

Programação RETR0

QB64

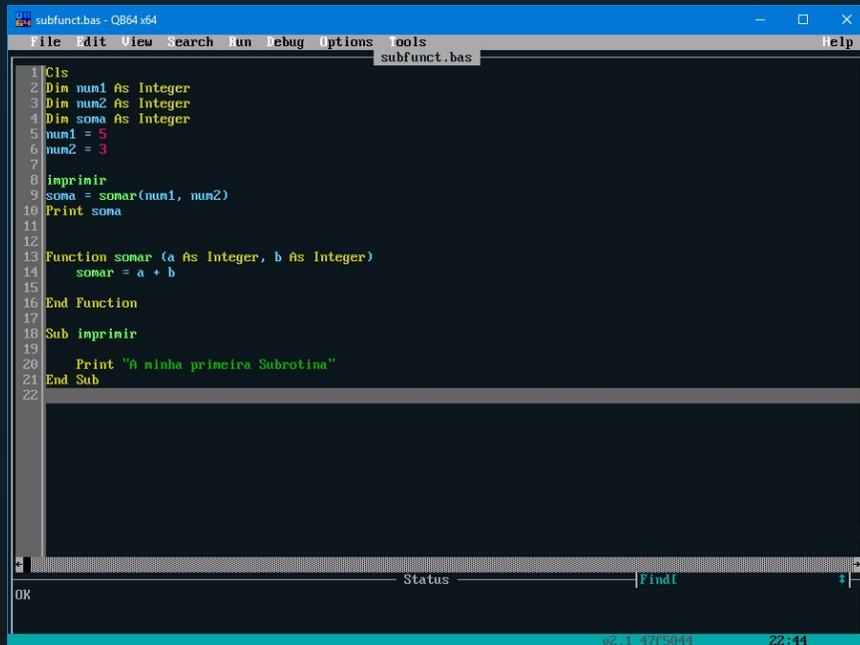
PSINF

QB64

- ❑ A partir deste momento passamos a utilizar o QB64
 - ❑ Versão **moderna** do **Qbasic**
 - ❑ **Retrocompatível** (corre os programas realizados até aqui)
 - ❑ Mais **user friendly**
 - ❑ Permite **extensões**
 - ❑ **Motor** de jogo sem recurso a **POO**
 - ❑ **Formulários** (com **POO**)
 - ❑ Mais **ferramentas** de **desenho**
 - ❑ **Maiores resoluções** de ecrã e **mais cores**

QB64 - Interface

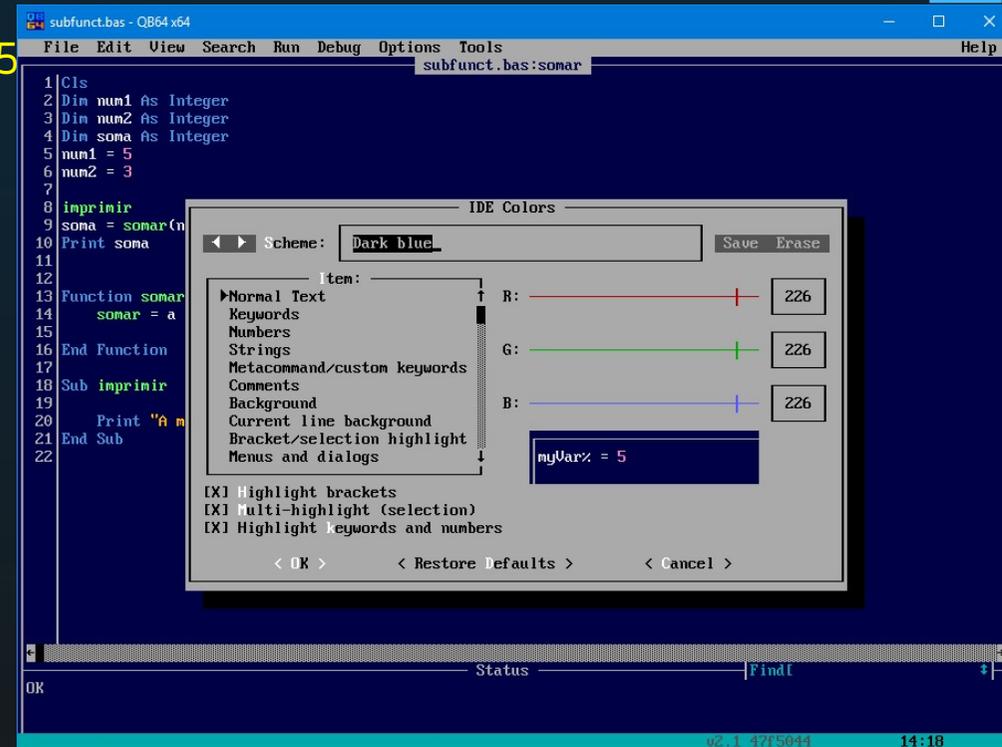
- ❑ A interface mais moderna e com mais cores (a lembrar Python)
- ❑ Permite um esquema de cores personalizável mais completo
- ❑ Resto das funcionalidades iguais ao Qbasic 4.5



```
1 Cls
2 Dim num1 As Integer
3 Dim num2 As Integer
4 Dim soma As Integer
5 num1 = 5
6 num2 = 3
7
8 imprimir
9 soma = somar(num1, num2)
10 Print soma
11
12
13 Function somar (a As Integer, b As Integer)
14     somar = a + b
15 End Function
16
17 Sub imprimir
18     Print "A minha primeira Subrotina"
19 End Sub
```

