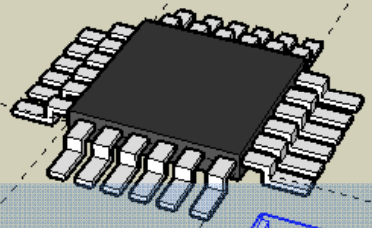
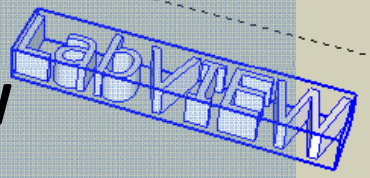
INTRODUÇÃO AO LabVIEW

INSTRUMENTAÇÃO ELECTRÓNICA E MEDIDAS

2011/2012





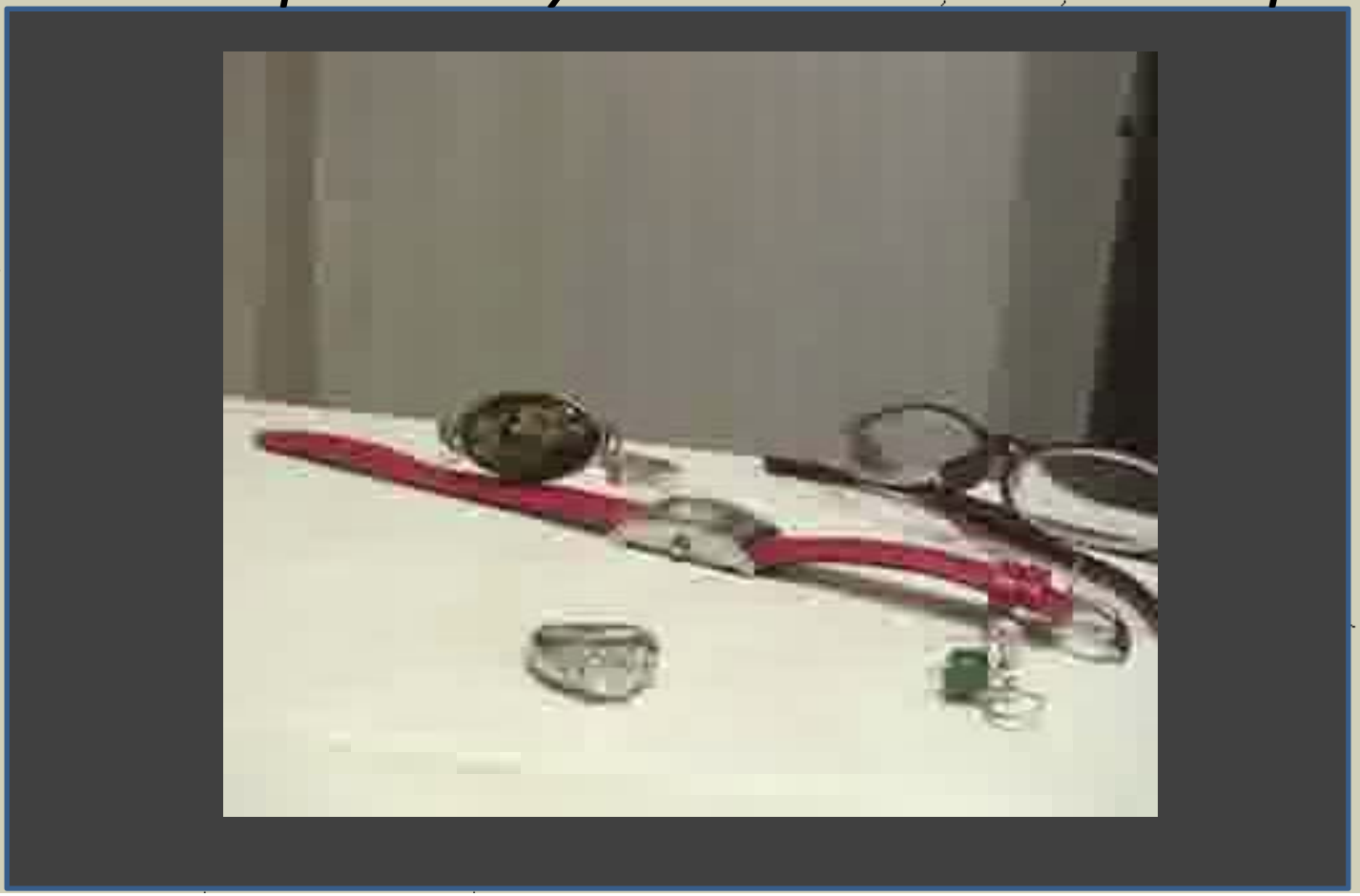
OBJECTIVOS:

- Introdução ao ambiente **LabVIEW**
- Modelo de programação em **G**
- Conceito de instrumentos virtuais (**VI**)
- Aquisição de dados e *Data Logging*

ipb INSTITUTO POLITÉCNICO DE BRAGANÇA
Escola Superior de Tecnologia e Gestão

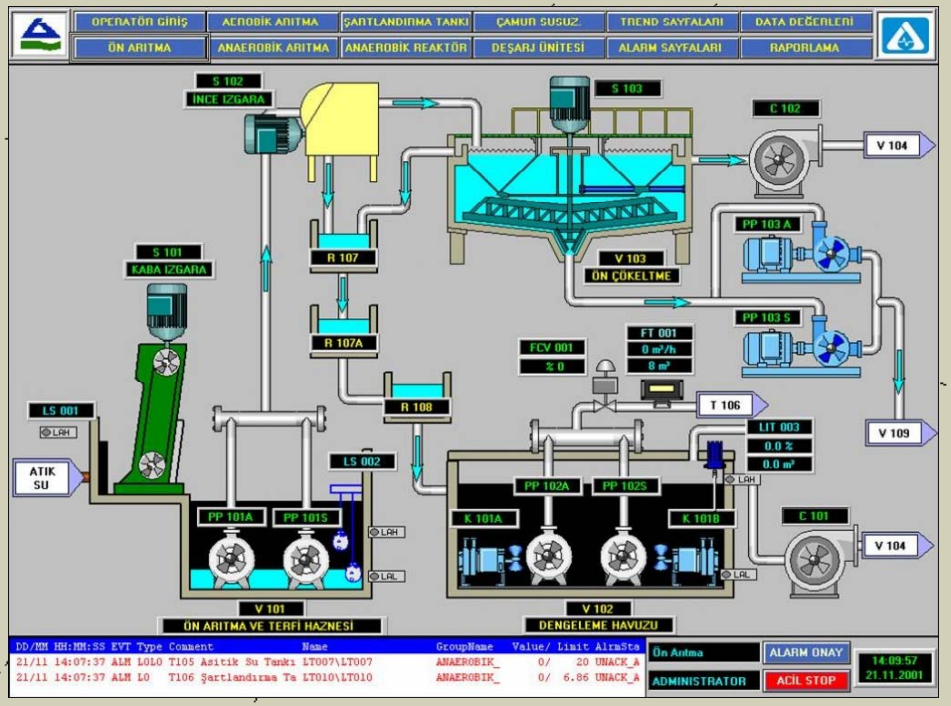
0.0 SCADA

- SCADA - ***s****upervisory* ***c****ontrol* ***a****nd* ***d****ata* ***a****cquisition*



