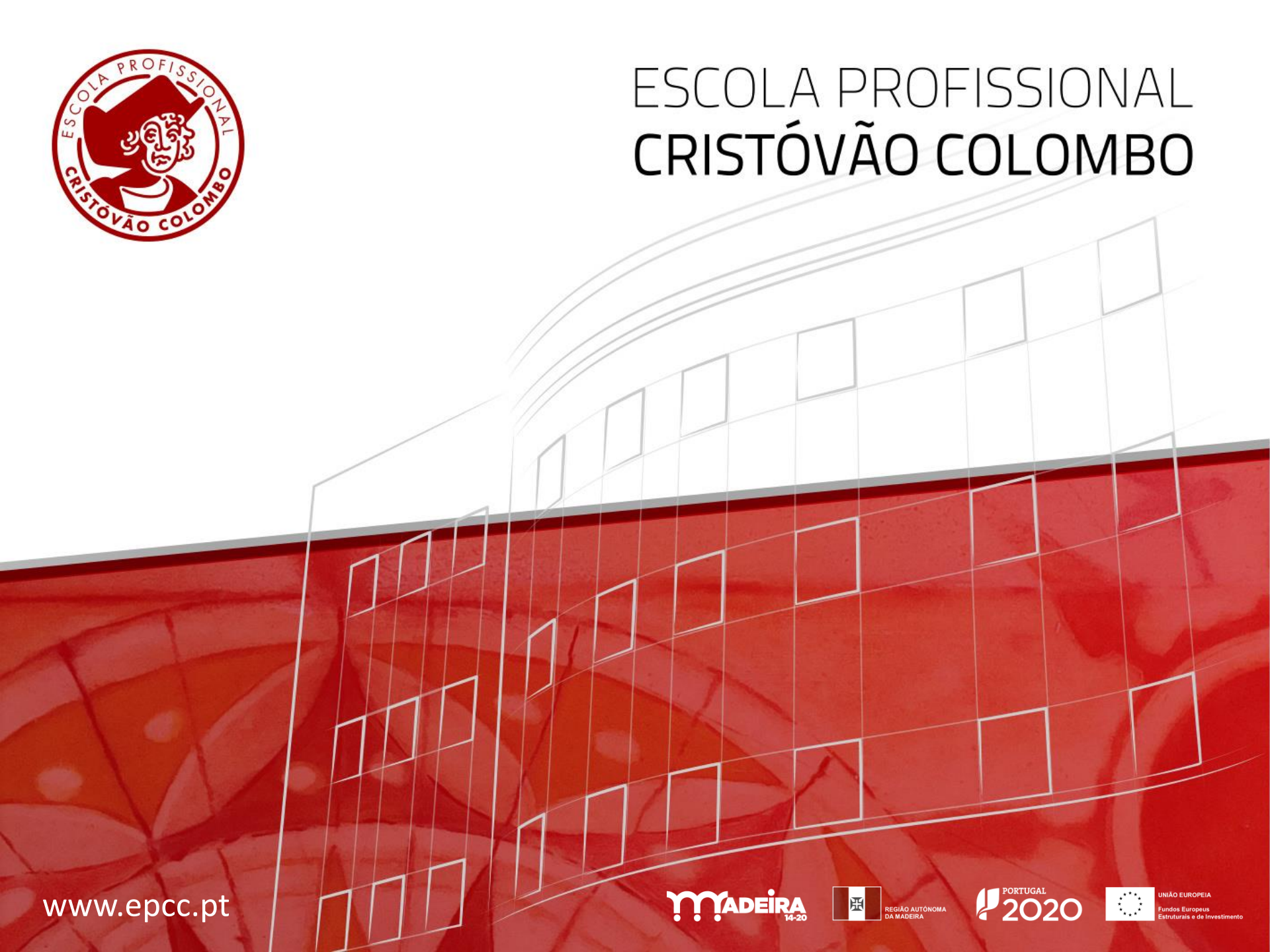
ESCOLA PROFISSIONAL
CRISTÓVÃO COLOMBO



www.epcc.pt

MADEIRA 14-20 · REGIÃO AUTÓNOMA DA MADEIRA · PORTUGAL 2020 · UNIÃO EUROPEIA · Fundos Europeus Estruturais e de Investimento

