

# EDUSCRATCH

PROGRAMANDO COM O



## LISTAS E DIÁLOGOS

Utilização de variáveis-lista em diálogos

(Out-2010)

## VARIÁVEIS – 2 (convém ler primeiro o documento “Diálogos.docx”)

Uma variável “Lista” é um conjunto de variáveis, todas com o mesmo nome, mas identificáveis pelo seu número de ordem, (índice), dentro da “Lista”. Imagine-se que se pretendia fazer, em Scratch, uma lista de telefones com os campos “Nome”, “Telefone” e “Fax”.

Usando variáveis individuais, cujos nomes não podem repetir-se, teríamos que as definir como “Nome1”, “Nome2”, etc., mais “Telefone1”, “Telefone2”, etc., mais “Fax1”, “Fax2”, etc., o que daria imenso trabalho para definir, para preencher e para consultar. Por isso se inventaram as variáveis-lista que, no Scratch, têm apenas uma dimensão linear. Aliás, as “Listas” do Scratch estão para as “variáveis individuais” assim como uma linha está para um ponto. (É bom saber que em linguagens de programação mais potentes, as “Listas” podem ter mais de uma dimensão, por exemplo: uma grelha bidimensional).

Se, em vez de variáveis individuais, usarmos variáveis “Lista” na nossa lista dos telefones, bastará criar três: “Nomes”, “Telefone” e “Fax”, sabendo que cada entrada na lista terá um índice (nº de ordem) que permite alinhar as três variáveis.

Depois das três listas preenchidas, se quisermos saber o nº de telefone do “Manelito”, por exemplo, só teríamos que mandar o Scratch ler todas as entradas na lista “Nomes” até encontrar o “Manelito” e, depois, com o nº de ordem do “Manelito”, iríamos à lista “Telefones” ler o que lá estivesse. (É bom saber que noutras linguagens há funções para fazer isto automaticamente).

Para fazer uma aplicação prática das listas, usei o projecto “Diálogos”, que gravei como “Variáveis-2” e juntei uma lista com as falas do Diabo, outra com as falas do Judeu e uma terceira com os tempos de duração de cada fala, para mostrar que as “Listas” tanto podem conter caracteres alfabéticos como números.

O bloco da figura ao lado tem 3 comandos para limpar as 3 “Listas”, seguidos de 12 para as preencher e mais uma variável simples para contar quantos elementos tem cada “Lista”.

FalasJudeu	FalasDiabo	Segundos
1 Que vai cá, ó marinheiro?	1 Oh! Que em má hora vieste!	1 3
2 Cuja é esta barca que preste?	2 Esta barca é do barqueiro.	2 4
3 Passai-me, por meu dinheiro.	3 E o bode há-de cá vir?	3 4
4 Pois também o bode há-de ir.	4 Que escusado passageiro.	4 5

```
quando receber Lista
faz Número igual a 1
repete Contador vezes
  diz item Número de FalasJudeu durante item Número de Segundos segundos
  espera item Número de Segundos + 0.5 segundos
  altera Número em 1
```

```
quando receber Lista
faz Número igual a 1
repete Contador vezes
  espera item Número de Segundos + 0.5 segundos
  diz item Número de FalasDiabo durante item Número de Segundos segundos
  altera Número em 1
```



“Listas” ficam com o aspecto desta figura(esquerda), que pode conferir-se com os comandos do bloco acima.

Repare-se que cada “Lista” é uma coluna duma tabela que poderia ter mais colunas como, por exemplo, a posição (x,y) para colocar os sprites.

No Variáveis-2.sb, há um botão que “anuncia Lista” quando clicado; e em cada personagem existe um bloco com um ciclo que executa “Contador” vezes para “dizer” cada um dos quatro itens gravados na sua “Lista” de falas.

O item é referido pelo seu índice e este está representado pela variável “Número” que se inicia a 1 e se faz variar até 4; (o 5 já não é utilizado).

Veja-se como os comandos “diz” e “espera” trocam de posição de um para outro personagem; e como se soma 0.5 segs às esperas, que usam os tempos gravados na “Segundos”

Isto agora ficou um bocado complexo e este exemplo do uso de “Listas” nem é muito feliz; mas dá

para ver como se reduziu drasticamente o número de comandos necessários para realizar o diálogo do Diabo e do Judeu. O uso de índices torna tudo mais simples, mas mais abstracto. Por isso, vale a pena revisitar esses últimos blocos da página anterior, para se detalhar o que já foi dito e acrescentar algo mais.

As falas têm o formato “diz [msg] durante [seg] segundos”; e como no campo da mensagem podemos utilizar uma qualquer variável, usámos um dos itens da variável-lista “FalasDiabo”.

Aqui, temos de identificar que item queremos, indicando o seu nº de ordem na lista: o índice. Mas este tanto pode ser fixo como variar e, assim, usamos a variável “Número” que fazemos variar para se referir a cada um dos itens da “Lista”.

Por outro lado, o campo dos segundos tem que indicar um valor numérico que tanto pode ser fixo como vir de uma variável. Usámos os valores previamente guardados na lista “Segundos” e indicámos com a mesma “Número” que item dessa lista devia ser usado, com o que simplificámos muito o nosso trabalho.

Com tudo isto, para além das “Listas”, vimos mais duas das muitas utilidades das variáveis individuais: índices de “Listas” e contadores, como é o caso da “Número”; mas esta variável tem ainda uma outra particularidade, que eu defini quando a criei no projecto Variáveis-2.sb: é uma variável local.

Com efeito, uma variável tem um âmbito de validade (scope), que pode ser global ou local. (E por isso o Scratch pergunta se a variável que estamos a criar é “para todos os sprites” ou “apenas para este”.

Por vezes convém que todos os sprites usem variáveis comuns, que uns preparam e outros usam, promiscuamente; outras vezes, convém que cada sprite use as suas variáveis que mais nenhum conhece. Neste caso, até podem ter nomes iguais, como é o caso da “Número” que aparece no ecrã da seguinte maneira:



A “Número” do Diabo não é a mesma que a “Número” do Judeu.

Para terminar o estudo das variáveis, só faltará dizer que elas podem servir também para passar parâmetros para rotinas especializadas em realizar uma determinada tarefa.

Imagine-se que, num determinado projecto, precisamos de desenhar quadrados de lado variável, espalhados por todo o ecrã. Das diversas soluções existentes, adoptámos a de sortear a posição e o lado, chamando de seguida uma rotina especializada em desenhar quadrados.

O código deste projecto poderia então ser como este aqui ao lado. No início define-se a cor e a espessura do traço e, quando se recebe “Quadrado”, sorteiam-se as coordenadas da posição e o comprimento do lado do quadrado.

Depois, chama-se o bloco “Desenhador” e espera-se que ele acabe de desenhar um quadrado usando os valores de X, Y e L. Se não esperássemos, o ciclo recomeçaria e iam alterar-se os valores quando o “Desenhador” ainda estava a desenhar, pelo que o quadrado, certamente, ficaria pouco ortogonal.

Repare-se como a variável “L” pode servir para comandar o ciclo “para sempre se...”. Nestas circunstâncias, se o número sorteado para “L” for 30, ainda é feito um quadrado com esse lado mas o ciclo não se repetirá, pois nunca mais “L” mudará de valor.

Merece aqui destaque o operador “não” a negar a igualdade L=30; e, no final do “Desenhador”, a maneira como invertemos o sinal de “L”.