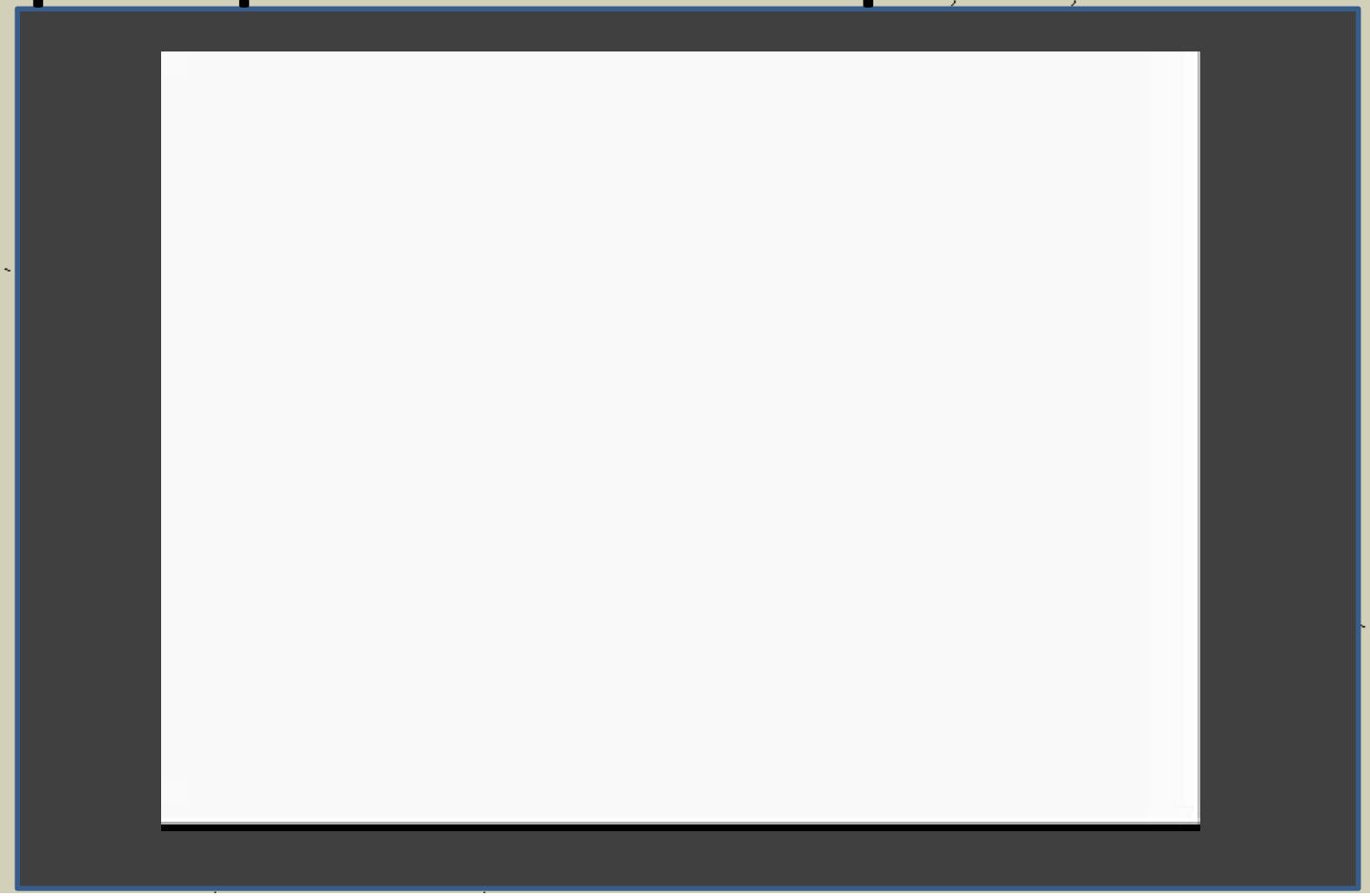
0.1 SCADA

“Hacker ganha acesso não autorizado a sistema de controle de água nos Estados Unidos e consegue destruir uma bomba de uma concessionária”



0.2 Aplicações do **LabView**

- **O que é que o LabView é capaz de fazer?**



1.0 Instrumentos Virtuais

- O LabVIEW funciona segundo um modelo de fluxo de dados.
- Um programa é designado por «Virtual Instrument» ou VI.
- A informação flui de «fontes» para «drenos» através de ligações.
- A informação (dados) são alterados ao longo do percurso.

1.0. Instrumentos Virtuais

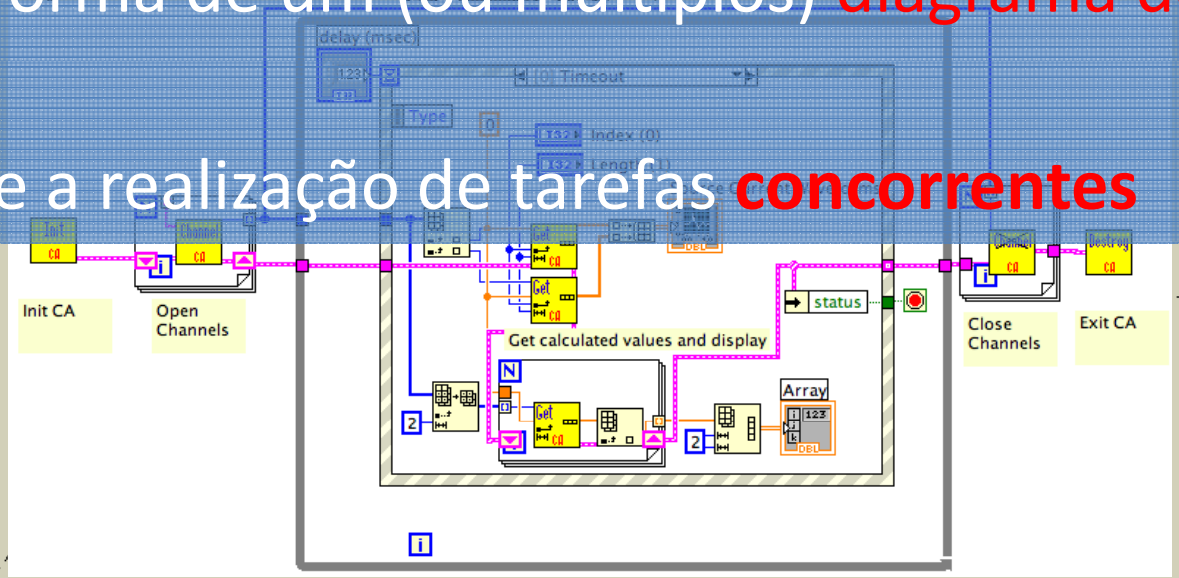
- O LabVIEW suporta dois tipos de VI:
  - VI **internos**
  - VI **criados pelo utilizador**
- VI internos são fornecidos pela aplicação e executam tarefas básicas.
- VI criados pelo utilizador consistem num GUI e num diagrama de blocos.
- Um programa em LabVIEW é definido sob **duas** plataformas:

1.0. Instrumentos Virtuais

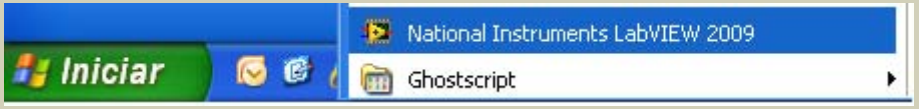
- **FONT-END**
  - Permite **interagir directamente** com o utilizador.
  - **Apresenta** informação sobre o processo a monitorizar ou controlar.
  - Permite ao utilizador **modificar parâmetros** do sistema (SCADA)

1.0. Instrumentos Virtuais

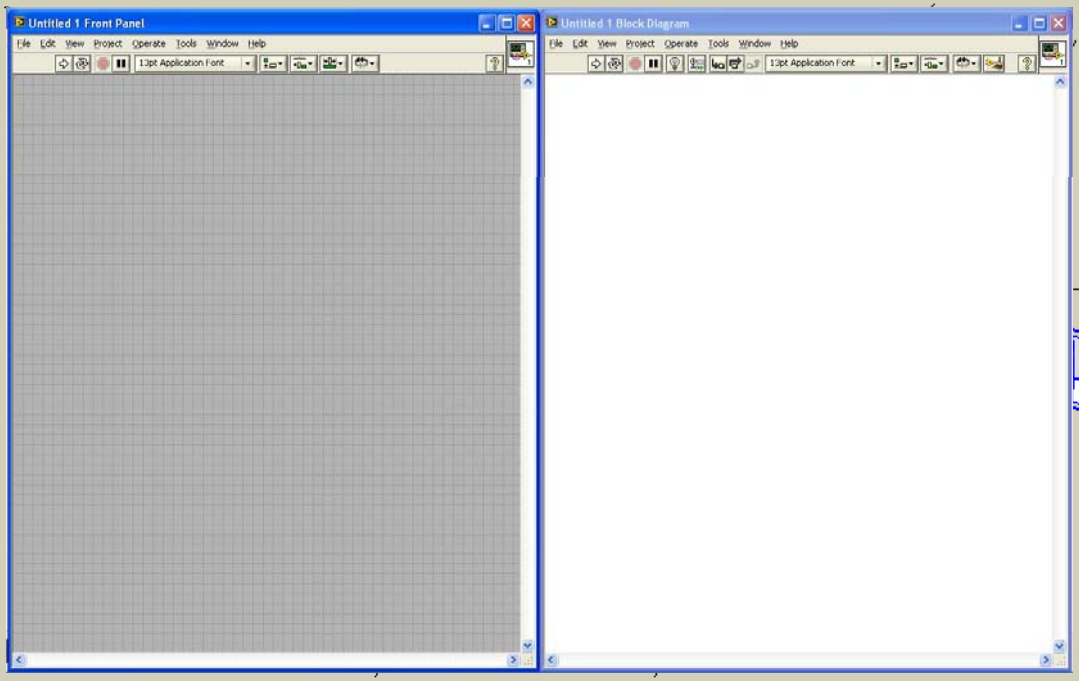
- **BACK-END**
  - Programa por detrás do **FRONT-END**.
  - Programação feita numa linguagem gráfica: **G**
  - Tem a forma de um (ou múltiplos) diagrama de blocos.
  - Permite a realização de tarefas **concorrentes**



