

8.2 Ciclos **WHILE**



Executa um conjunto de operações **ATÉ** que uma condição lógica seja satisfeita.

Não requer o conhecimento a priori do número de ciclos a efectuar.

- A estrutura **WHILE** já contém o seu próprio contador [*i*] que inicia em **zero** e incrementa automaticamente.



8.2 Ciclos WHILE



Construa e execute o seguinte VI:

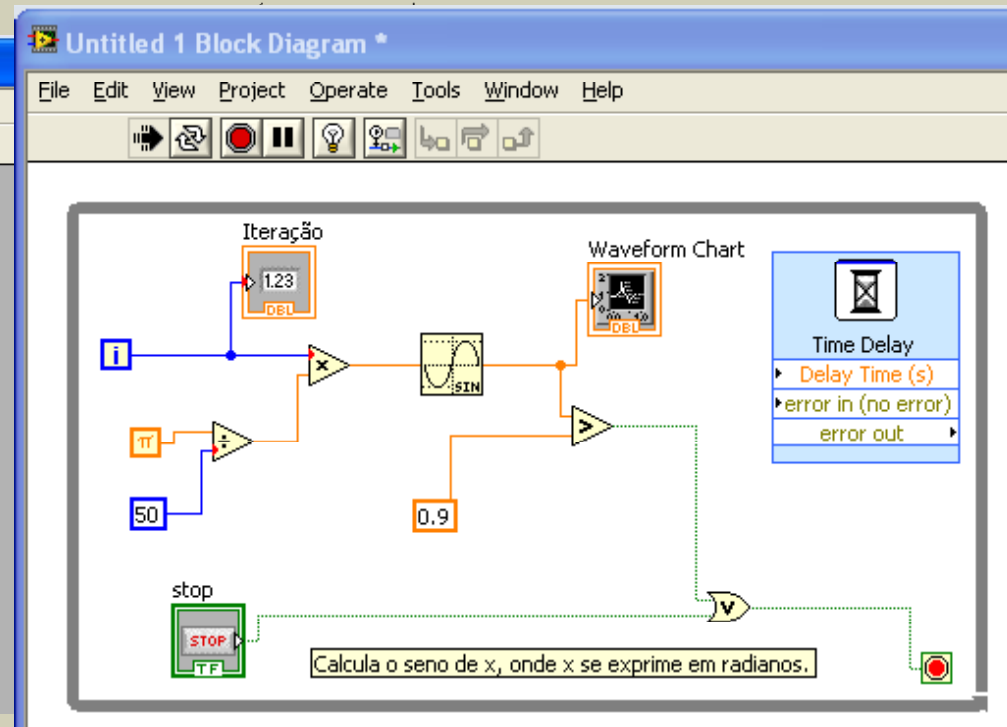
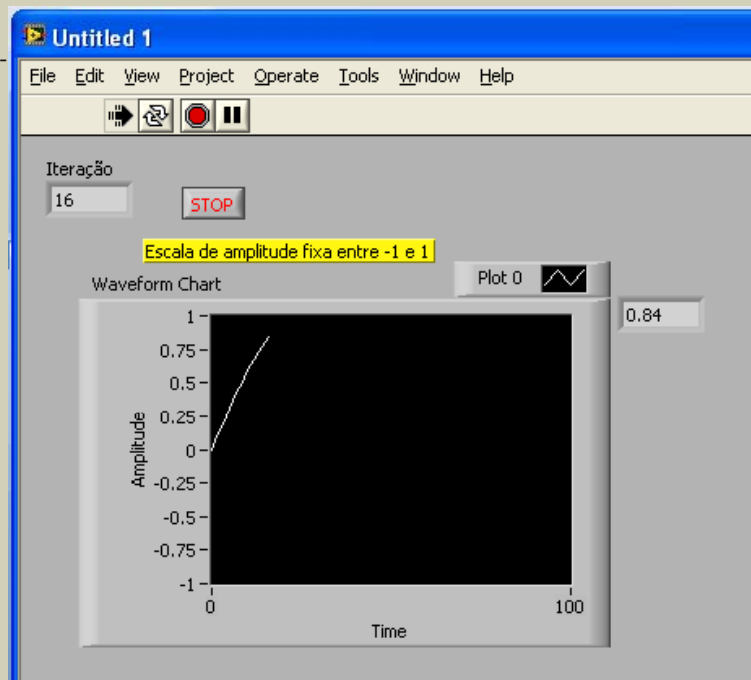
The screenshot illustrates the LabVIEW environment with three main components:

- Front Panel (Untitled 1):** Shows a 'STOP' button and a 'Numeric' control with the value '2'.
- Block Diagram (Untitled 1 Block Diagram):** Shows a flow starting from an information icon, passing through a 'Numeric' control (value 1.23), then a 'Time Delay' block (circled in red), and finally a 'STOP' button. A red arrow points from the 'Time Delay' block to the 'Configure Time Delay' dialog.
- Configure Time Delay [Time Delay] Dialog:** Shows the 'Time delay (seconds)' field set to '1.000'. Buttons for 'OK', 'Cancel', and 'Help' are visible.
- Timing Palette:** A palette of timing-related functions is shown, with 'Time Delay' selected.
- Functions Palette:** A portion of the main functions palette is visible on the right, showing categories like 'Programming', 'Measurement I/O', and 'Signal Analysis'.

8.2 Ciclos WHILE



EXEMPLO: Terminar o ciclo WHILE caso o utilizador pressione STOP ou a amplitude de um sinal sinusoidal atinja o valor 0.9.

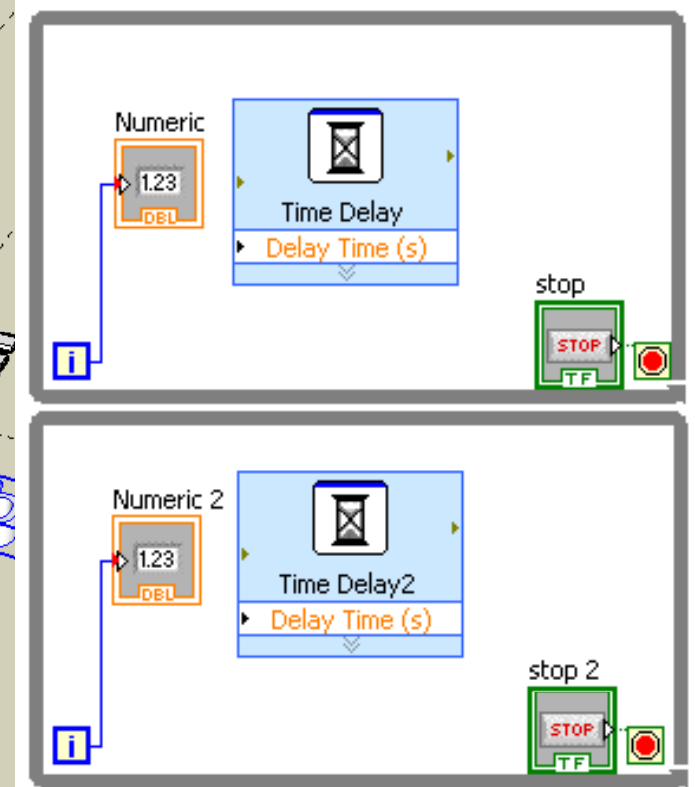
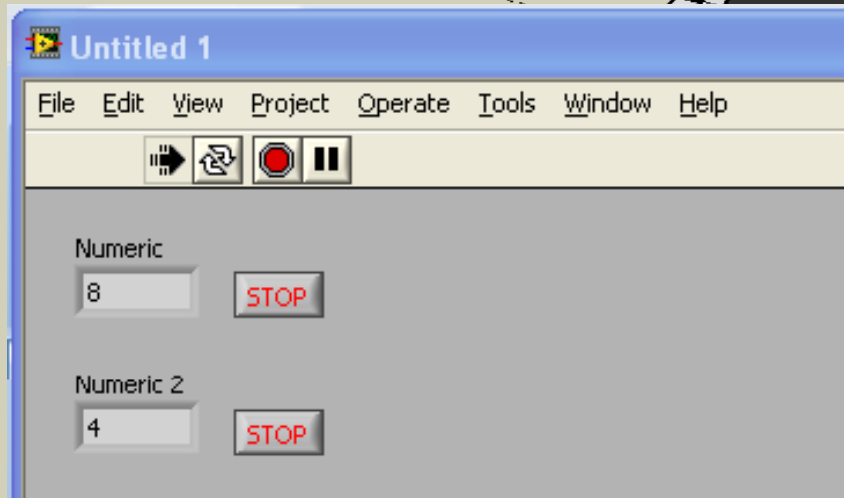


8.2 Ciclos WHILE



O LabVIEW permite a execução concorrente de vários processos simultaneamente.