OK

v2.1 42f5044 22:44



```
1 Cls
2 Dim num1 As Integer
3 Dim num2 As Integer
4 Dim soma As Integer
5 num1 = 5
6 num2 = 3
7
8 imprimir
9 soma = somar(num1, num2)
10 Print soma
11
12
13 Function somar (a As Integer, b As Integer)
14     somar = a + b
15 End Function
16
17 Sub imprimir
18     Print "A minha primeira Subrotina"
19 End Sub
```

IDE Colors

Scheme: Dark blue

Item: Normal Text

R: 226

G: 226

B: 226

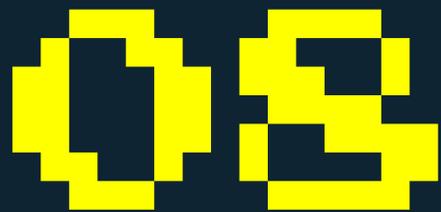
myVar% = 5

Highlight brackets
 Multi-highlight (selection)
 Highlight keywords and numbers

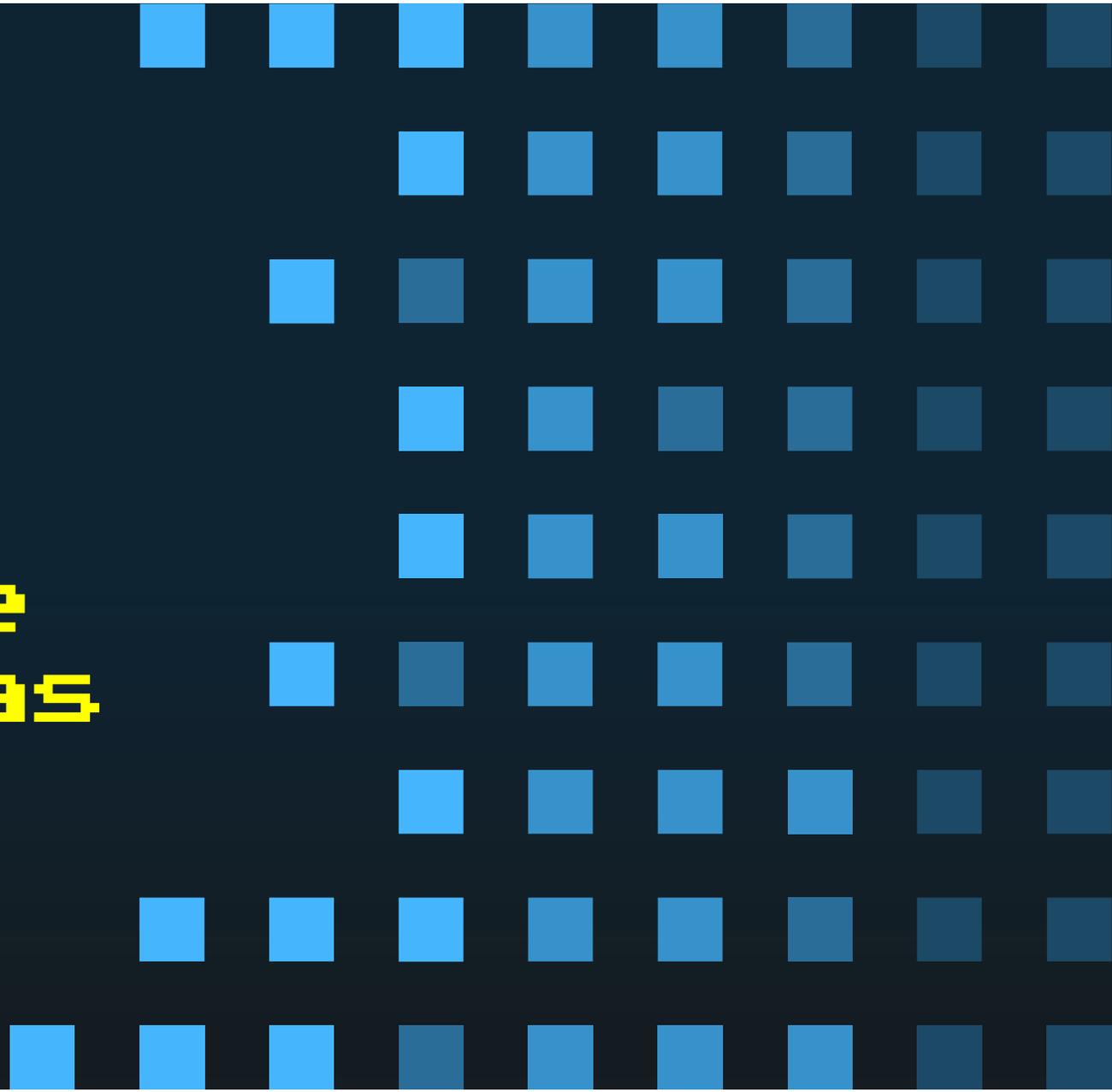
OK < Restore Defaults > < Cancel >

OK

v2.1 42f5044 14:18



Funções e Subrotinas



Funções e Subrotinas

- ❑ As **funções** e as **subrotinas** podem ser muito **úteis** para **evitar repetição** de **código**.
- ❑ **Imaginem** um **jogo** com **5 inimigos iguais**. Todos com a mesma quantidade de vida, a mesma forma de morrer e o mesmo movimento no cenário de jogo.
- ❑ Para **definir** todas estas **ações** teríamos de **repetir código** para cada um deles.
- ❑ Com recurso a funções basta **chamar** no **código** a **função** respetiva e a mesma pode ser chamada quantas vezes quisermos **sem repetir código**.
- ❑ Imaginem um bombo de festa no meio de uma orquestra. Quando o maestro dá indicação ele toca.

Funções Vs Subrotinas