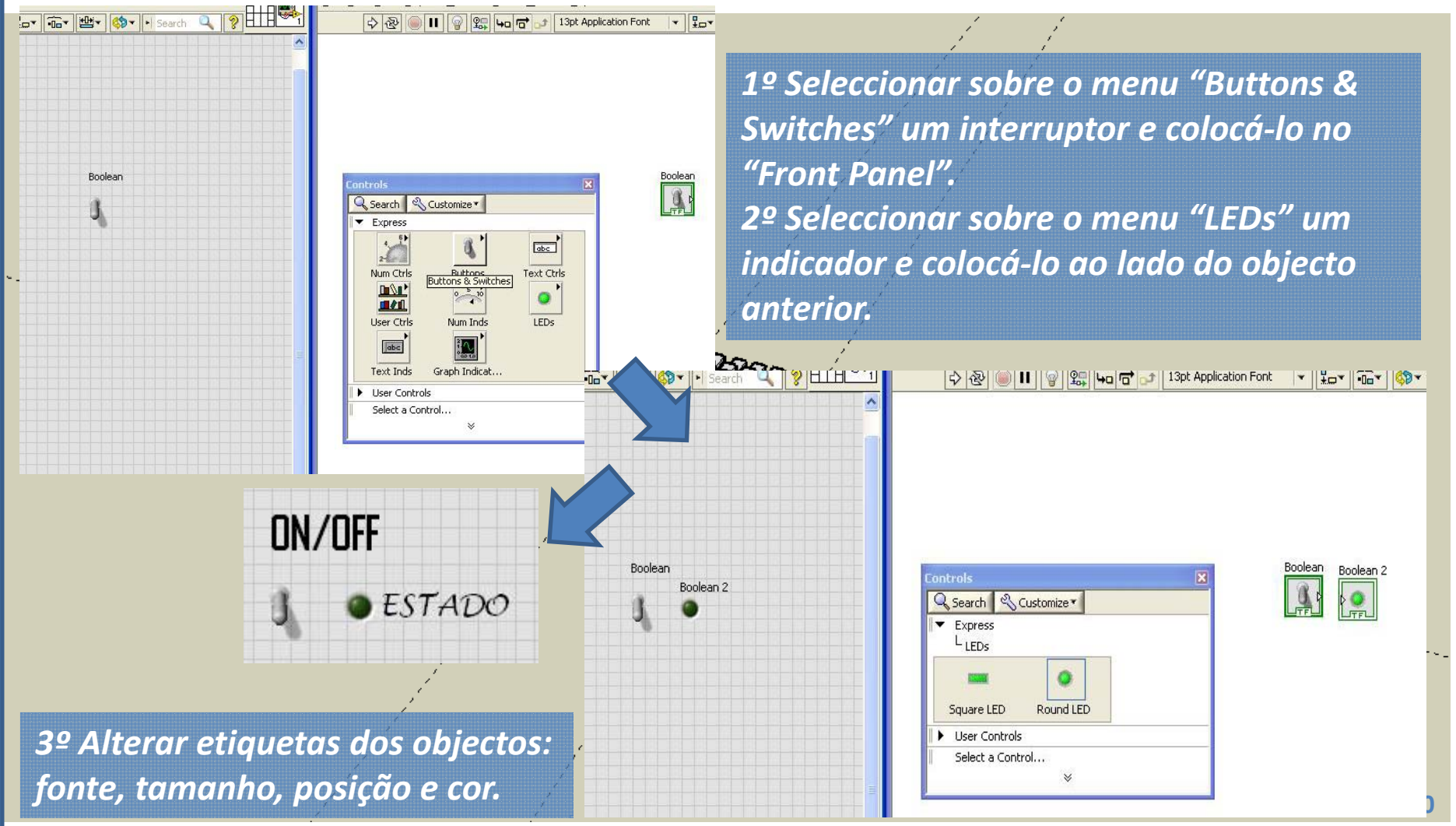
2.0 Construção de um VI



New VI



2.0 "Olá Mundo..."

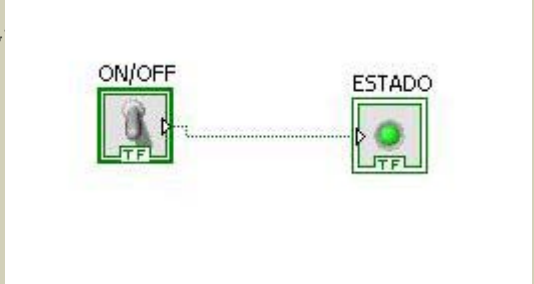


1º Seleccionar sobre o menu "Buttons & Switches" um interruptor e colocá-lo no "Front Panel".
2º Seleccionar sobre o menu "LEDs" um indicador e colocá-lo ao lado do objecto anterior.

3º Alterar etiquetas dos objectos: fonte, tamanho, posição e cor.

2.0 "Olá Mundo..."

*4º Observe atentamente o "**Diagrama de Blocos**".*
*5º Posicione o ponteiro do "**rato**" em frente do botão ON/OFF e faça **click** com o botão esquerdo do mesmo.*
*6º Posicione o ponteiro do rato em frente do pequeno triângulo do bloco "Estado" e faça **click** com o mesmo botão.*
*7º Observe o aparecimento de uma **ligação** entre os dois blocos.*

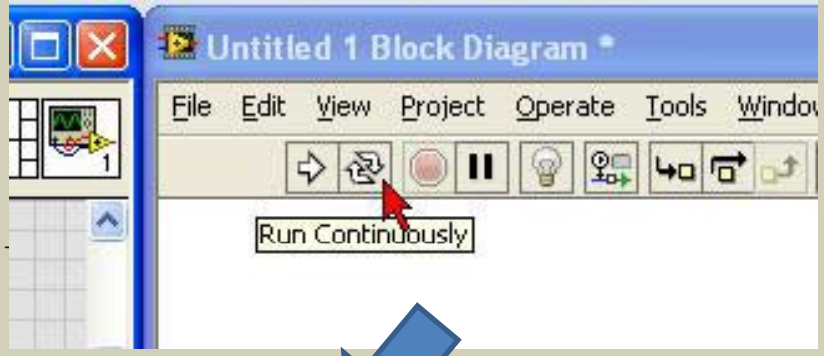


Ligações distinguidas por cores e perfís.
Essa distinção depende do tipo de dados e blocos que se estão a ligar. Por exemplo:

Verde	–	*Booleano*
Rosa	–	*String*
Azul	–	*Número Inteiro*
Laranja	–	*Número real*

2.0 "Olá Mundo..."

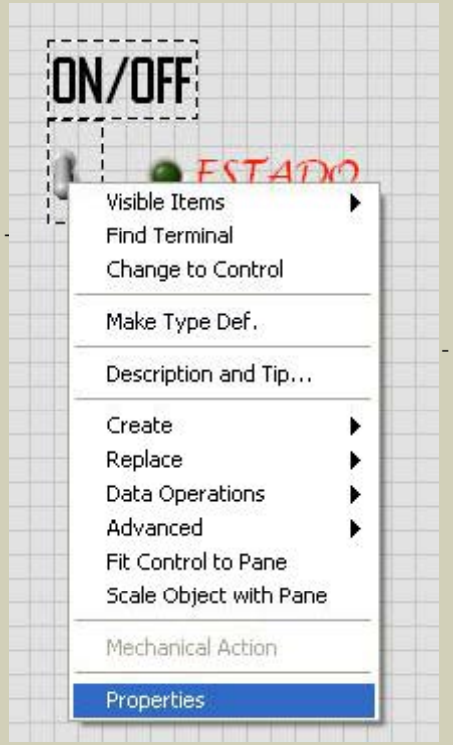
8º Execução e Teste...



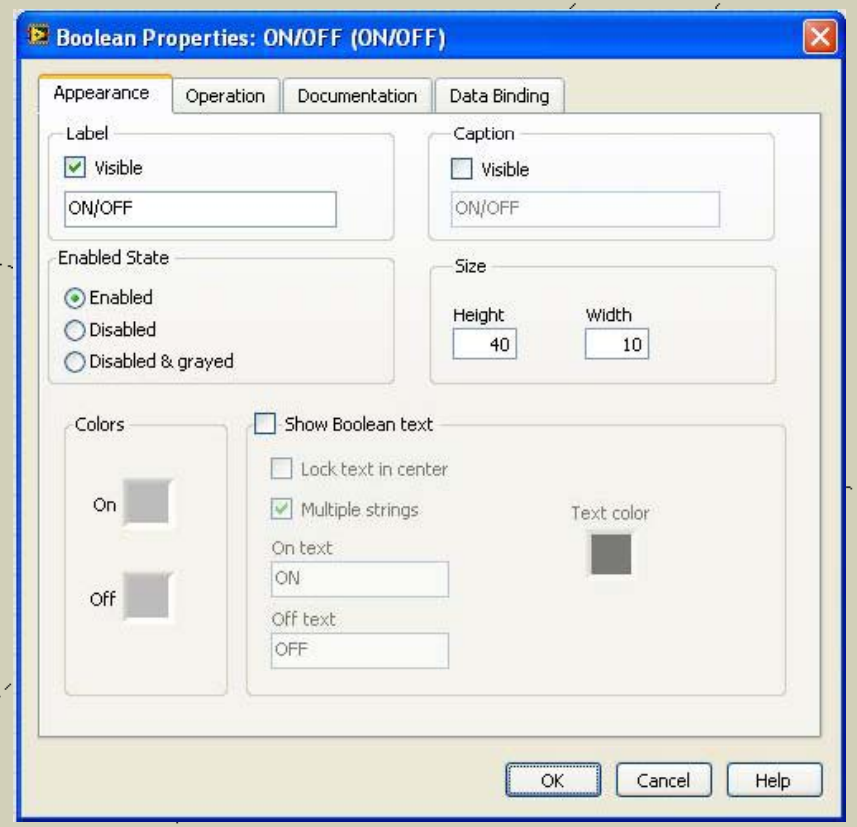
Para além disso:
Outras propriedades dos objectos...

2.1 Propriedades

1º click com o botão direito do rato sobre o objecto...