EXEMPLO: Colocar a execução em paralelo de dois ciclos WHILE com diferentes tempos de atraso (1 e 2 segundos).



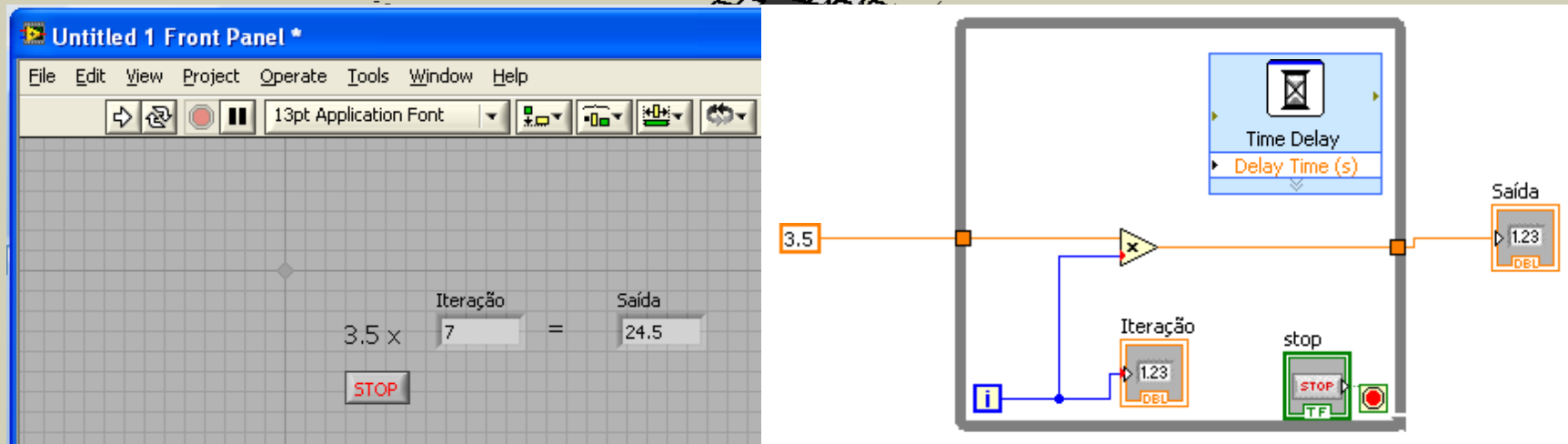
120

8.2 Ciclos WHILE



Tal como o ciclo **FOR**, o ciclo **WHILE** também permite instaurar túneis para a passagem de dados.

EXEMPLO: Construa e simule o seguinte VI...



8.3 Exercícios



EX 21: Construa um ciclo que se mantenha em execução até que o valor originado, por um gerador de números aleatórios inteiros entre 0 e 100, seja igual a 50. Coloque um atraso de 100 ms dentro do ciclo e apresente, no **front panel**, o número de iterações produzidas.

EX 22: Apresente um VI capaz de fazer um LED piscar com uma frequência igual a 1Hz:

a) Com recurso a shift-registers

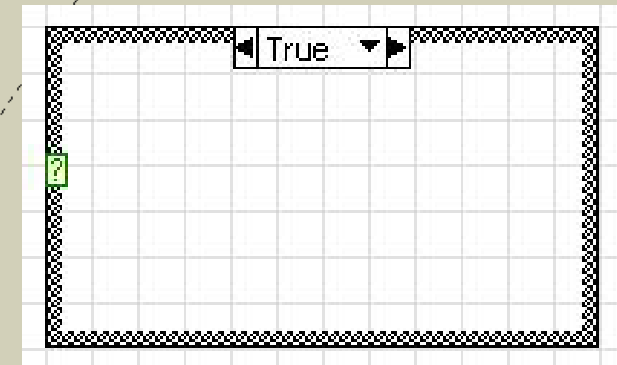
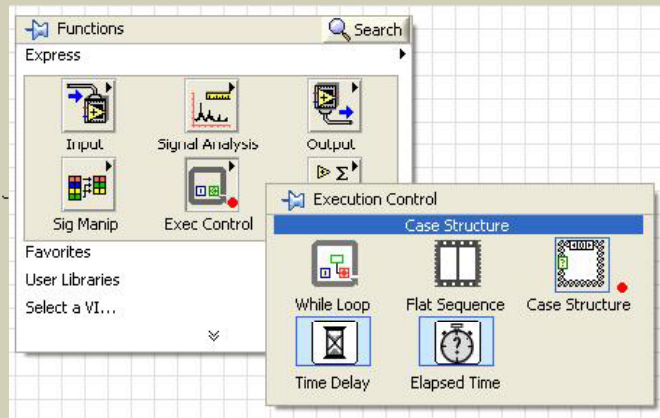
b) Sem recurso a shift-registers (quando [i] for “par” o LED acende e quando [i] for ímpar o LED apaga)

EX 23: Desenvolva um VI capaz de calcular o factorial de um valor arbitrário (positivo, inteiro e menor que 100)

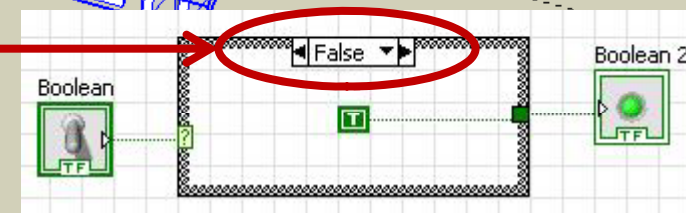
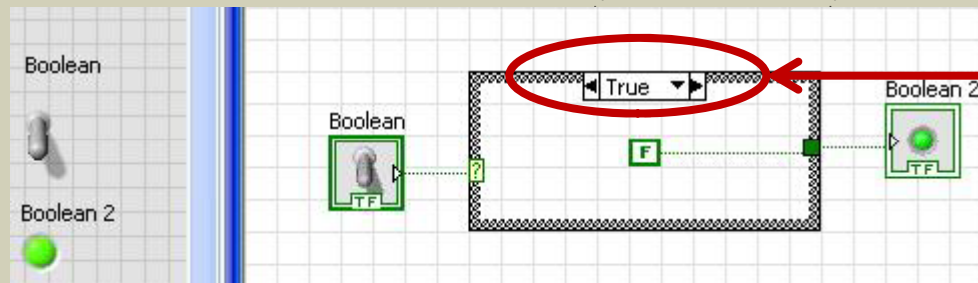
8.4 Estrutura CASE



Estruturas do tipo **CASE** permitem que determinadas acções tomem lugar em função de possíveis valores de uma variável de decisão.



