string e posteriormente carregá-la com valores usando o operador de atribuição:

```
char str[10];
```

```
str = "Joao";
```



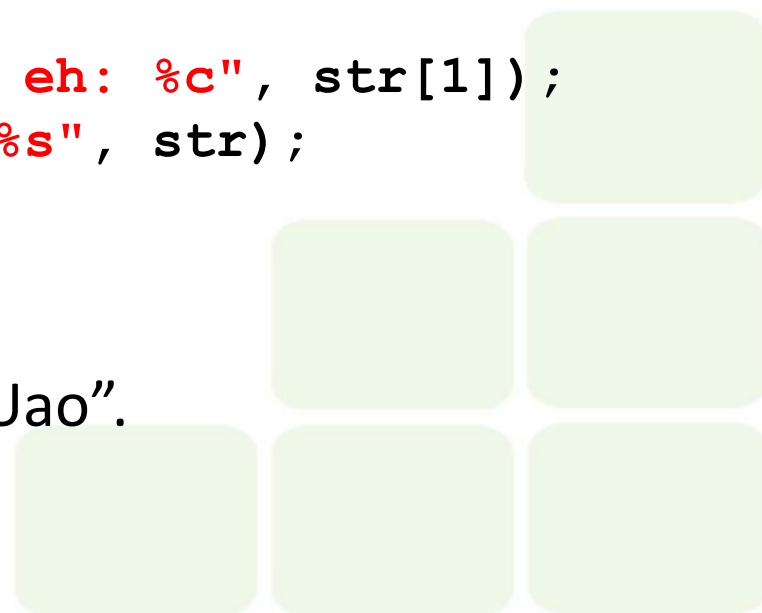


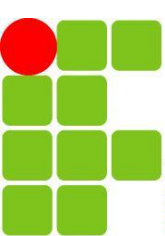
Uso Básico de Strings

- Podemos alterar valores de posições da string. Exemplo:

```
int main() {  
    char str[10] = "Joao";  
    printf("\n\nString: %s", str);  
    printf("\nSegunda letra: %c", str[1]);  
    str[1] = 'U';  
    printf("\nAgora a segunda letra eh: %c", str[1]);  
    printf("\n\nString resultante: %s", str);  
    return(0);  
}
```

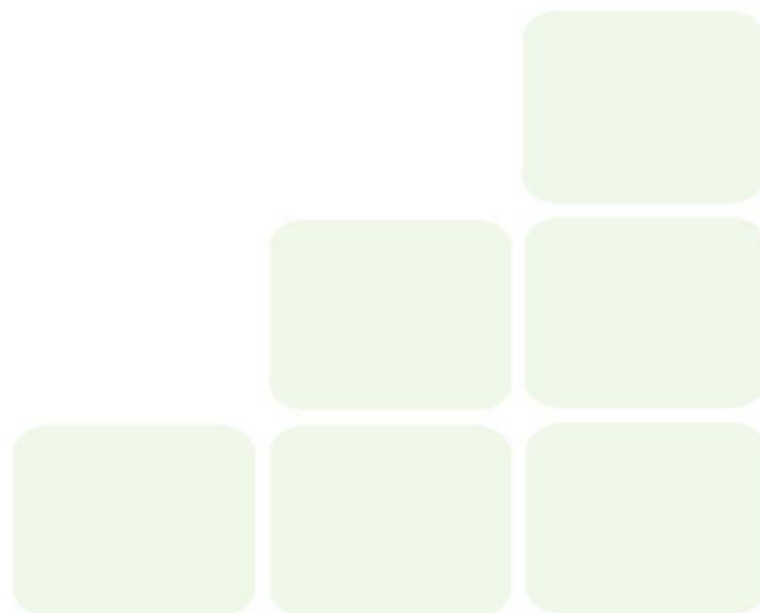
- Resultado: Será impresso na tela "JUao".

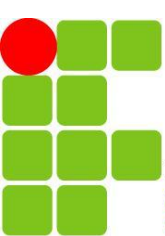




Principais Funções

- Utilizar strings da forma básica em C é bastante limitado, sendo assim, existem funções para o manipulação de strings em C. Veremos as principais:
 - gets
 - strcpy
 - strcat
 - strlen
 - strcmp





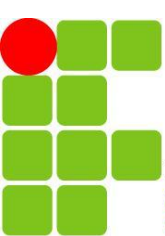
Principais Funções

- **gets**
 - A função **gets** lê uma string do teclado e pode ser definida da seguinte forma:

```
gets (<nome_da_string>) ;
```

- A função **gets** pode ser substituída pela função **scanf** na seguinte forma:

```
scanf ("%s", <nome_da_string>) ;
```