- ❑ Para além da sintaxe a **diferença** é que as **funções** retornam **valores** para o local onde foram **chamadas** e as **subrotinas não**
- ❑ **Retornar valores???** Vejamos as diferenças primeiro de sintaxe:

Função	Subrotina
<pre>FUNCTION NAME <PARAMETERLIST> < STATEMENTS > NAME = <EXPRESION> END FUNCTION</pre>	<pre>SUB NAME <PARAMETERLIST> STATEMENT BLOCK END SUB</pre>

Subrotina

- Exemplo (sem parâmetros)

```
subrotina.bas - QB64 x64
File Edit View Search Run Debug Options Tools Help
subrotina.bas*
1 Cls
2
3 imprimir
4
5 Sub imprimir
6   Print "A minha primeira Subrotina"
7 End Sub
8
9
10
```

```
Untitled
A minha primeira Subrotina
Press any key to continue
```

- 1º chamada depois declarada
(noulras linguagens costuma ser ao contrário)

Subrotina

- Exemplo com parâmetros

The image shows a screenshot of a Visual Basic IDE. The main window, titled "quadrado.comsub.bas - QB64 x64", displays the following code:

```
1 Cls
2 Dim num As Integer
3 Input "Digite o número que quer ver o seu
4 quadrado (num)
5 Sub quadrado (numero As Integer)
6     Print "O quadrado do número "; numero
7 End Sub
```

An "Untitled" console window in the foreground shows the program's execution output:

```
Digite o número que quer ver o seu quadrado: 5
O quadrado do número 5 é igual a 25
```

At the bottom of the console window, the text "Press any key to continue" is visible.