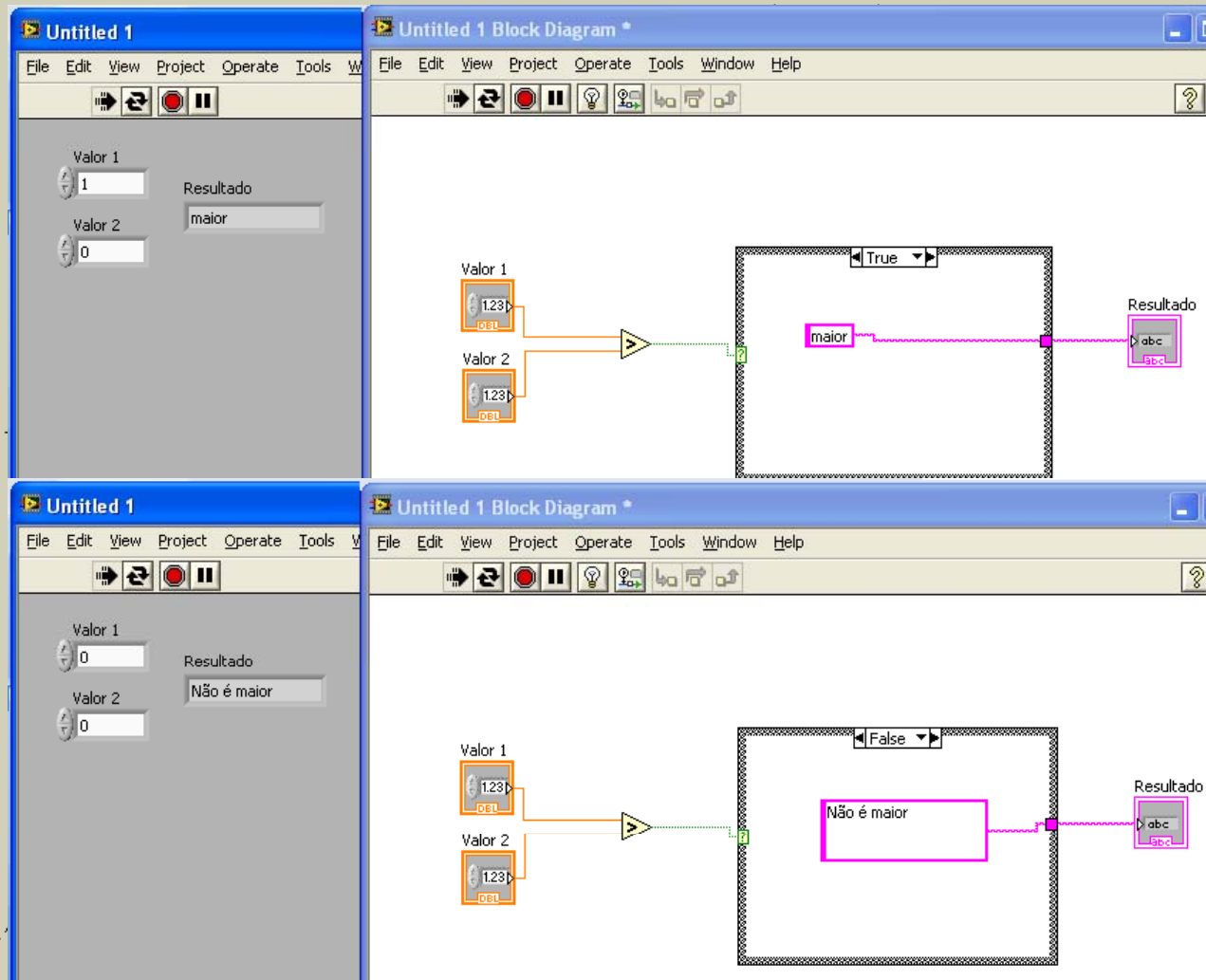
EXEMPLO: Construa e simule o seguinte VI...



8.4 Estrutura CASE



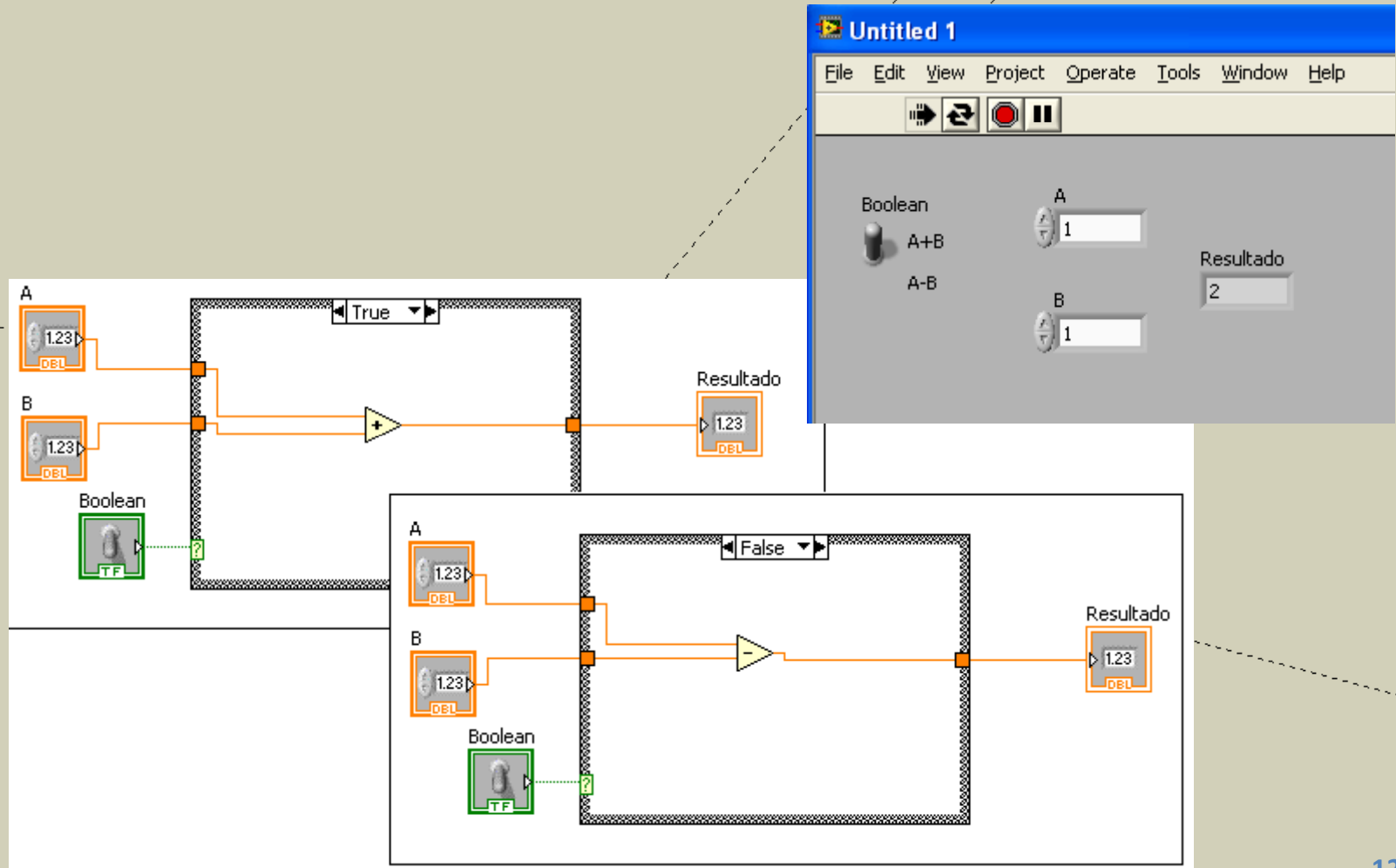
• CONDIÇÕES LÓGICAS



8.4 Estrutura CASE



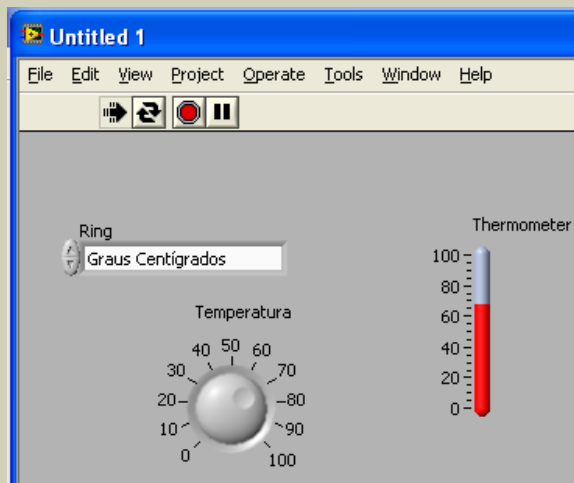
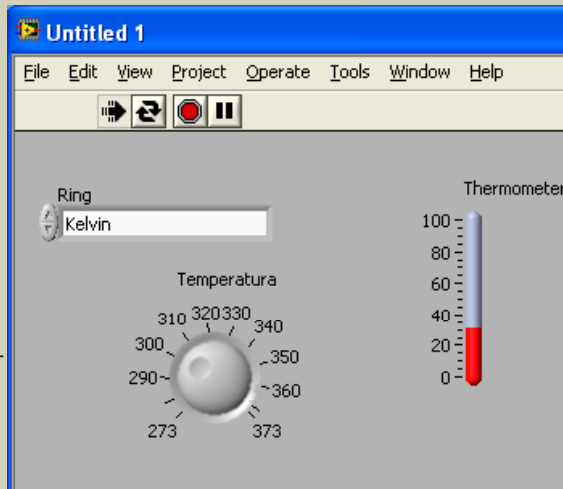
• CONDIÇÕES LÓGICAS



