

Introdução à Computação – Eng. Civil

Módulo 6

Strings

Angelo Martins – Dep. Eng. Informática - ISEP

- Conceito e aplicação
  - O que são e para que servem as strings. As rotinas de manipulação de strings incluídas na biblioteca do Visual Basic.
- Rotinas com strings
  - Utilização de strings como parâmetros de funções e subrotinas.
- Exemplos de aplicação
  - Manipulação de caracteres e palavras.

- O que são strings?
  - Uma grande parte das linguagens de programação permitem manipular caracteres e/ou conjuntos de caracteres.
  - Uma string, que em português se refere normalmente por “cadeia de caracteres”, será uma entidade que representa um conjunto de caracteres.
  - Em Visual Basic, uma string é representada internamente como um vector de números inteiros, em que cada número representa um carácter.

- Sintaxe
  - String

```
Dim <nome> As String
```
  - Vector de Strings

```
Dim <nome>( <nº de elementos> ) As String
```
  - Exemplo

```
Dim st1 As String, st As String
...
st1 = txtMyString.Text
st = "Olá" + " " & st1
```

## Funções do VB - Trim

- Trim, LTrim e RTrim

- Retornam uma string idêntica à original, mas com espaços eliminados no início e/ou fim da string.

```
Dim st1 As String, st2 As String
```

```
st1 = " Olá Mundo! "
```

```
st2 = Trim(st1) ← st2 = "Olá Mundo!" – Foram eliminados os  
espaços no início e no fim da string.
```

```
st2 = LTrim(st1) ← st2 = "Olá Mundo! " – Foram eliminados os  
espaços no início da string (left).
```

```
st2 = RTrim(st1) ← st2 = " Olá Mundo!" – Foram eliminados os  
espaços no fim da string (right).
```

- Os espaços que esteja entre outros caracteres da string não são afectados por estas rotinas.

## Funções do VB - Len

- Len

- Retorna o número de caracteres da string.

```
Dim st1 As String, k As Integer
```

```
st1 = " Olá Mundo! "
```

```
k = Len(st1) ← k = 16 – a string tem 16 caracteres, incluindo espaços.
```

```
k = Len(Trim(st1)) ← k = 12 – Foram eliminados os espaços no início e no  
fim da string passada à função Len().
```

- Exemplo de aplicação: Verificar se uma string está vazia

```
If Len(st1) = 0 then
```

```
...
```

```
Endif
```

## Funções do VB – Right, Left

### • Left, Right

Left(<original> as String, <nº\_caracteres> as Integer) as String

- Copiam o número de caracteres indicado da string original.

```
Dim st1 As String, st2 As String, k As Integer
```

```
st1 = "Olá Mundo!"
```

```
k = 2
```

```
st2 = Left(st1, k)
```

```
st2 = Right(st1, k)
```

st1 = "Ol" – os dois primeiros caracteres da string, a partir da esquerda.

st2 = "o!" – os dois primeiros caracteres da string, a partir da esquerda.

- Exemplo de aplicação: eliminar o primeiro caracter de uma string.

```
st1 = Right(st1, Len(st1)-1)
```

Copia todos os elementos da string, a partir da direita, com a exceção do primeiro..

## Funções do VB – Mid

### • Mid

Mid(<original> as String, <pos\_init> as Integer [, \_  
<nº\_caracteres> as Integer]) as String

- Copia o número de caracteres indicado a partir da posição inicial passada como parâmetro. O número de caracteres é opcional. Se não for indicado, copia tudo até ao fim da string.

```
Dim st1 As String, st2 as String
```

```
st1 = "Olá Mundo!"
```

```
st2 = Mid(st1, 2, 3)
```

```
st2 = Mid(st1, 6, 8)
```

st2 = "lá " – copia 3 caracteres a partir da segunda posição.

st2 = "undo!" – copia os caracteres a partir da sexta posição. Como o número de caracteres ultrapassa o limite da string, apenas são copiados os caracteres existentes.