Funções

- Exemplo sem parâmetros

```
function1.bas - QB64 x64
File Edit View Search Run Debug Options Tools Help
function1.bas
1 Cls
2
3
4 Print "O produto de 3*2="; valor
5
6
7 Function valor ()
8     valor = 3 * 2
9 End Function
10
```

```
Untitled
O produto de 3*2= 6
Press any key to continue
```

Funções

- Exemplo com parâmetros

```
function2.bas - QB64 x64
File Edit View Search Run Debug Options Tools Help
function2.bas*
1 Cls
2 Dim num1 As Integer
3 Dim num2 As Integer
4 Dim soma As Integer
5 num1 = 5
6 num2 = 3
7
8 soma = somar(num1, num2)
9 Print soma
10
11 Function somar (a As Integer, b As Integer)
12     somar = a + b
13 End Function
```

Retorna valor aqui

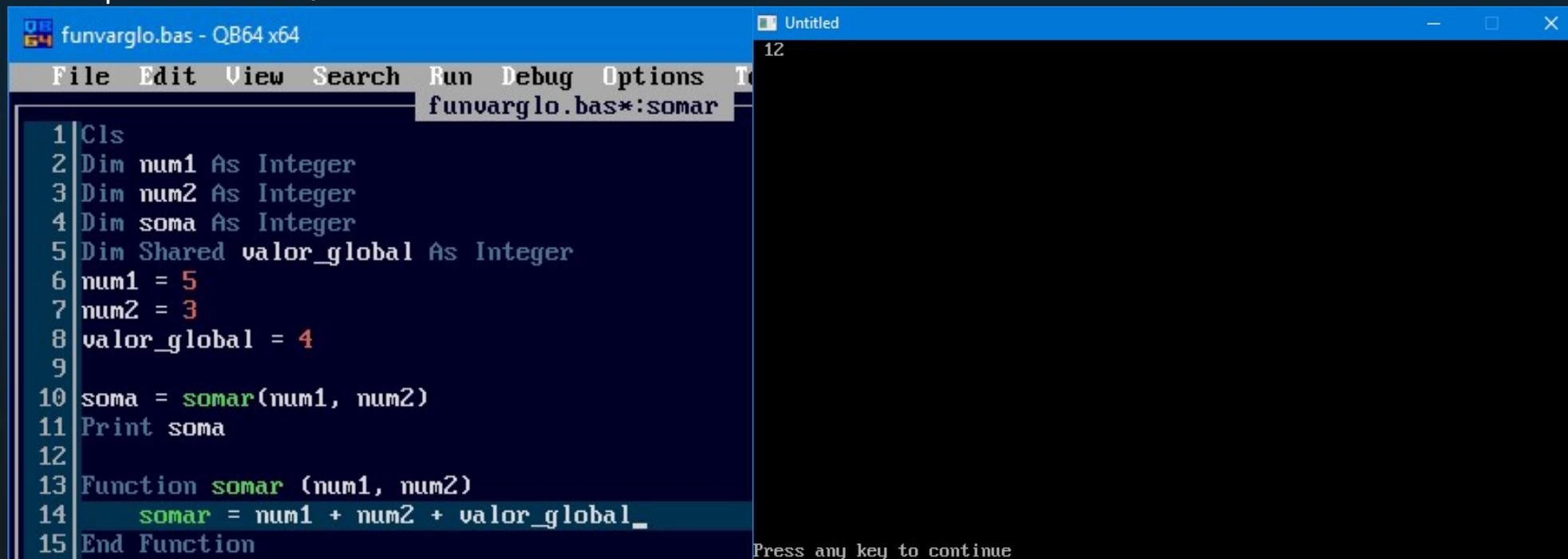
```
Untitled
8
Press any key to continue
```

Variáveis globais e locais

- ❑ As variáveis definidas até agora foram sempre locais
- ❑ As variáveis locais só existem no local onde são definidas
- ❑ Se definidas dentro de uma subrotina ou função apenas existem aí
- ❑ Se definidas fora de uma subrotina ou função apenas existem aí
- ❑ Por outro lado as variáveis globais têm de ser definidas no início do programa e vão existir em qualquer lado, podendo ser manipulados os seus valores dentro ou fora de funções e subrotinas
- ❑ Mas como declarar variáveis globais?

Variáveis globais

- Sintaxe
 - DIM SHARED nome_variável as tipo_variável
- Exemplo com funções:



```
QB64 funvarglo.bas - QB64 x64
File Edit View Search Run Debug Options T
funvarglo.bas*:somar
1 Cls
2 Dim num1 As Integer
3 Dim num2 As Integer
4 Dim soma As Integer
5 Dim Shared valor_global As Integer
6 num1 = 5
7 num2 = 3
8 valor_global = 4
9
10 soma = somar(num1, num2)
11 Print soma
12
13 Function somar (num1, num2)
14     somar = num1 + num2 + valor_global_
15 End Function
```

Untitled
12
Press any key to continue

Para hoje

Realizar os guiões no Website.

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik** and illustrations by **Stories**