8.4 Estrutura CASE



• CONDIÇÕES: INTEIROS

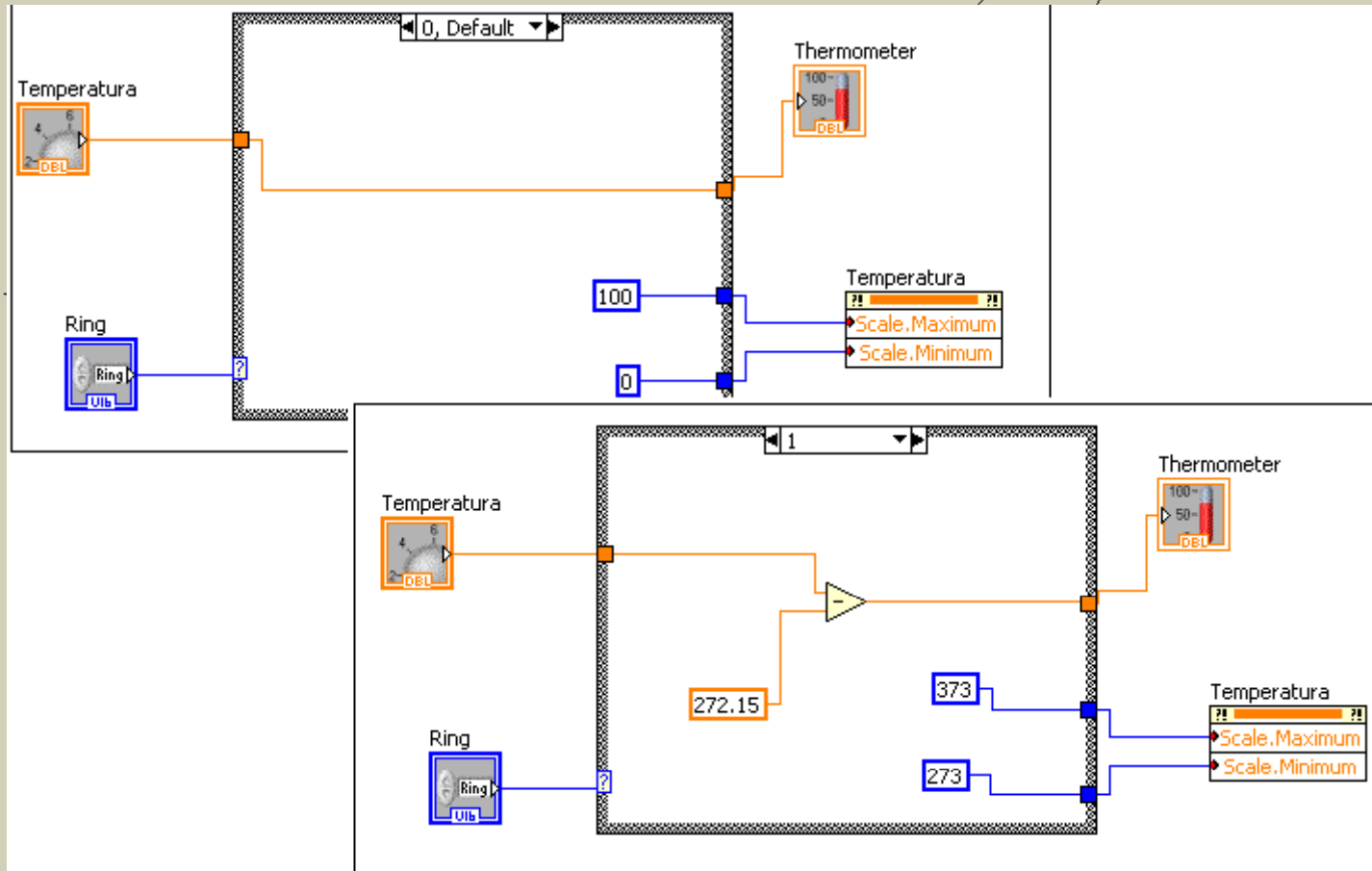


Temperatura

- Visible Items
- Find
- Hide Control
- Change to Indicator
- Change to Constant
- Description and Tip...
- Numeric Palette
- Create**
 - Constant
 - Control
 - Indicator
 - Local Variable
 - Reference
 - Property Node**
 - Invoke Node
- Data Operations
- Advanced
- ✓ View As Icon
- Representation
- Properties

- Blinking
- Caption
- Data Binding
- DataSocket
- Description
- Disabled
- Focus Key Binding
- Indicator
- Key Focus
- Label
- Skip When Tabbing
- Synchronous Display
- Tip Strip
- Value
- Value (Signaling)
- Visible
- XControl
- Data Entry Limits

8.4 Estrutura CASE



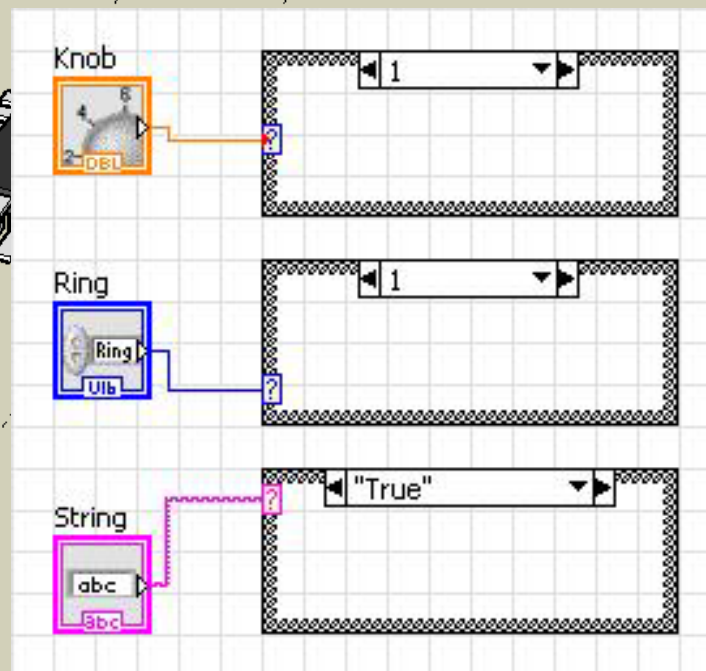
8.4 Estrutura CASE



Por defeito a estrutura CASE é apresentada com dois casos distintos: TRUE e FALSE .

A paleta de opções pode ser facilmente estendida para situações não-booleanas.

O objecto CASE adapta-se automaticamente ao controlo associado à variável de decisão.

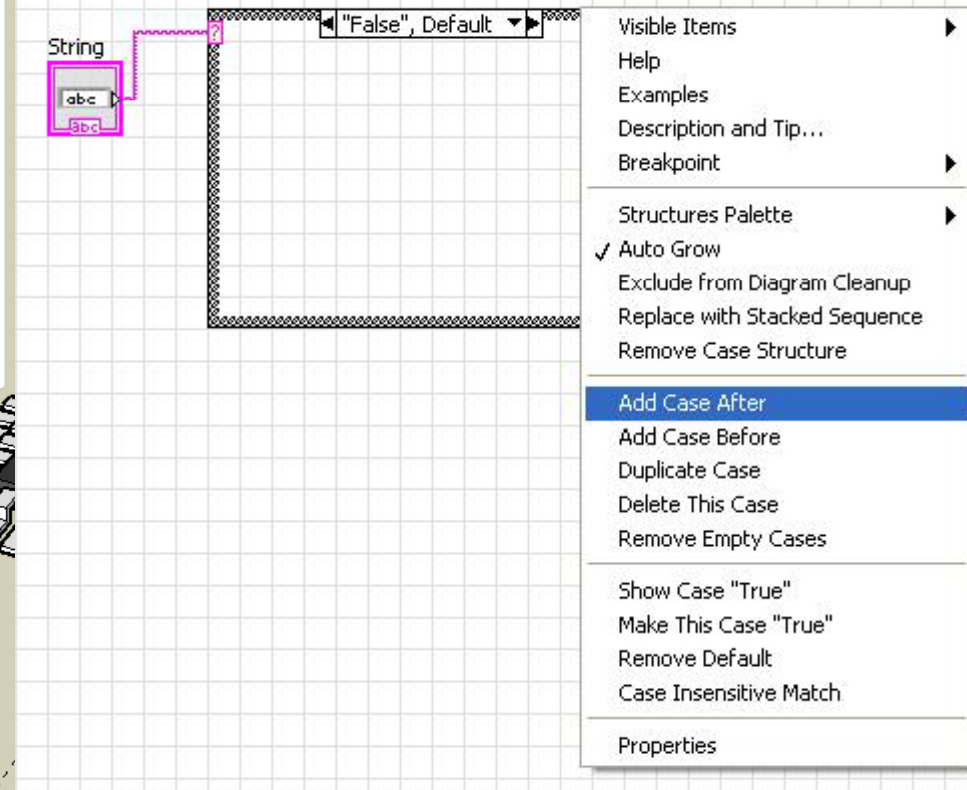


128

8.4 Estrutura CASE



O número de situações a serem contempladas pela estrutura CASE podem ser aumentadas ou diminuídas recorrendo ao **pop-up** menu associado a esse objecto...

