

## Programas Básicos

## Conteúdo

---

- Exemplo de um programa
  - Apresentação sumária da construção de um programa em VB.
- Objectos e o mecanismo de eventos
  - Apresentação dos conceitos básicos de programação em ambiente MS Windows.
- Tipos de objectos e dados
  - Apresentação do conjunto limitado de objectos que são utilizados nos primeiros programas e dos principais tipos de dados.

## Passos na construção do programa

- A construção de um programa obedece a um conjunto de passos:
  - Definir objectivos e funcionalidades do programa
  - Desenhar a interface do programa com o utilizador
  - Ajustar (configurar) as propriedades dos objectos
  - Escrever o código
  - Testar

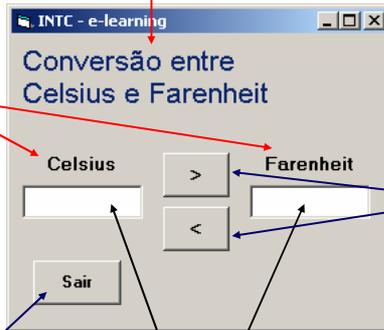
## Exemplo de programa

- Pretende-se um programa que:
  - efectue a conversão ente graus Celsius e Fahrenheit
  - as formulas de conversão são:
    - $^{\circ}\text{C} = (^{\circ}\text{F} - 32) * 5/9$
    - $^{\circ}\text{F} = ^{\circ}\text{C} * 9/5 + 32$
- Decidiu-se que:
  - devem ser visíveis os valores nas duas escalas
  - a interface deve ser simples e auto-explicativa.

## Desenho da Interface

Um *Label* a indicar qual a função do programa.

Dois *Label*s para identificar a função de cada uma das *TextBox*.



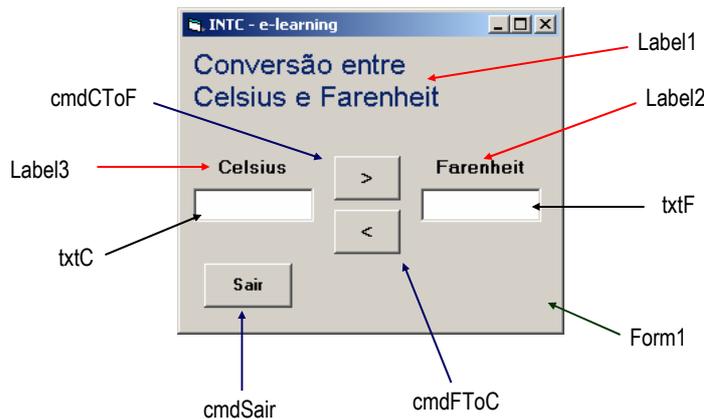
Dois *CommandButton*, um para cada sentido de conversão. Quando o utilizador carrega num destes botões é efectuada a conversão correspondente.

Um *CommandButton* para terminar o programa.

Duas *TextBox* para o utilizador introduzir valores da temperatura.

## Configurar a Interface

A *form* do programa já tem o aspecto final, mas falta ainda atribuir nomes aos objectos.



## Escrever o Código

O código do programa é o seguinte:

`Option Explicit` ← Instrução para o interpretador do VB

```
Private Sub cmdCToF_Click()  
    Dim c As Single  
  
    c = Val(txtC.Text)  
    txtF.Text = 9 / 5 * c + 32  
End Sub
```

← Código do botão cmdCToF

```
Private Sub cmdFToC_Click()  
    Dim f As Single  
  
    f = Val(txtF.Text)  
    txtC.Text = (f - 32) * 5 / 9  
End Sub
```

← Código do botão cmdFToC

```
Private Sub cmdSair_Click()  
    End  
End Sub
```