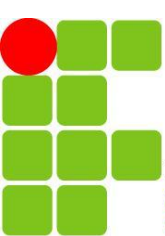


Principais Funções

- gets
 - Exemplo:

```
int main() {  
    char string[100];  
    printf ("Digite o seu nome: ");  
    // gets(string);  
    scanf ("%s", string);  
    printf ("\n\n Ola %s", string);  
    return (0);  
}
```



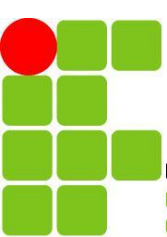


Principais Funções

- **strcpy**
 - A função `strcpy` copia a `<string_origem>` para a `<string_destino>` e pode ser definida da seguinte forma:

```
strcpy(<string_destino>, <string_origem>);
```

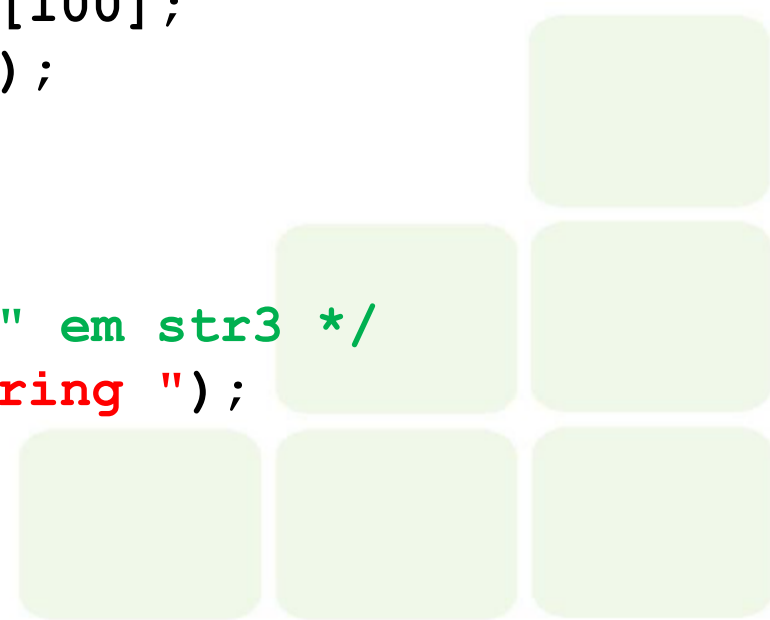
- *Obs.: Para utilizarmos a função `strcpy` devemos utilizar a biblioteca `string.h`*

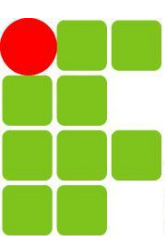


Principais Funções

- strcpy
 - Exemplo:

```
#include <stdio.h>
#include <string.h>
int main () {
    char str1[100], str2[100], str3[100];
    printf("Entre com uma string: ");
    gets(str1);
    /* Copia str1 em str2 */
    strcpy(str2, str1);
    /* Copia "Voce digitou a string" em str3 */
    strcpy(str3, "Voce digitou a string ");
    printf("\n\n%s%s", str3, str2);
    return (0);
}
```



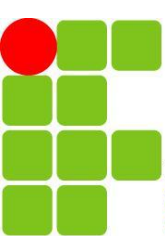


Principais Funções

- **strcat**
 - A função `strcat` copia a `<string_origem>` para o final da `<string_destino>` e pode ser definida da seguinte forma:

```
strcat(<string_destino>, <string_origem>);
```

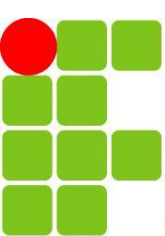
- *Obs.1: Para utilizarmos a função `strcat` devemos utilizar a biblioteca `string.h`*
- *Obs.2: Chamamos essa operação de concatenação de strings.*



Principais Funções

- strcat
 - Exemplo:

```
#include <stdio.h>
#include <string.h>
int main() {
    char str1[100], str2[100];
    printf ("Entre com uma string: ");
    gets (str1);
    strcpy (str2, "Voce digitou a string ");
    //str2 armazenará Voce digitou a string + o conteúdo de str1
    strcat (str2, str1);
    printf ("\n\n%s", str2);
    return (0);
}
```