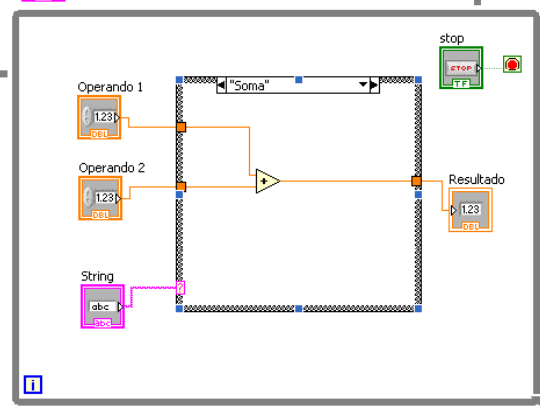
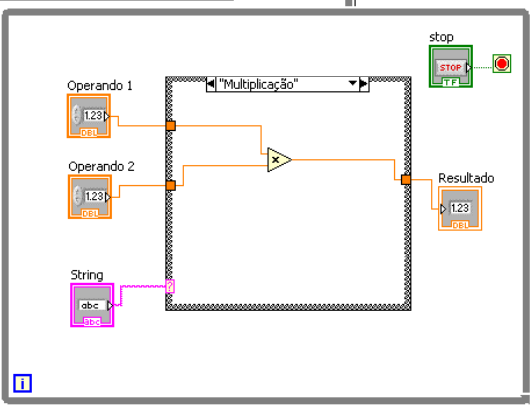
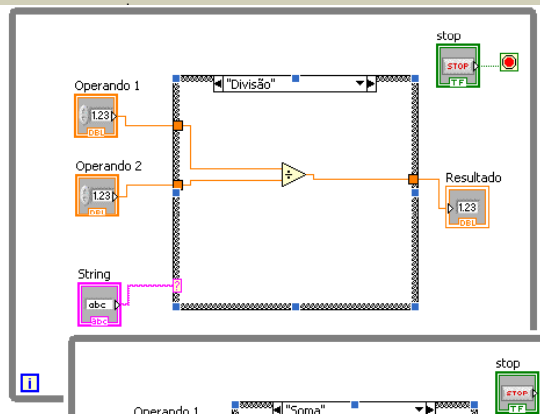
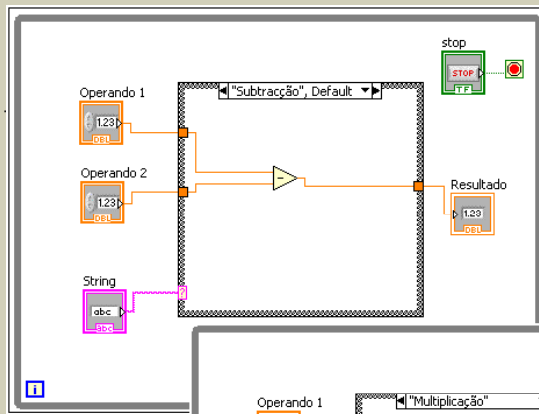
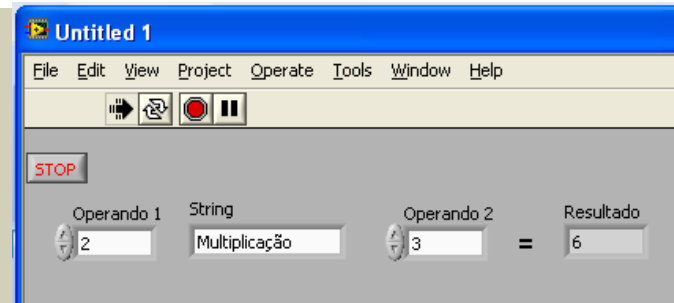


8.4 Estrutura CASE



• CONDIÇÕES: STRINGS

CASE dentro de WHILE



8.4 Exercícios



EX 24: Um LED deve ser comandado por um **String Control** de acordo com as seguintes ordens: **Acende, Apaga e Pisca.**

EX 25: Controlo da cor do fluído no interior de um tanque.



8.4 Exercícios



EX 26a: Construa um SubVI capaz de gerar o efeito do lançamento, ao ar, de uma moeda. A saída será booleana, i.e. VERDADEIRO para CARA e FALSO para CORÔA.

EX 26b: Pretende-se construir um jogo de apostas com base no subVI do exercício anterior. O jogador possui a capacidade de apostar no resultado do próximo lançamento da moeda usando um botão. Quando o botão estiver ON admite-se que o jogador aposta em “caras” caso contrário aposta em “coroas”.

Após o lançamento (gerado por um segundo botão) o resultado “ganhou” ou “perdeu” deve ser apresentado num string indicator.



132

8.4 Exercícios



EX 26c: Adicione ao programa anterior um controlo e um indicador numérico. O indicador numérico será responsável por apresentar o saldo do jogador e o controlo deverá representar o valor apostado pelo jogador numa determinada jogada. A figura em baixo ilustra a nova interface gráfica.

Caso o jogador ganhe, o valor da aposta é acrescentado ao saldo. Se o jogador perder o valor da aposta será retirado do saldo

EX 26d: O saldo deve ser iniciado com 50€. Para além disso o jogo deve terminar quando o saldo do jogador chegar a zero ou então desistir. Não devem ser permitidas apostas superiores ao saldo.



133

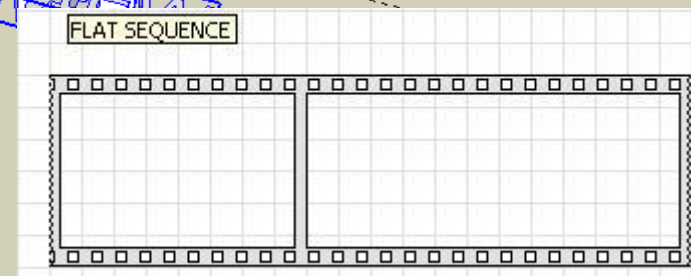
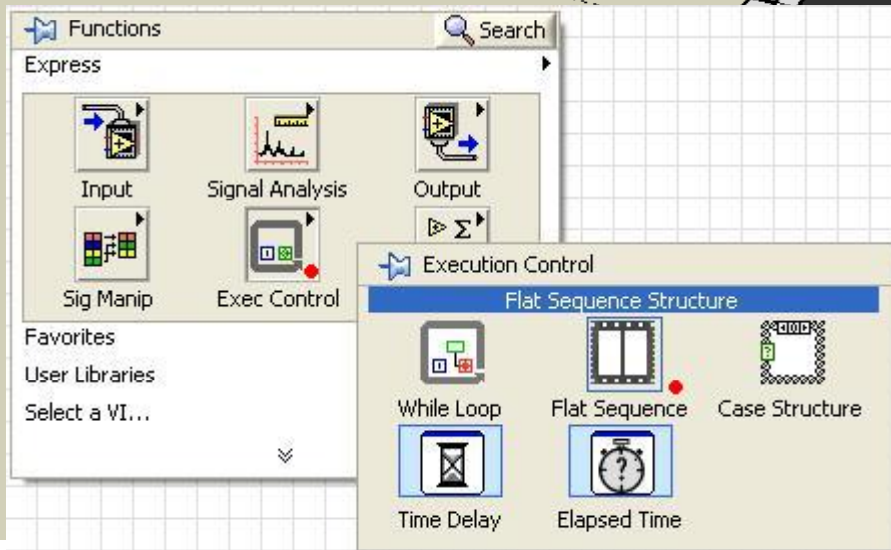
9.0 Sequências



Por defeito, no diagrama de blocos, todos os eventos ocorrem de forma **concorrente**.

A necessidade da execução sequencial de diagramas obriga à utilização de um tipo alternativo de estrutura de controlo: **sequências**

ADORO CONCORRENTES!!!!