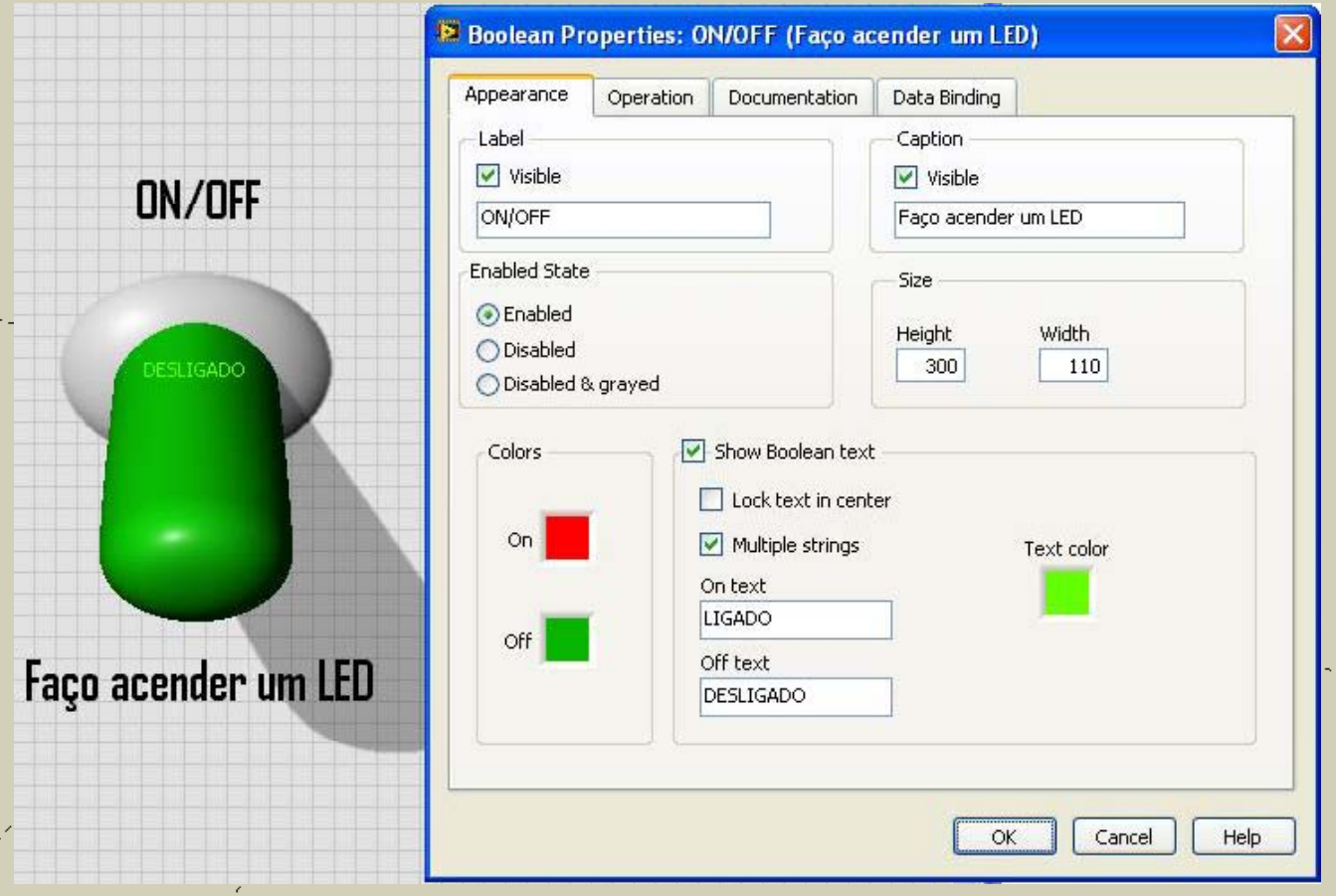


2º selecção da opção "Properties"



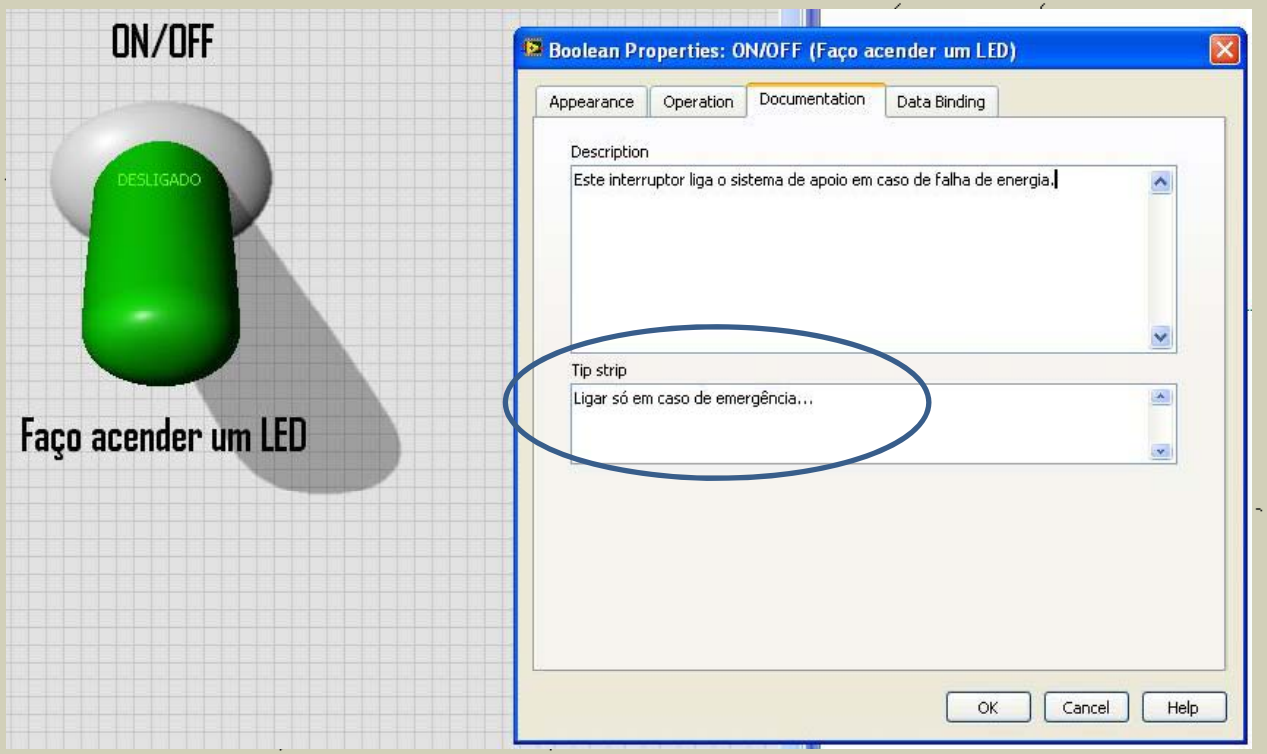
2.1 Propriedades

3º efectue as seguintes alterações e observe o comportamento do objecto...



2.1 Propriedades

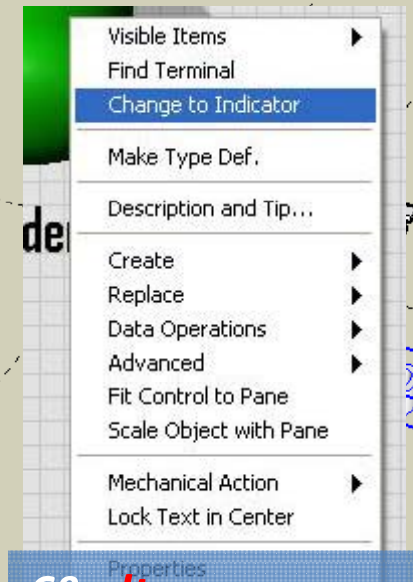
4º efectue as seguintes alterações e observe o comportamento do objecto...



2.1 Propriedades

5º efectue as seguintes alterações e observe o comportamento do objecto...

Alguns objectos podem comutar de papel CONTROL/INDICATOR ou INDICATOR/CONTROL



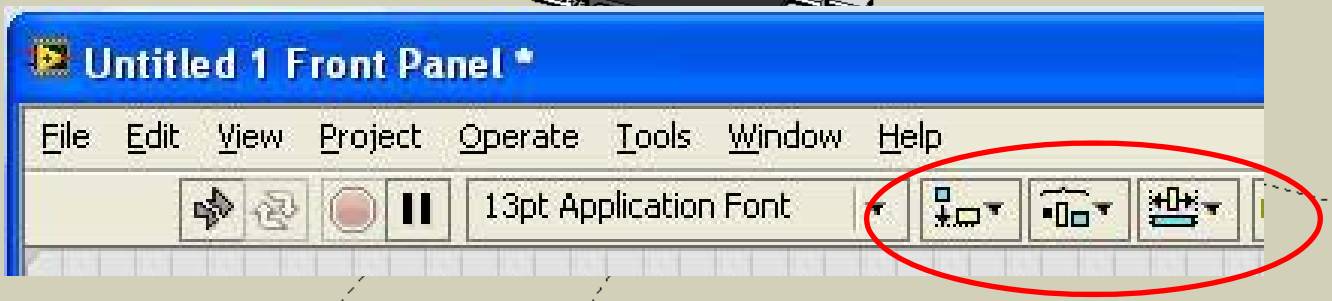
6º altere os comportamentos do interruptor e LED. Execute a simulação e conclua...