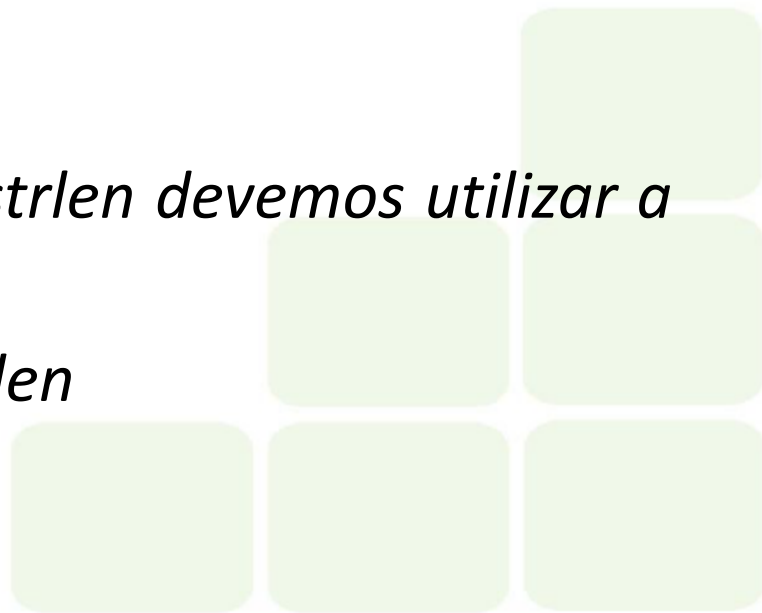


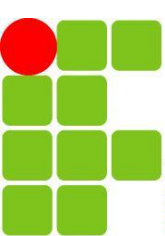
Principais Funções

- `strlen`
 - A função `strlen` retorna o comprimento da string `<string>` e pode ser definida da seguinte forma:

```
int strlen(<string>) ;
```

- *Obs.1: Para utilizarmos a função `strlen` devemos utilizar a biblioteca `string.h`*
- *Obs.2: O `'\0'` não é contado em `strlen`*

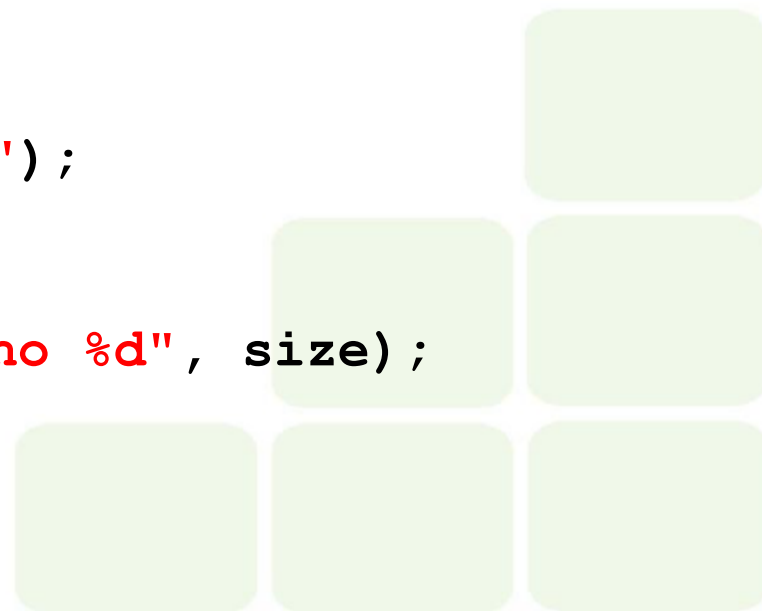


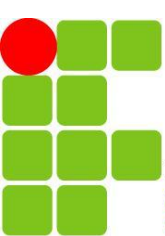


Principais Funções

- strlen
 - Exemplo:

```
#include <stdio.h>
#include <string.h>
int main(){
    int size;
    char str[100];
    printf("Entre com uma string: ");
    gets(str);
    size = strlen(str);
    printf("\n\nA string tem tamanho %d", size);
    return(0);
}
```





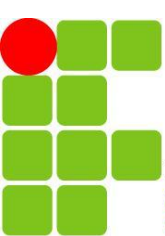
Principais Funções

- strcmp

- A função strcmp compara a <string1> com a <string2>, se forem idênticas, a função retorna 0, senão, se a string1 for menor retorna -1, se for maior, 1. Podendo ser definida da seguinte forma:

```
int strcmp(<string1>, <string2>);
```

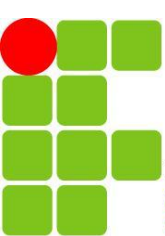
- *Obs.1: Para utilizarmos a função strcmp devemos utilizar a biblioteca string.h*



Principais Funções

- strcmp
 - Exemplo:

```
#include <stdio.h>
#include <string.h>
int main() {
    char str1[100], str2[100];
    printf("Entre com uma string: ");
    gets(str1);
    printf("\n\nEntre com outra string: ");
    gets(str2);
    if(strcmp(str1, str2))
        printf("\n\nAs duas strings são diferentes.");
    else
        printf("\n\nAs duas strings são iguais.");
    return(0);
}
```

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.