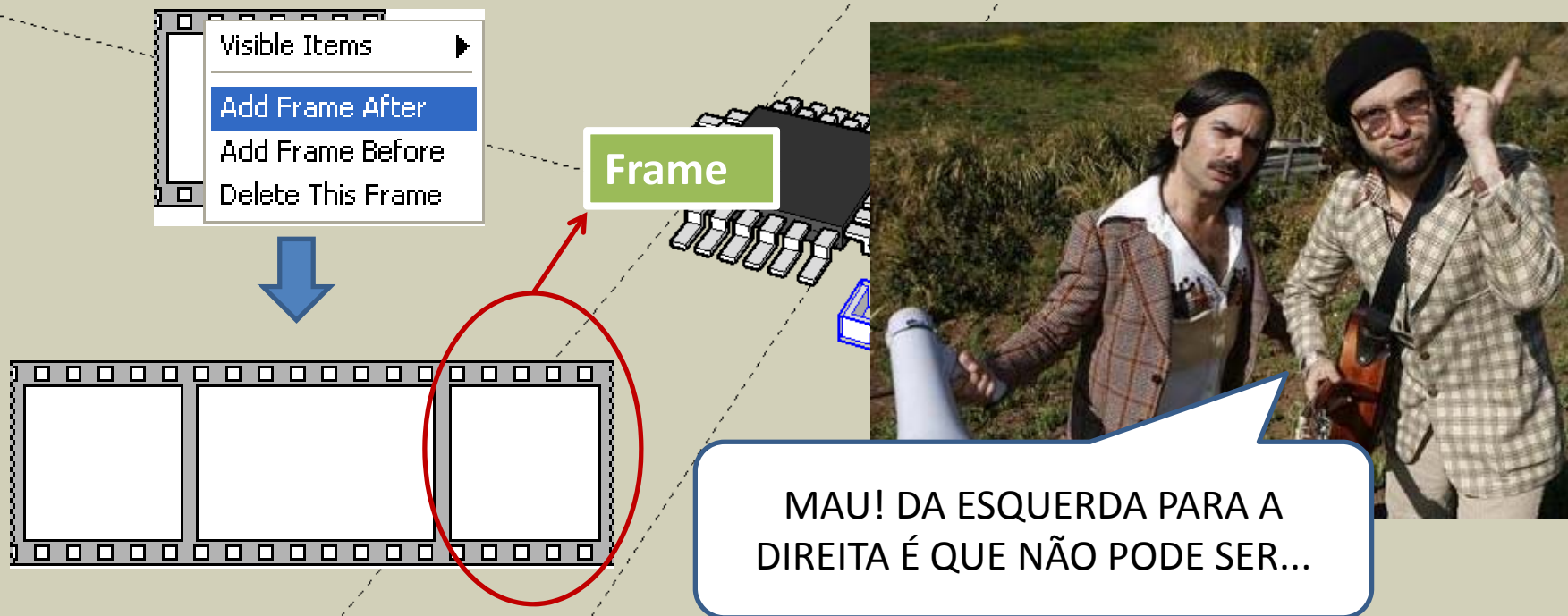


9.1 Sequências Planas



Sequências planas (*flat sequences*)

- São sempre executadas da **esquerda** para a **direita**.
- É iniciado apenas com uma “frame” e permite a adição de outros quadros.



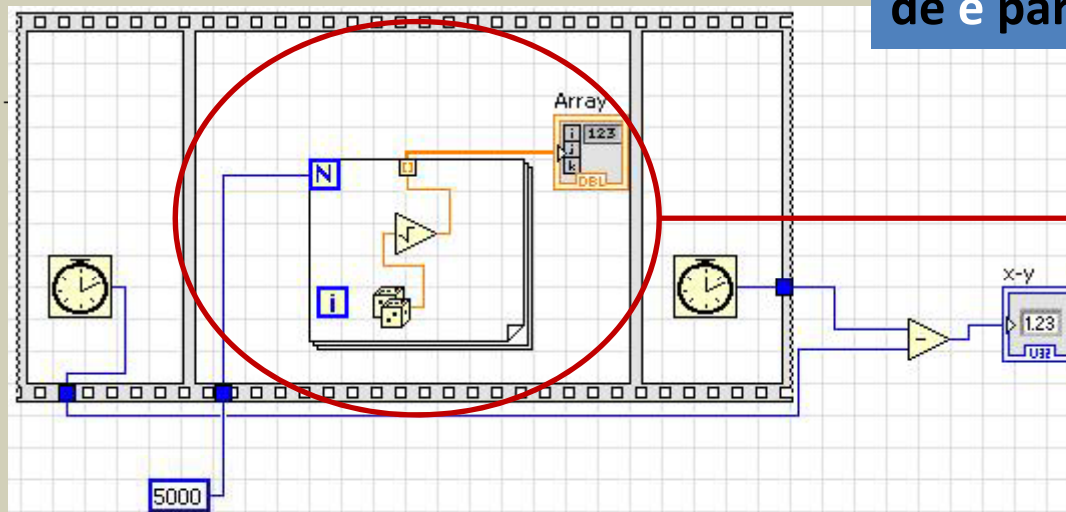
135

9.1 Sequências Planas



EXEMPLO: Construa e simule o seguinte VI...

Observe-se a existência de Túneis de e para a sequência...

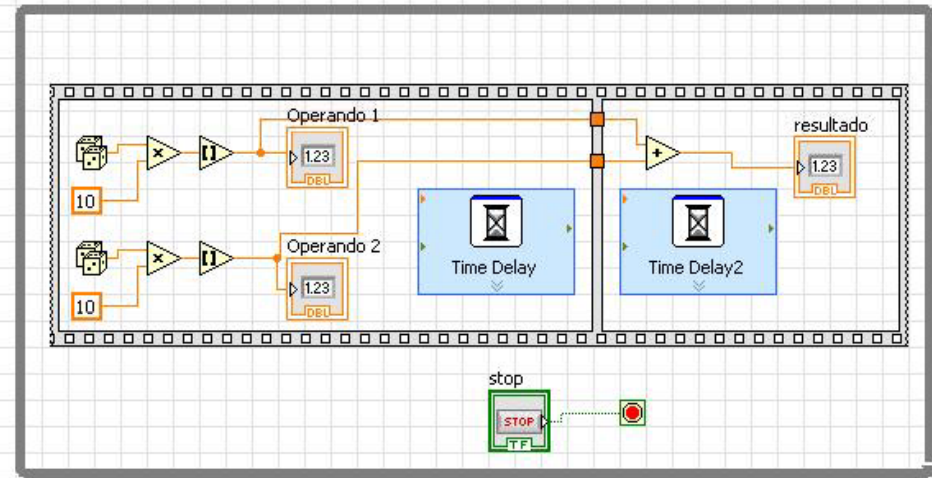
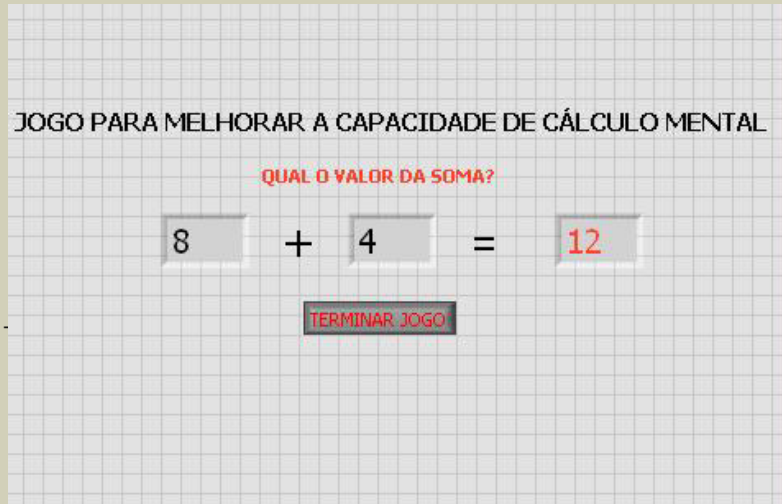


Subdiagrama

EXEMPLO: Jogo dos Números...

Construir um VI que apresente dois números inteiros entre 0 e 10, aguarde 2 segundos e apresente o resultado da soma durante 1 segundo.