### • Instr

```
Instr([<pos_init> as Integer,] <original> as String, _  
      <padrão> as String) as Integer
```

- Procura uma string dentro de outra. A posição inicial é opcional.

```
Dim st1 As String, k As Integer  
st1 = "Olá Mundo!"  
k = Instr(st1, " ") ← k = 4 – posição do primeiro espaço.  
  
k = Instr(5, st1, " ")  
← k <= 0 – não há nenhum espaço a partir da posição 5.
```

### • UCase, LCase

```
UCase(<original> as String) as String  
LCase(<original> as String) as String
```

- Retorna uma string idêntica à original, mas com os caracteres convertidos para maiúsculas (UCase) ou minúsculas (LCase). Os número e caracteres de pontuação não são afectados.

```
Dim st1 As String, st2 as String  
st1 = "Ola 1,2,3!"  
st2 = UCase(st1) ← st2 = "OLA 1,2,3!"  
  
st2 = LCase(st1) ← st2 = "ola 1,2,3!"
```

## Algoritmos – Caracteres

- Separação de uma string em caracteres
  - Pretende-se manipular cada um dos caracteres de uma string. O VB não tem um tipo de dados caracter, pelo que cada caracter será guardado numa string.
  - Vamos utilizar dois métodos:
    - Método 1
      - Utilizam-se as funções Left() e Right()
      - Ciclo Do...Loop
    - Método 2
      - Utilizando a função Mid()
      - Ciclo For ... Next ou Do ... Loop

## Separação de caracteres

- Método 1 – destrói string original

```
Dim st As String, c As String
Do While st <> ""
    c = Left(st, 1)
    ' Aqui pode-se fazer algo com os caracteres
    st = Right(st, Len(st) - 1)
Loop
```

Enquanto a string tiver caracteres.

Obtém o primeiro caracter da string.

Elimina o primeiro caracter da string, copiando todos os caracteres menos o primeiro.

- Método 2 – preserva string original

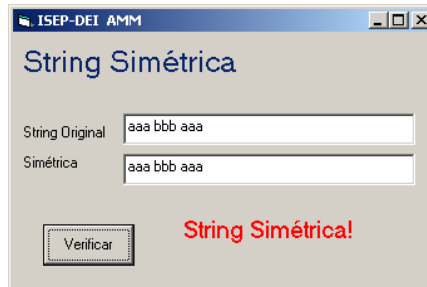
```
Dim st As String, c As String, v As Integer
For v = 1 To Len(st)
    c = Mid(st, v, 1)
    ' Aqui pode-se fazer algo com os caracteres
Next
```

Para cada caracter da string.

Obtém o caracter na posição v string.

## Exemplo

- Elabore uma função que retorne a simétrica de uma string passada como parâmetro. Utilizando a função, elabore um programa que verifique se uma string é simétrica.



## Exemplo - Código

```
Private Sub cmdVer_Click()  
    Dim sto As String, sti As String  
  
    lblRes.Visible = False  
    sto = txtSt.Text  
    sti = invString(sto)  
    txtSim.Text = sti  
    If sti = sto Then  
        lblRes.Caption = "String Simétrica!"  
    Else  
        lblRes.Caption = "String Assimétrica!"  
    End If  
    lblRes.Visible = True  
End Sub  
  
Function invString(ByVal st As String) As String  
    Dim c As String, iv As String  
    Do While st <> ""  
        c = Left(st, 1)  
        st = Right(st, Len(st) - 1)  
        iv = c + iv  
    Loop  
    invString = iv  
End Function
```