2.1 Exercícios

EX1: *Faça com que o LED fique com a cor "preto" quando inactivo e "vermelho" quando activo*

EX2: *Adicione outro LED ao "Front Panel". Faça com que o mesmo interruptor ligue ambos os LEDs*

EX3: *Faça uma grelha (matriz) de 25 LED (cinco linhas com cinco LEDs cada)*

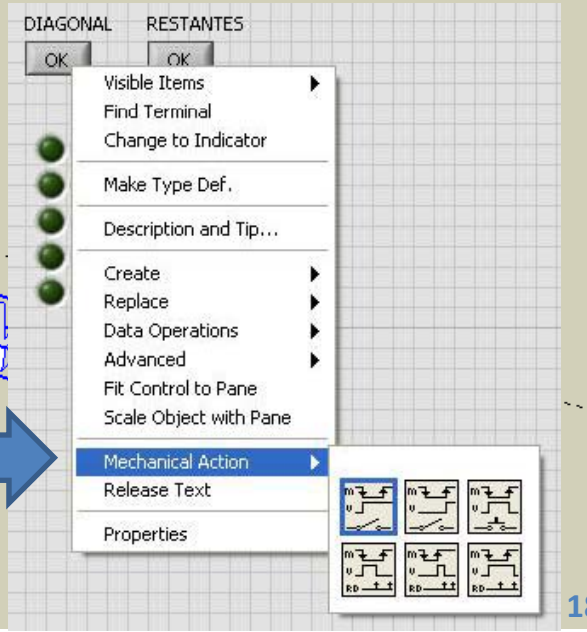


Utilize as ferramentas de alinhamento

2.1 Exercícios

EX4: *Coloque dois botões associados à matriz anterior conforme se mostra na imagem à direita.*

*A activação do botão da esquerda liga apenas os LED na diagonal. O botão da esquerda liga todos os outros. O botão da esquerda apenas permanece activo enquanto o utilizador o mantiver premido. (Utilize as opções em "**Mechanical Action**" para atribuir essas características aos botões)*



2.2 Operações Lógicas

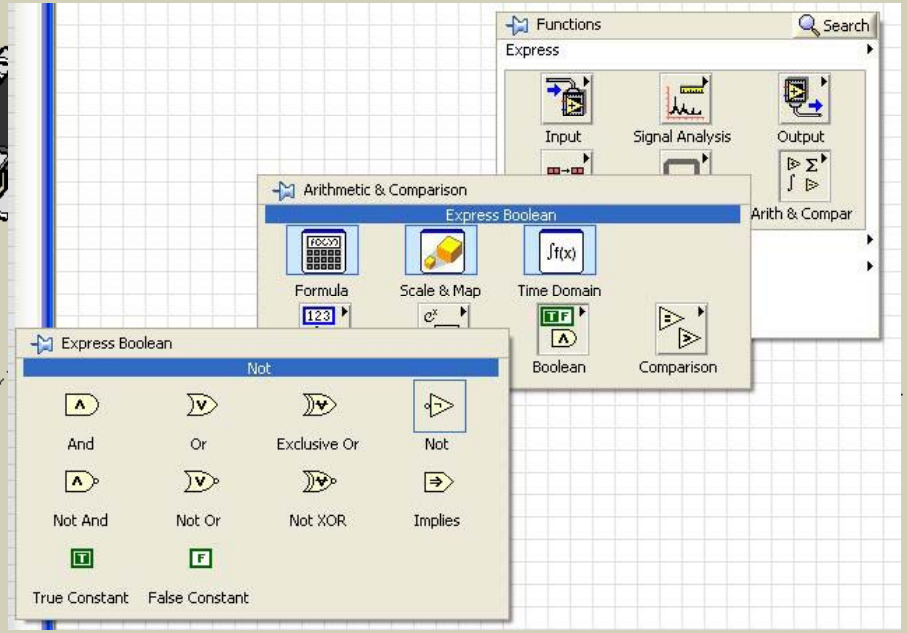
Operações Lógicas Elementares

NEGAÇÃO (NOT)
CONJUNÇÃO (AND)
DISJUNÇÃO (OR)

NEGAÇÃO

"click" com o botão direito do rato sobre a janela "**Block Diagram**" e seleccionar:

ARITH&COMP\BOOLEAN\NOT



2.2 Operações Lógicas

Exemplo: ***Um Interruptor deve comandar o estado de dois LEDs. Quando um LED estiver ligado o outro deve estar desligado e vice-versa.***

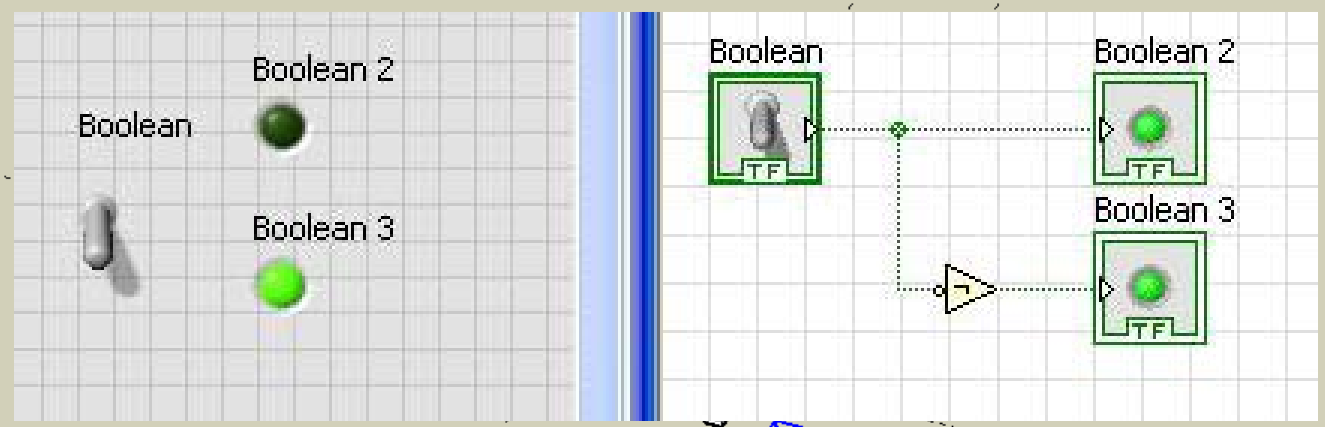


TABELA VERDADES:

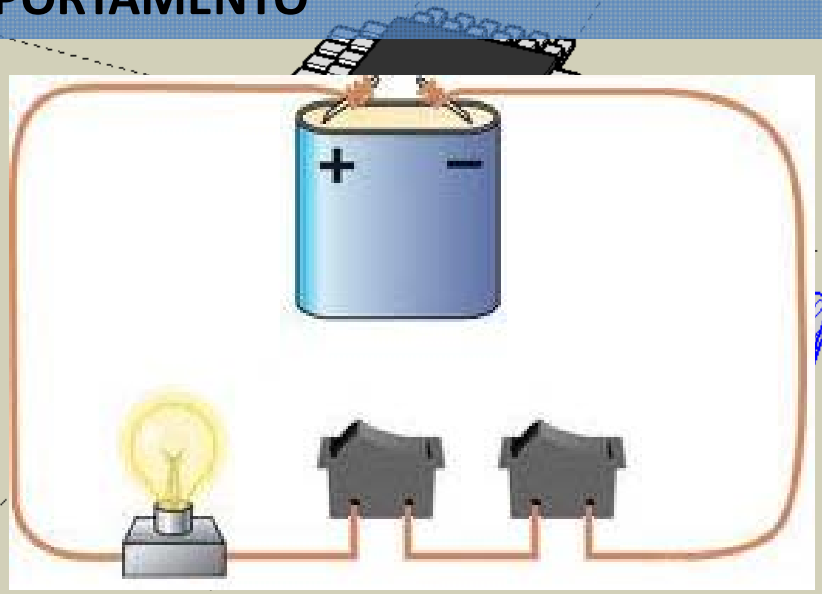
ENTRADA	SAÍDA
VERDADE (1)	FALSO (0)
FALSO (0)	VERDADE (1)

2.2 Operações Lógicas

CONJUNÇÃO (E)

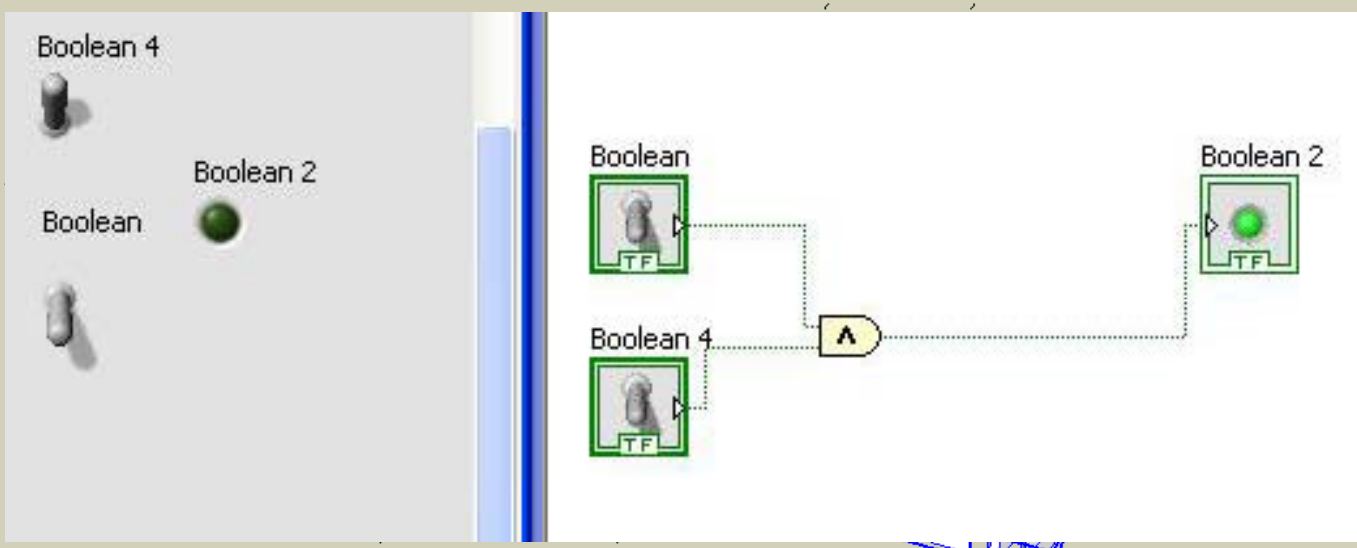
"click" com o botão direito do rato sobre a janela "**Block Diagram**" e seleccionar:
ARITH&COMP\BOOLEAN\AND

MODELO DE COMPORTAMENTO



2.2 Operações Lógicas

Exemplo: *Um LED apenas deve ligar quando dois interruptores se encontrarem fechados.*