9.1 Sequências Planas



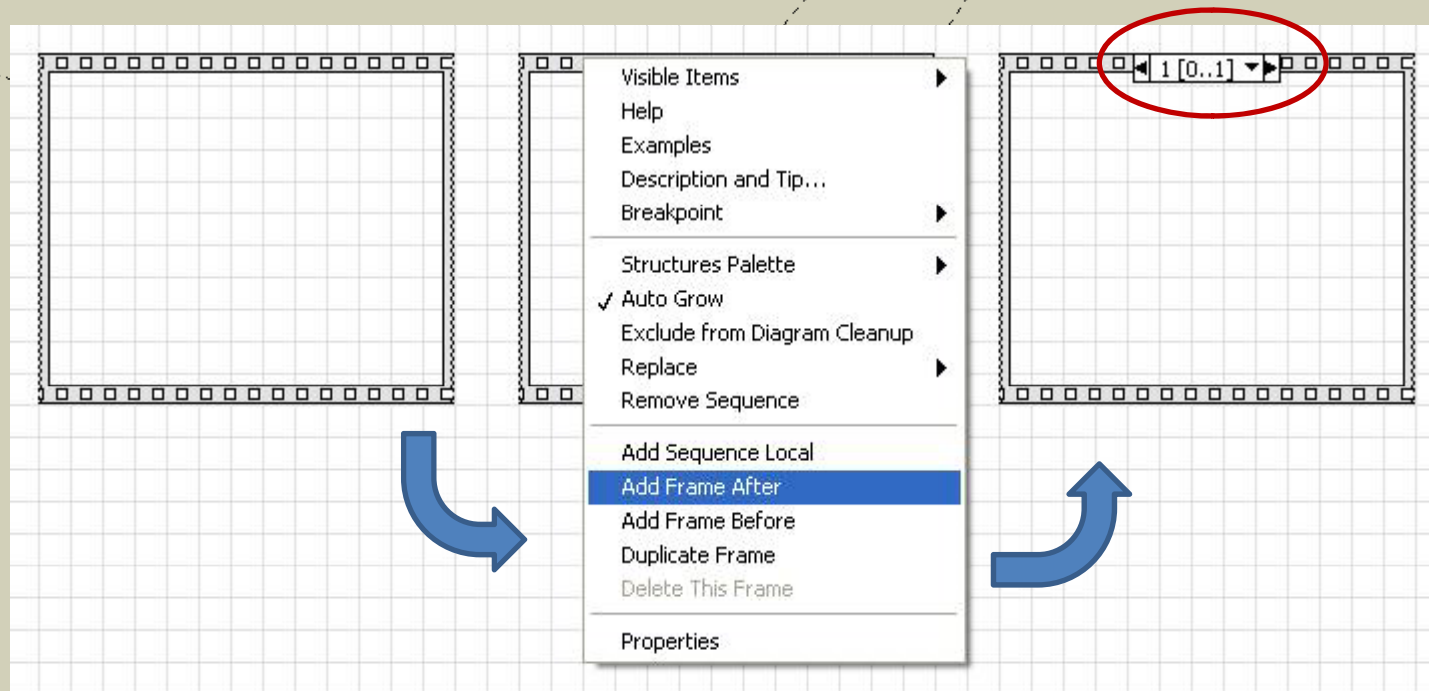
ESTE PROGRAMA É CAPAZ DE ME DAR JEITO... ANDO-ME A ENGANAR COM AS CONTAS DA REFORMITA!

9.2 Sequências Sobrepostas



EX 27: *Altere o “Jogo dos Números” de modo a que o utilizador tenha a possibilidade de escolher uma das quatro operações elementares.*

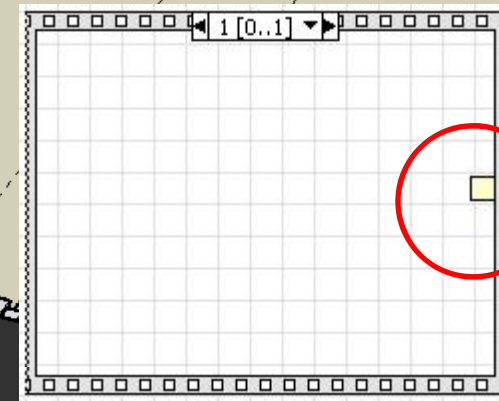
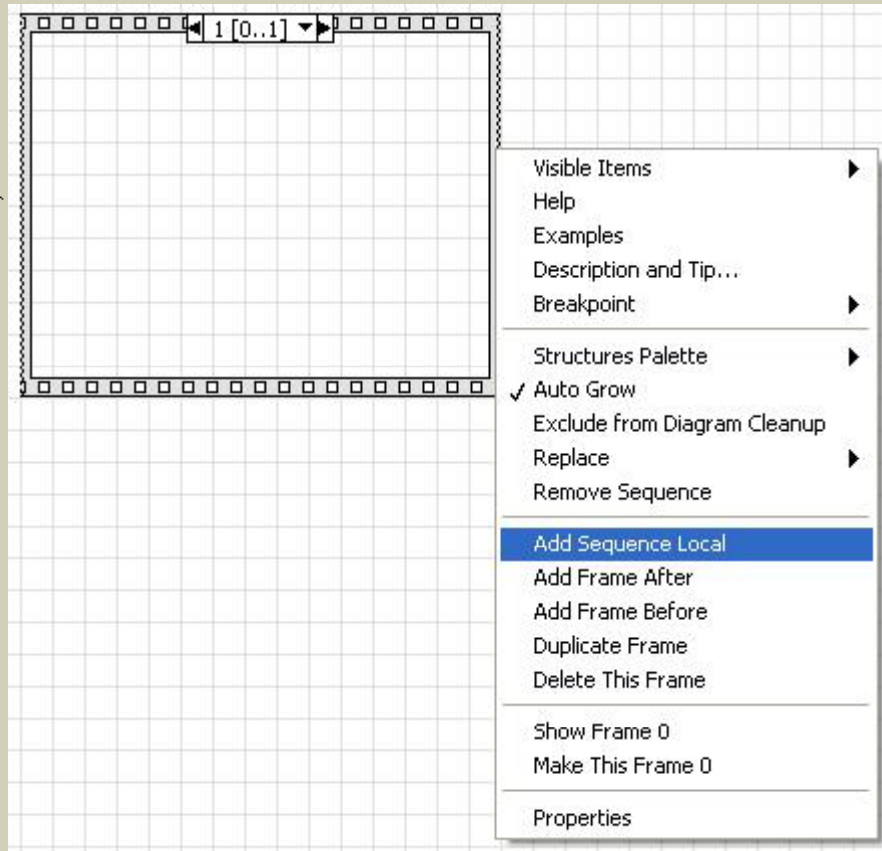
Referem-se a sequências cujas frames se encontram sobrepostas