← Código do botão cmdSair

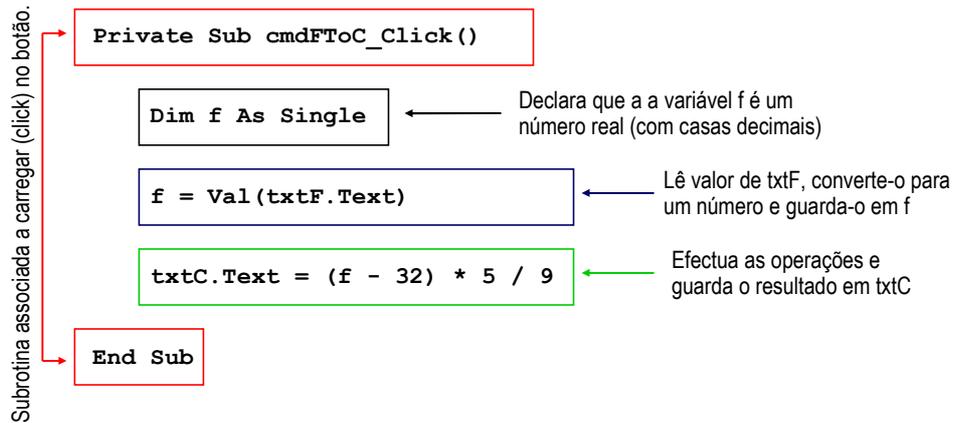
## Testar

Uma vez terminado o programa, é preciso efectuar testes. Para tal, introduzem-se valores conhecidos e verifica-se se o resultado está mesmo correcto. Não adianta introduzir valores à sorte!



## Explicação do Código

Vejamos qual o significado do código do botão cmdFToC:



## Objectos e Eventos

- A interface de um programa em VB é composta por objectos. No exemplo utilizamos 4 tipos:
  - Form
  - Label
  - TextBox
  - CommandButton
- Quando o utilizador interage com o programa (introduz valores, arrasta a janela, carrega num objecto, etc.) são desencadeados eventos. No exemplo só reagimos a 3 eventos:
  - Click() em cada um dos botões: cmdSair, cmdFToC e cmdCToF

## Objectos

---

- . Form
  - usualmente traduzido como *formulário*
  - contentor rectangular (janela) onde são colocados os outros objectos
- . Label
  - traduzido como *etiqueta*
  - pode conter apenas texto (caracteres e algarismos), sendo utilizado para dar indicações ao utilizador ou para apresentar resultados.
  - o utilizador não consegue alterar directamente o conteúdo do label.

## Objectos II

---

- . TextBox
  - usualmente traduzido como *caixa de texto*
  - pode conter apenas texto (caracteres e algarismos), sendo utilizado para o utilizador introduzir dados ou para apresentar resultados.
- . CommandButton
  - traduzido como *botão de comando*
  - pode conter texto (caracteres e algarismos) ou uma figura, sendo utilizado para desencadear acções.
  - o utilizador não consegue alterar o texto ou figura apresentada no botão.

## Tipos de Dados

- A representação de informação em VB está condicionada a um conjunto de tipos de dados básicos predefinidos.
  - Podem ainda ser criados tipos de dados compostos por vários elementos dos tipos básicos
  - Podem ser definidos objectos, que expandem as facilidades de manipulação dos tipos de dados compostos
- Nesta disciplina só vamos utilizar quatro tipos de dados básicos:
  - Números Inteiros
  - Números Reais
  - Caracteres
  - Valores lógicos

## Números Inteiros

- Os números inteiros permitem representar valores numéricos inteiros positivos e negativos.
- Há vários tipos, mas só serão utilizados dois:
  - Integer
    - Gama de valores: -32768 a 32767  $\Rightarrow [-2^{15}, 2^{15}-1]$
    - Declaração: `Dim myvar as Integer`
  - Long
    - Gama de valores: -2147483648 a 2147483647  $\Rightarrow [-2^{31}, 2^{31}-1]$
    - Declaração: `Dim myvar as Long`
- As operações suportadas são:
  - Operações aritméticas: +, -, /, \*
  - Quociente e resto: \, mod