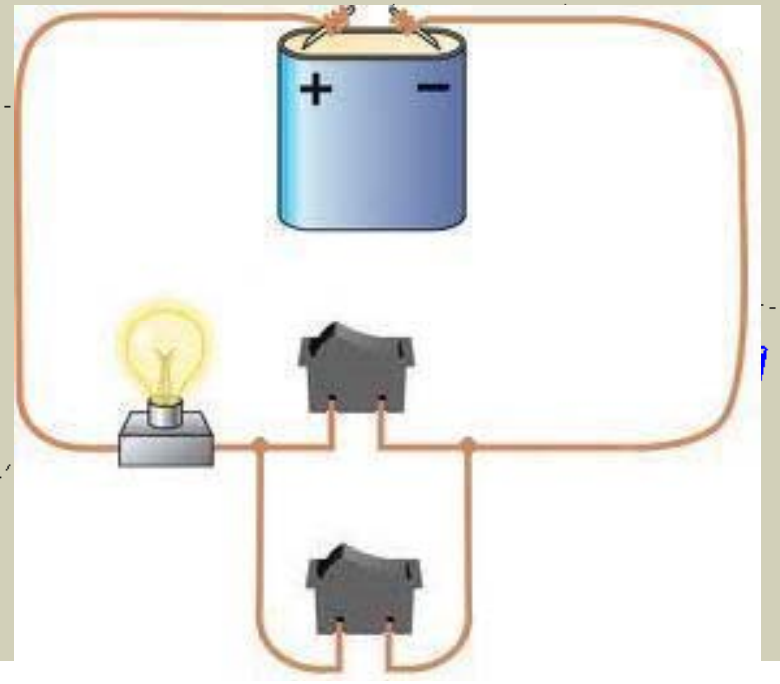
Exercício: *Um LED deve ligar quando dois interruptores estiverem ligados e um terceiro desligado.*

2.2 Operações Lógicas

DISJUNÇÃO (OU)

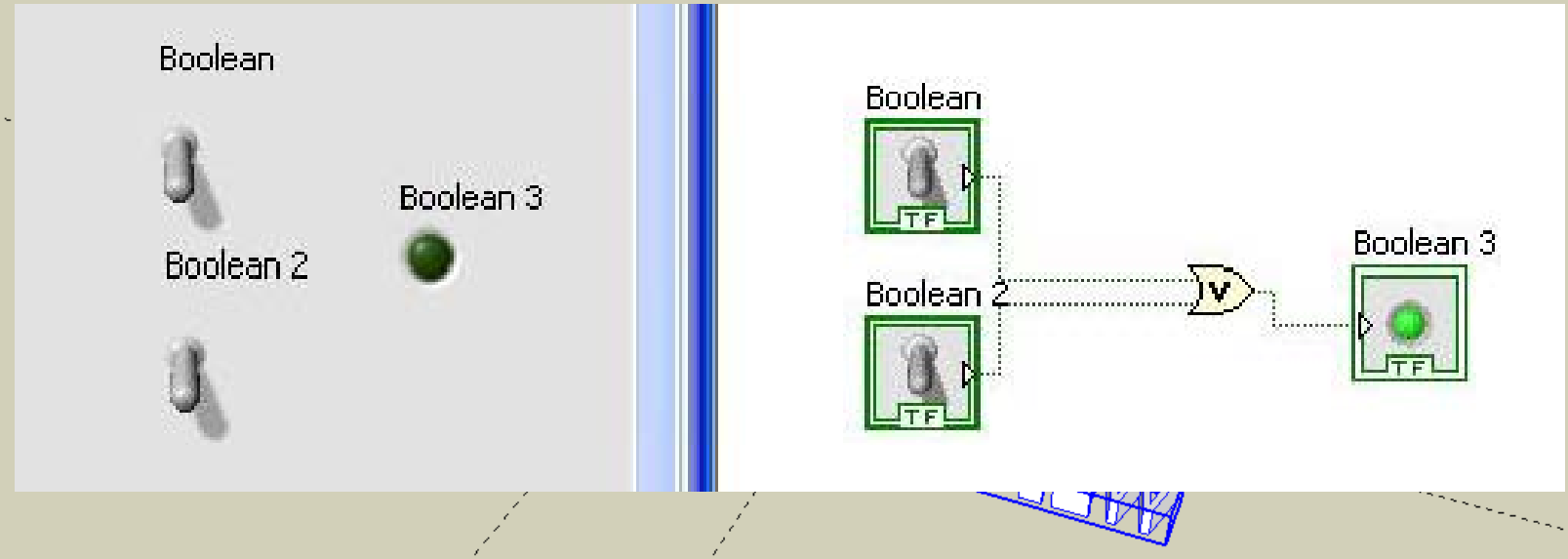
"click" com o botão direito do rato sobre a janela "**Block Diagram**" e seleccionar:
ARITH&COMP\BOOLEAN\OR

MODELO DE COMPORTAMENTO



2.2 Operações Lógicas

Exemplo: *Um LED deve ligar quando um ou ambos os interruptores se encontrarem fechados.*



Exercício: *Um LED deve ligar quando dois interruptores estiverem ligados OU quando um terceiro se encontrar desligado.*

2.2 Operações Lógicas

EX5: ***Obtenha as tabelas de verdades das operações lógicas AND e OR***

BOTÃO 1	BOTÃO 2	LED
F	F	?
F	V	?
V	F	?
V	V	?

EX6: ***Desenhe um diagrama de blocos de modo que o LED apenas ligue quando o estado dos botões for diferente.***

EX7: ***Controlo de uma bomba de água...***

"Um depósito de água é alimentado por uma bomba que tira a água de um poço. O depósito serve para rega e abastecimento de água a uma casa de habitação. Pretende-se que a bomba só entre em funcionamento quando se tira água para rega, e simultaneamente, para a casa de habitação, ou quando a água do depósito não estiver acima de um determinado nível"

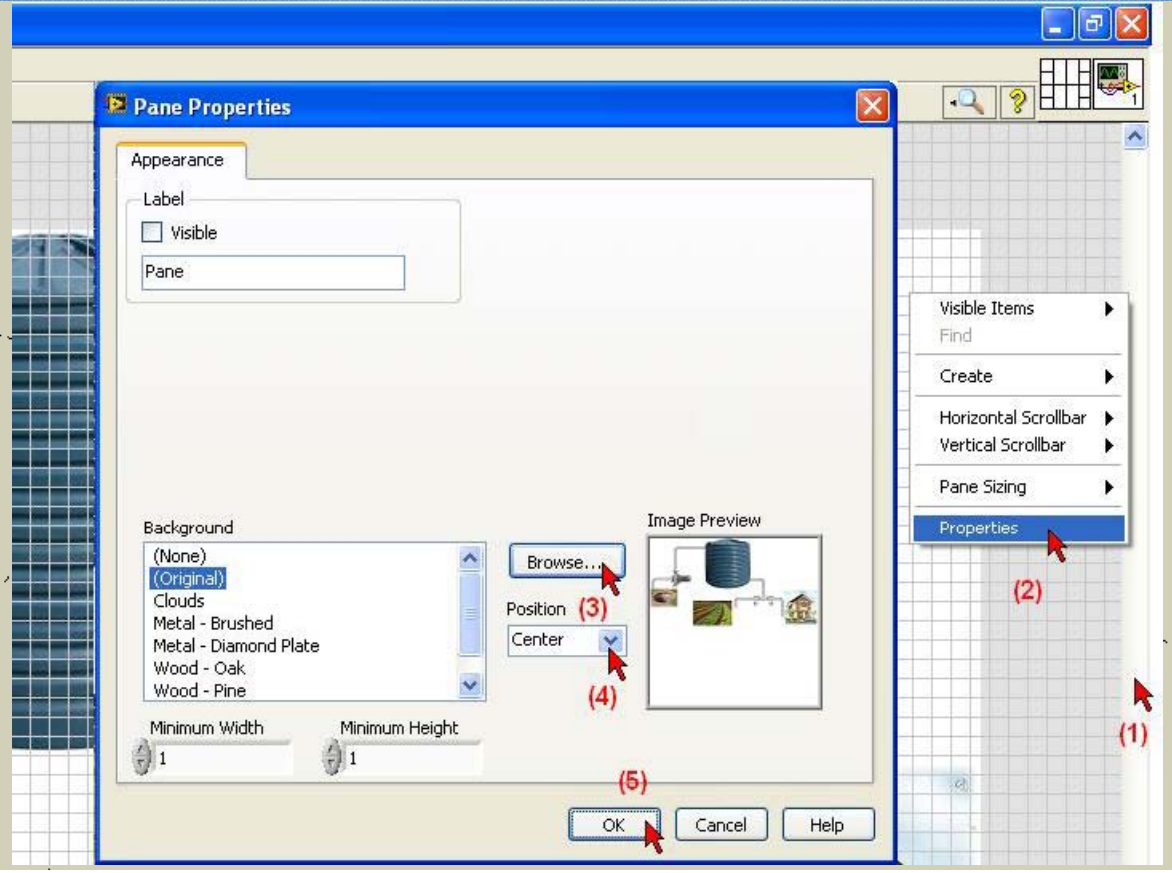
2.2 Operações Lógicas

Procedimento de resolução:

1º Coloque a imagem (disponível em www) como pano de fundo

Como?!