9.2 Sequências Sobrepostas



Passagem de parâmetros entre frames: *variáveis locais*

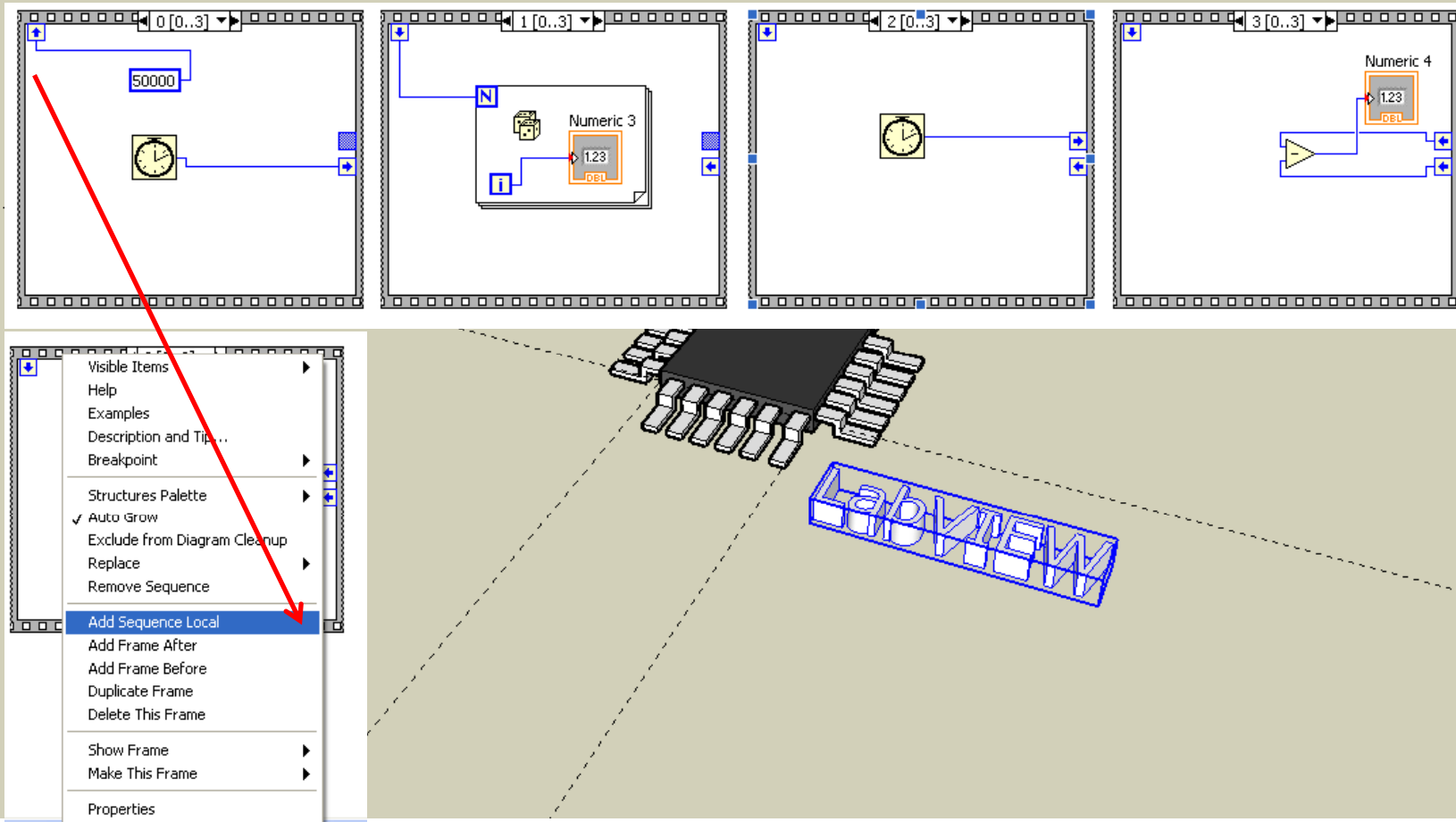


O valor de uma variável local criada num frame ficará disponível para todas as frames criadas apenas subsequentemente.

9.2 Sequências Sobrepostas



EXEMPLO: Construa e simule o seguinte VI...

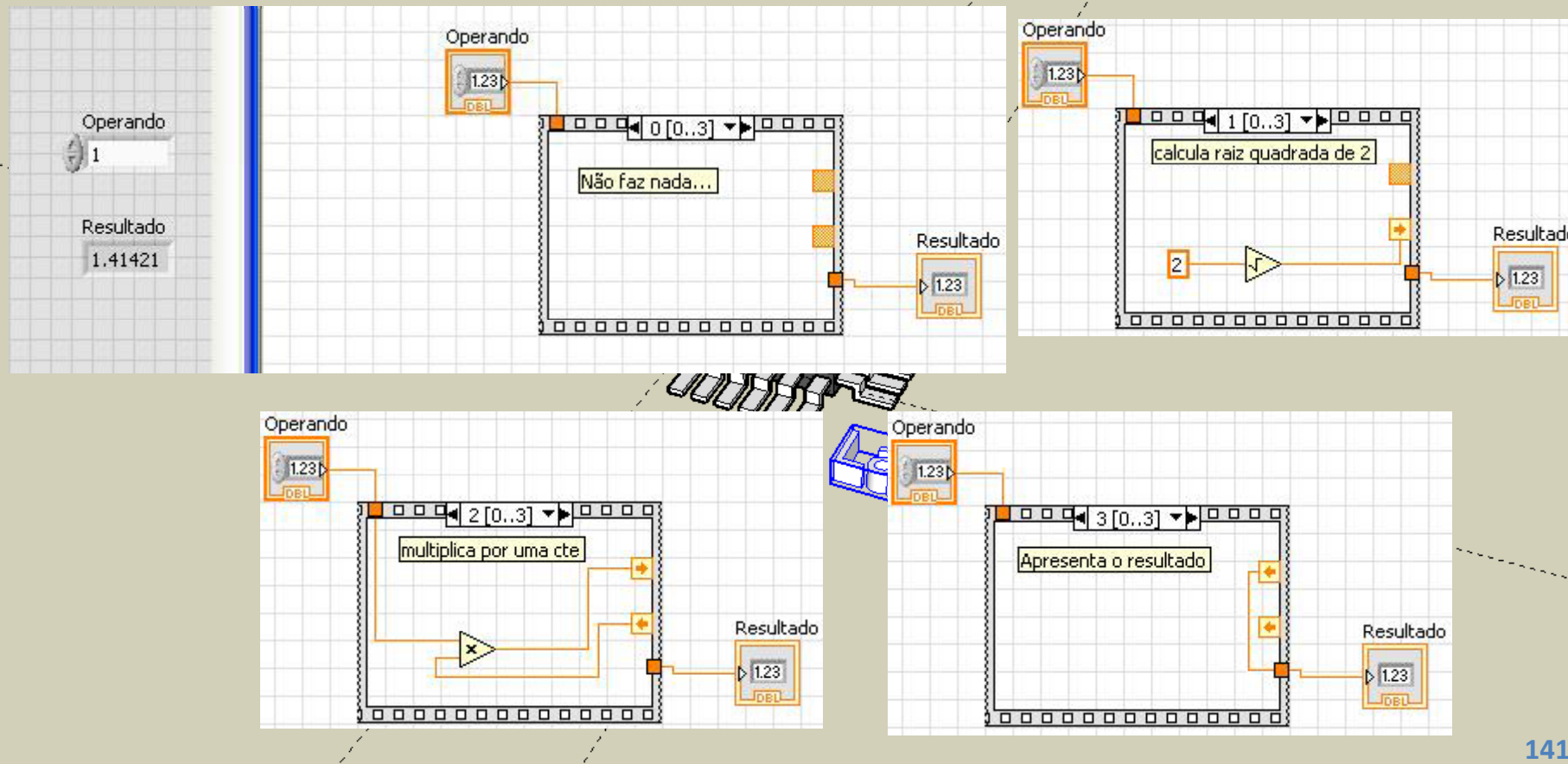


140

9.2 Sequências Sobrepostas



EXEMPLO: Construa e simule o seguinte VI...

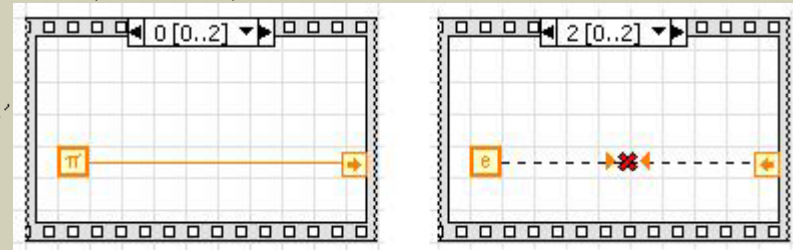


9.2 Sequências: algumas considerações



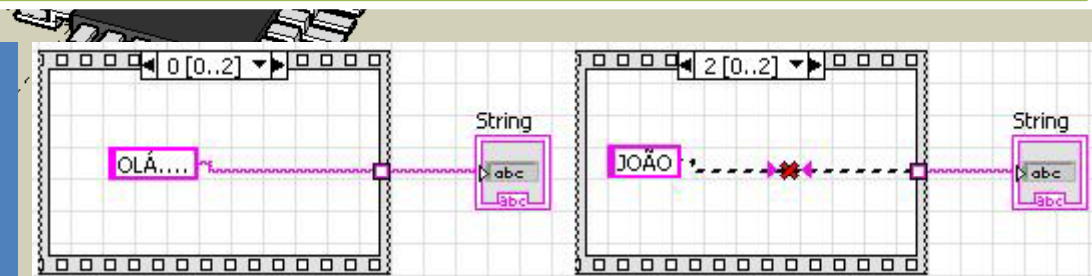
Uma **variável local** só pode ser instanciada **UMA** vez

Uma variável local é inicializada com o valor 3.141... Na frame 0 e não pode ser alterada, na frame 2, com o valor 2.718...



Duas frames **NÃO** podem atribuir valores ao **mesmo** túnel

Na frame 0 pretende-se atribuir a *string* “OLÁ” a um indicador e na frame 2 uma *string* diferente ao mesmo objecto.



Recorde a necessidade de efectuar justamente o contrário numa estrutura CASE.

9.3 Exercícios



EX 28: *Dois ciclos são executados simultaneamente. Para ambos o critério de paragem consiste na igualdade de um número inteiro, entre 0 e 100, gerado aleatoriamente dentro do ciclo ser igual a 50. Quando ambos os ciclos terminarem deve ser efectuada a soma do número de iterações de ambos. Se esse valor for menor que 100 deve ser accionado um LED vermelho. Caso seja maior, ou igual a 100, deve ser ligado um LED verde.*

EX 29: *Deve ser construído um VI que repita dez vezes o exercício anterior e preencha uma tabela com o resultado da soma dos ciclos em cada ensaio.*

EX 30: *Um ciclo FOR é executado 10 vezes. Em cada iteração é gerado um número aleatório inteiro entre 0 e 100. Quando este ciclo terminar, um segundo ciclo FOR é executado onde N será o maior dos valores aleatórios gerados no ciclo anterior.*

143