## Números Reais

- . Os números reais permitem representar valores numéricos positivos e negativos com casas decimais.
- . Utilizam a forma mantissa + expoente, e há dois tipos:
  - Single (precisão simples  $\Rightarrow$  IEEE 32-bit)
    - . Gama de valores:  $[-3.4028235E+38, -1.401298E-45] \cup 0 \cup [1.401298E-45, 3.4028235E+38]$
    - . Declaração: `Dim myvar as Single`
  - Double (precisão dupla  $\Rightarrow$  IEEE 64-bit)
    - . Gama de valores:  
 $[-1.79769313486231570E+308, -4.94065645841246544E-324] \cup 0 \cup [4.94065645841246544E-324, 1.79769313486231570E+308]$
    - . Declaração: `Dim myvar as Double`
- . São suportadas todas as operações com a exceção do quociente e resto.

## Caracteres

- . Permitem representar caracteres do alfabeto, algarismos, operadores e outros símbolos gráficos.
  - Exemplos
    - . `"olá!"`
    - . `"O resultado foi 3-1"`
  - Declaração
    - . `Dim st as String`
  - Operações
    - . + e &, ambas fazem a concatenação de strings.:  
`"eu" + "fui" -> "eufui"`  
`"marquei" & " " & 3 & " golos" -> "marquei 3 golos"`
    - . Há um conjunto alargado de funções que permitem manipular strings. Um exemplo:
      - `val("25") -> 25` (retorna o valor numérico representado na string)

## Valores Lógicos

- Permitem representar valores lógicos (verdadeiro e falso). Em VB estas estas variáveis podem tomar os valores True ou False.
  - Exemplos:

```
Dim res as Boolean, m as Boolean
res = True
m = (4 > 5)      (m tem valor False)
```
  - Operadores
    - And, Or, Not, Xor
    - >, <, >=, <=, <>
- O tipo de dados boolean será essencialmente utilizado em ciclos e funções/subrotinas.

## Constantes

- Uma constante consiste num nome que é dado a um valor numérico, a um valor booleano ou a uma string, e que pode ser usado num programa em substituição desse valor. Este valor não pode ser alterado durante a execução do programa.
- Se for definida uma constante com um valor, bastará alterar uma só vez essa definição para que tal mudança automaticamente se repercute em todas as ocorrências dessa constante no programa.
  - Definição:

```
Const <nome> As <tipo> = <expressão>
```
  - Exemplos:

```
Const numElem As Integer = 25
Const meuEmail As String = "amm@isep.ipp.pt"
```

## Variáveis

- Uma variável é um contendor em que informação de um dado tipo pode ser guardada de modo a ser manipulada num programa (<http://en.wikipedia.org/wiki/Variable>)
- Cada variável é caracterizada por:
  - um nome
    - identificador da variável, tem que começar por uma letra e não pode conter espaços ou outros caracteres de pontuação:
      - Nomes válidos: `abc`, `a12`, `ola_mundo`
      - Nomes inválidos: `1qa`, `ola-mundo`, `num!`, `ab cd` (são duas variáveis)
  - um tipo de dados
    - Pode ser qualquer tipo de dados predefinido ou definido pelo programador
  - âmbito (visibilidade)
    - Uma variável só está acessível dentro do bloco em que foi definida.

## Variáveis II

- Uma variável pode ser definida explicitamente ou o VB cria a variável com o tipo de dados que ache apropriado na primeira vez que esta for utilizada.
  - Definição Explícita

```
Dim <nome> As <tipo>
Dim a as Integer, b as Double
```
  - Definição Implícita

```
Dim <nome>
Dim a, b
a = 2          \ variável inteira
b = 3.5       \      "      real
c = True      \      "      booleana
```
  - Atribuição

```
<nome_da_variável> = <expressão>
```

O programador pode introduzir comentários no programa, que são ignorados pelo VB. Um comentário começa com um apóstrofe e termina no fim da linha.

## Conclusão

---

- . Introduzimos os conceitos de:
  - Programa
  - Objecto
  - Tipo de dados
  - Constante
  - Variável
  
- . Estes conceitos vão ser explorados nas aulas práticas, resolvendo exercícios.