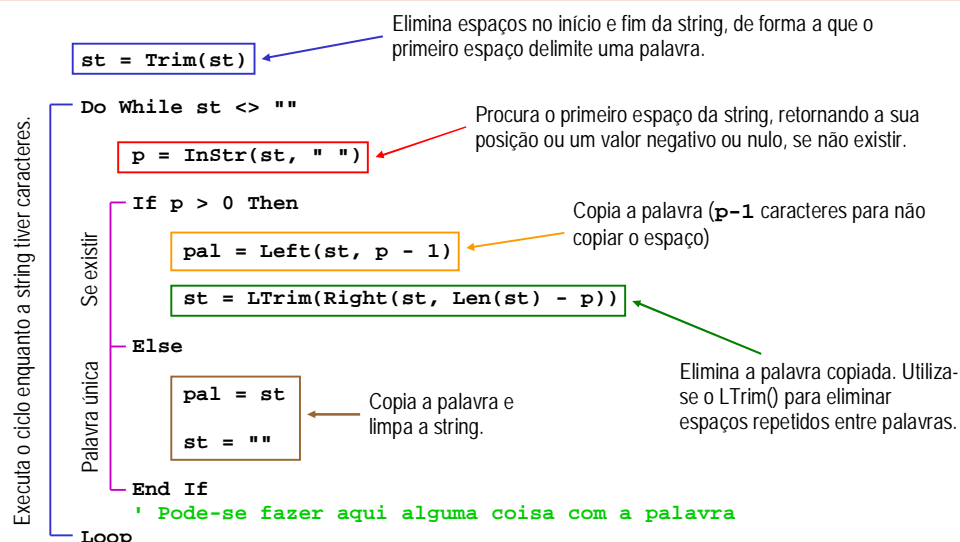
← Não se pode utilizar a função Val() ao ler strings! Lê-se directamente de uma TextBox ou InputBox.

← Reconstrói a string colocando os caracteres pela ordem inversa, isto é, o último é colocado no início.

## Algoritmos – Palavras

- Separação de uma string em palavras
  - Pretende-se manipular palavras. Entende-se como uma palavra:
    - Conjunto de caracteres entre espaços, entre o início da string e um espaço, entre um espaço e o fim da string, ou todos os caracteres de uma string, se esta não tiver espaços.
  - Vamos utilizar um método destrutivo, em que se procura o primeiro espaço e se assume que entre o início da string e o espaço se encontra uma só palavra.
    - Utilizam-se as funções Left(), Right(), Instr() e Trim()
    - Ciclo Do...Loop

## Separação de palavras





## Execução passo a passo

- Teste com a string: " Olá mundo! "

Iteração	st	x <> ""	p	p>0	pal
	" Olá mundo! "				" "
	"Olá mundo!"				
1		True			
			4		
				True	
					"Olá"
	"mundo!"				
2		True			
			0		
				False	
					"mundo!"
	" "				
3		False			

## Exemplo

- Elabore uma função que recebe uma string como parâmetro e retorna o número de palavras que a string contém.

```

Public Function nPal(ByVal st As String) As Integer
    Dim n As Integer, p As Integer

    st = Trim(st)
    If st <> "" Then
        Do
            p = InStr(1, st, " ")
            If p > 0 Then
                st = LTrim(Right(st, Len(st) - p))
                End If
                n = n + 1
            Loop While p > 0
        End If
        nPal = n
    End Function

```

Conta o número de palavras encontradas

## Outro Exemplo

- Elabore uma função que recebe uma string e um número n como parâmetros e retorna uma string com as palavras da primeira com menos de n caracteres.

```
Function truncaPal(ByVal st As String, ByVal n As Integer) As String
    Dim p As Integer, pal As String, stout As String
    st = Trim(st)
    Do While st <> ""
        p = InStr(st, " ")
        If p > 0 Then
            pal = Left(st, p - 1)
            st = LTrim(Right(st, Len(st) - p))
        Else
            pal = st
            st = ""
        End If
        If Len(pal) < n Then
            stout = stout + " " + pal
        End If
    Loop
    truncaPal = Trim(stout)
End Function
```

Se a palavra contém menos de n caracteres, então é acrescentada.