(1) Click com o botão direito do rato na barra de scroll
(2) Seleccionar "Properties" no pop-up menu
(3) Seleccionar "Browse" e escolher a imagem
(4) Seleccionar "Center" da pop-up box
(5) Confirmar em "OK"



2.2 Operações Lógicas

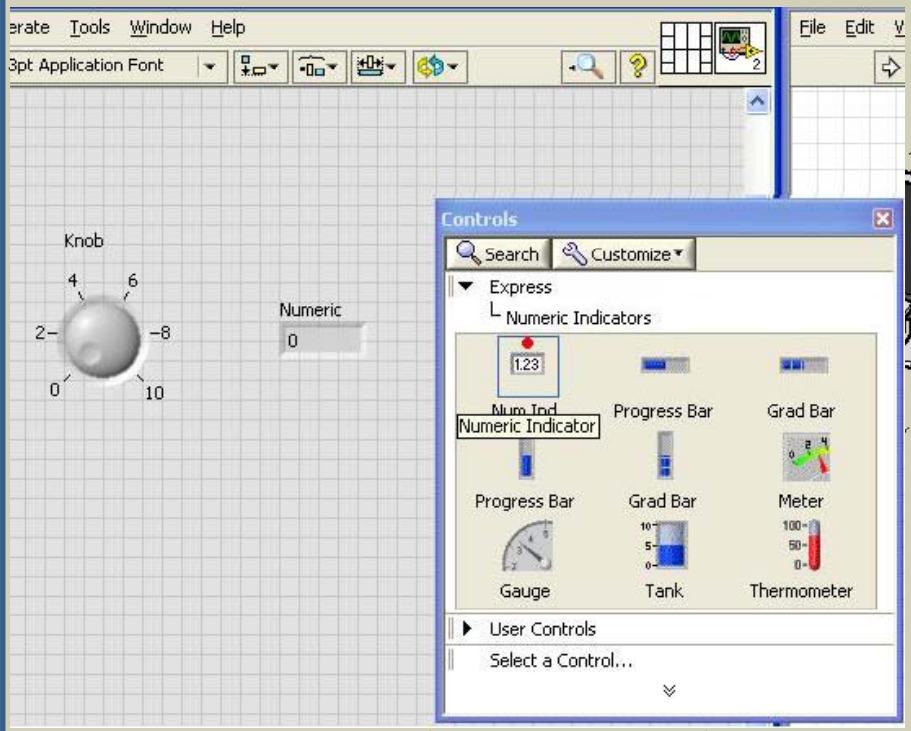
*2º Utilize um LED (**vermelho**) para assinala a activação da bomba, LED (**verdes**) para assinalar a demanda de água por parte da habitação e da rega e LED (**azul**) para o indicador de nível de água no tanque.*

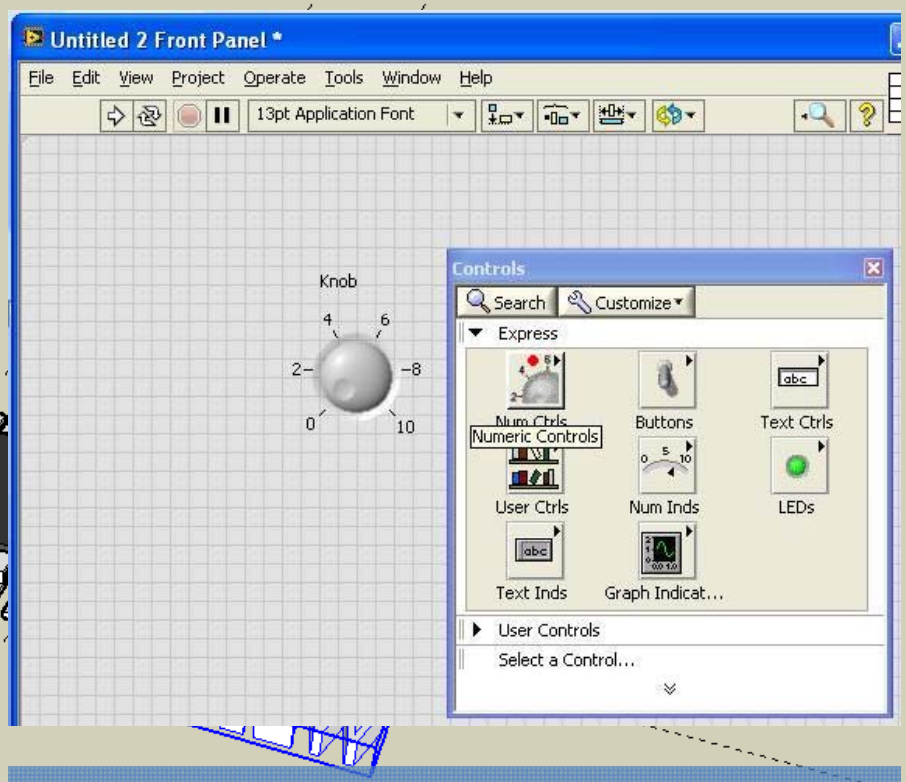
3º Utilize interruptores para simular o indicador de nível do tanque e a demanda de água pela habitação e rega.

E já está... O vosso primeiro protótipo de um ambiente SCADA

2.3 Controlos numéricos

1º Seleccionar sobre o menu "Numeric Controls" o objecto "Knob"

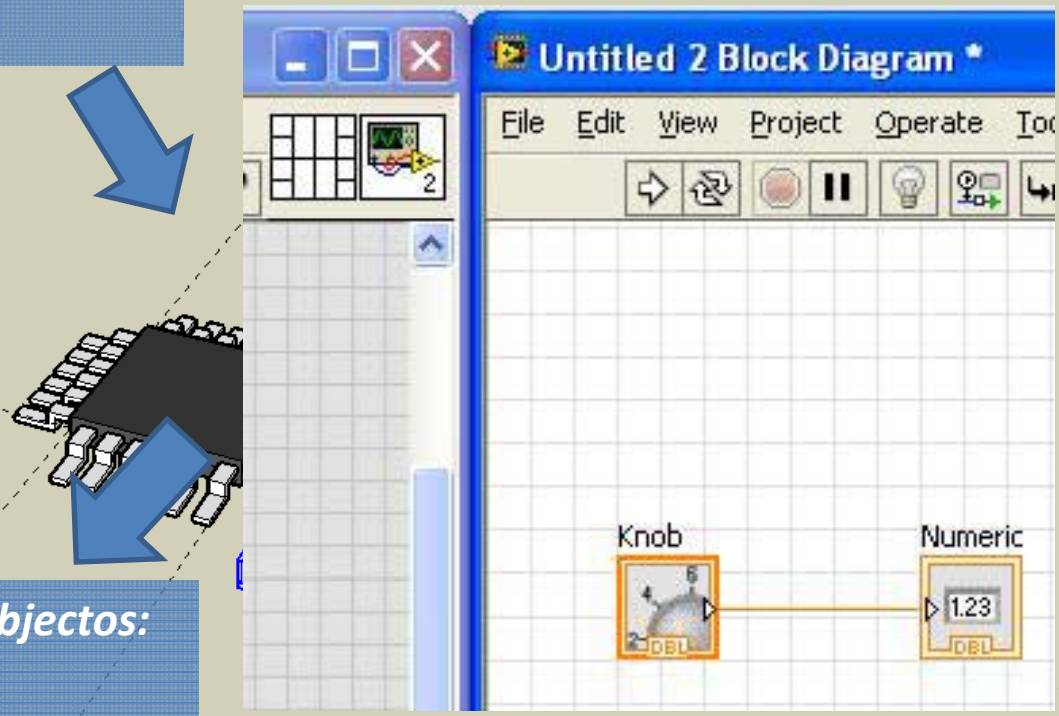




2º Seleccionar sobre o menu "Numeric Indicators" o objecto "numeric indicator"

2.3 Controlos numéricos

3º Efectuar a seguinte ligação no diagrama de blocos e testar ...