Sistemas Operativos – Processamento Computacional

String

Autor da apresentação

09/09/2014

Strings

As *strings* mais não são que *array*'s de caracteres. Aquando da declaração de uma variável do tipo *string* especifica-se a sua dimensão máxima, no entanto, quando se pretende usar a variável para armazenar uma *string* com um número de caracteres inferior ao máximo, usa-se o carácter nulo '\0' para indicar o fim da *string*.

Strings

Se pretendermos declarar uma *string* para armazenar um conjunto de 10 caracteres devemos escrever:

Char str[11];

Visto, o ultimo caractere sem o ‘\0’

Strings

Se pretendermos declarar uma *string* para armazenar um conjunto de 10 caracteres devemos escrever:

Char str[11];

Visto, o ultimo caractere sem o ‘\0’

Strings - Exemplo

```
int main(){
// Criação do array de caracteres
char nome[80];
printf("Digite: ");
/* Usamos para ler
scanf()- até digitar espaço ou enter
gets()- termina na digitação do enter
*/
// não necessita endereco visto ser um array
//scanf("%s",nome);
gets(nome);// termina apenas na digitação do enter
printf("Digite:%s ",nome);
```

Strings - Exemplo

```
/*iniciação por array generico, \0 indica termino da palavra*/
char nomes1[20] = {'A','Z','D','h','\0'};
printf("Digite:%s ",nomes1);
/* com aspas dupla, preenche automaticametne \o*/
char nomes2[20] = "Sergio araujo";
printf("Digite:%s ",nomes2);
/* sendo uma string podemos aceder a cada posição do array*/
nomes2[2] = 'D';
printf("Digite:%s ",nomes2);

/* \0 a var nomes2 tem 20 carateres de espaço
 caso seja atribuido o '\0' numa posição especifica
  o interpretador ignora todo o restante texto
*/
nomes2[2] = '\0';
printf("Digite:%s ",nomes2);
```

Strings – Biblioteca String.h

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* String :
array de carateres, permite armazenar
uma sequencia de carateres*/
int main(){
char n1[80] = "Sérgio";
char n2[80];
/* *Sendo as var um array não é posivel
 fazer atribuição simples do seu conteudo */

 n1 = n2; // dá erro
```

Strings – Biblioteca String.h

```
/* Possivel solução */
int i;
while(n1[i]!= '\0'){
    n2[i]= n1[i];
    i++;
}
n2[i] = '\0';
printf("Digite:%s ",n1);
printf("Digite:%s ",n2);
/* verificamos que é moroso e trabalhoso
fazer uma copia por exemplo */
```

Strings – Biblioteca String.h

```c
#include <stdlib.h>
#include <string.h>

/* String : array de carateres , permite armazenar
uma sequencia de carateres*/

int main(){
/* strlen - tamanho da string, não contabiliza o \0*/
// captura o seu conteudo, não o tamanho da criação[80]
char nStr[14] = "Sérgio Araujo";
int tamanho_string = strlen(nStr);
 printf("tam:%d ",tamanho_string);

/* strcpy(para,de) - copia, alterando caso não esteja vazia*/
char n3[] = "ola";
strcpy(n3,nStr);
printf("Nova: %s ",n3);
}
```

Strings – Biblioteca String.h

```c
/* strcat(para,de). concatena array de carateres*/
char n4[40] = "Eu ";
char n5[40] = "Sérgio";

strcat(n4,n5);
printf("Nova: %s ",n4);

/* strcmp(para,de). verifica se são iguais devolve 0,
   diferentes devolve valores diferente de 0*/
char n6[40] = "Eu";
char n7[40] = "Sérgio";
if(strcmp(n6,n7) == 0)
  printf("Iguais");
else
  printf("Dif");
```