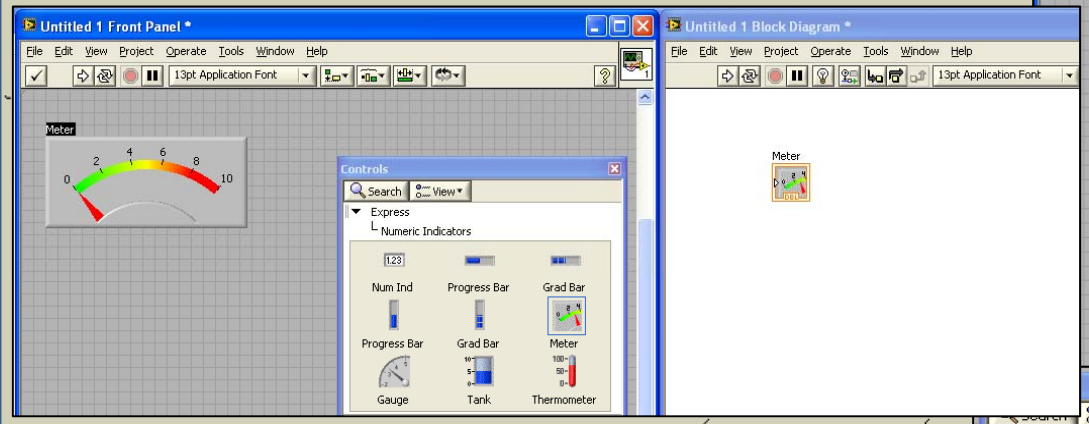


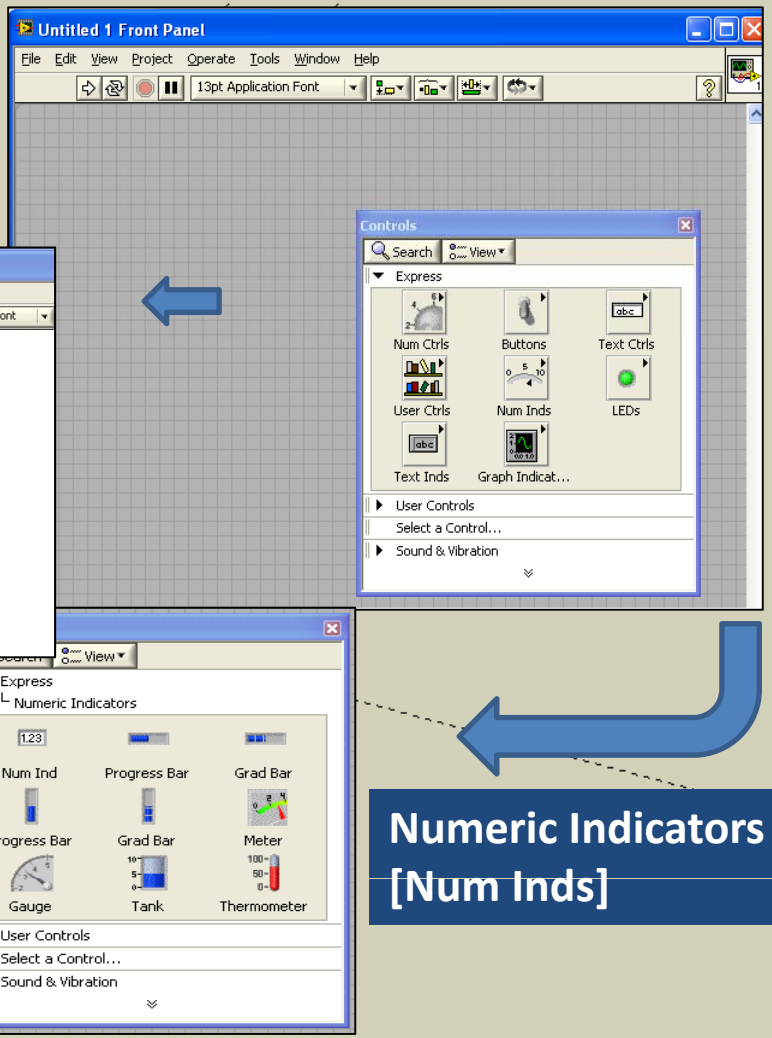
4º Explorar as propriedades dos objectos:

- *Escalas*
- *Aspecto*
- *Etiquetas*
- *etc.*

2.3 Controlos numéricos

"Botão" direito do "Rato" sobre painel frontal abre lista de CONTROLOS





Arrastar o objecto e posicioná-lo no painel frontal.
Observe o objecto que é criado na janela do diagrama de blocos.

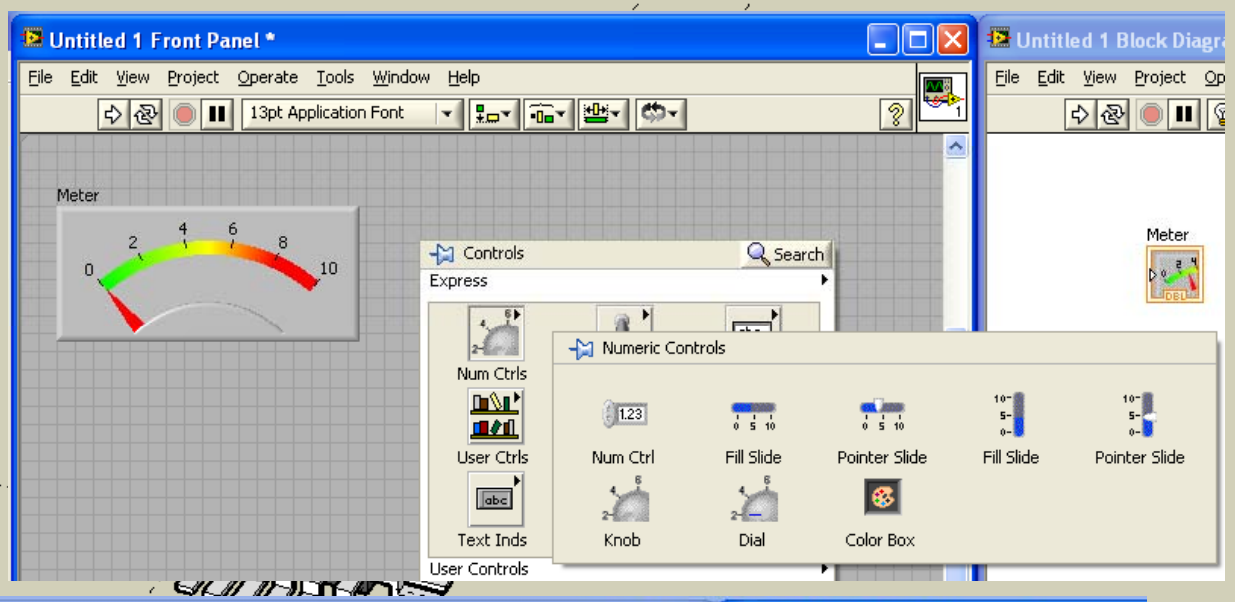
[Meter]

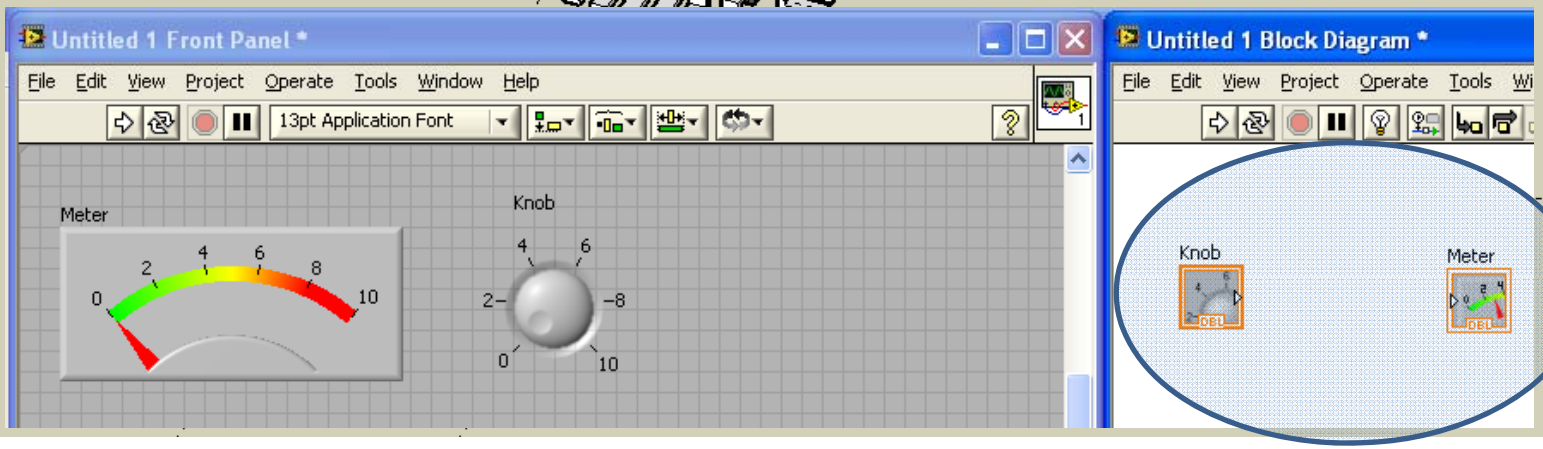
Numeric Indicators [Num Inds]

3.1 Controlos numéricos

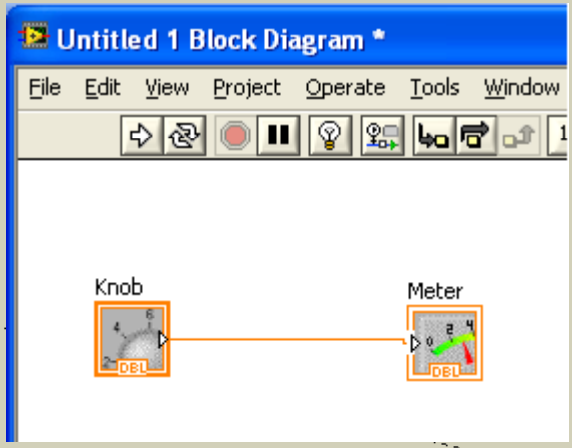
"Botão" direito do "Rato" sobre painel frontal

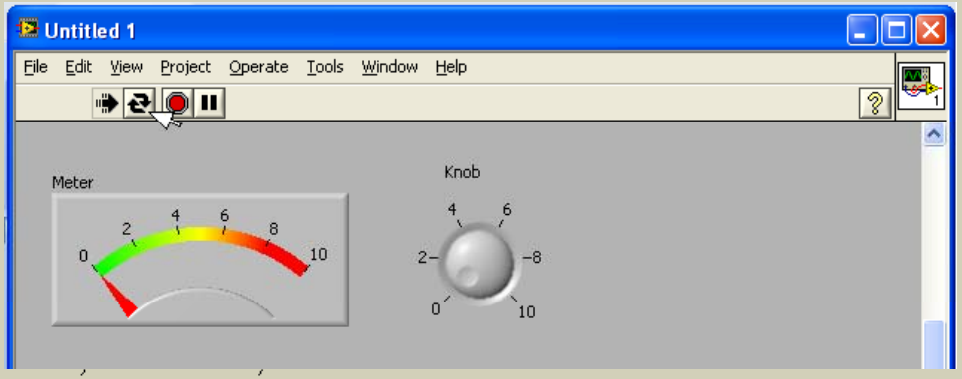
[Knob]

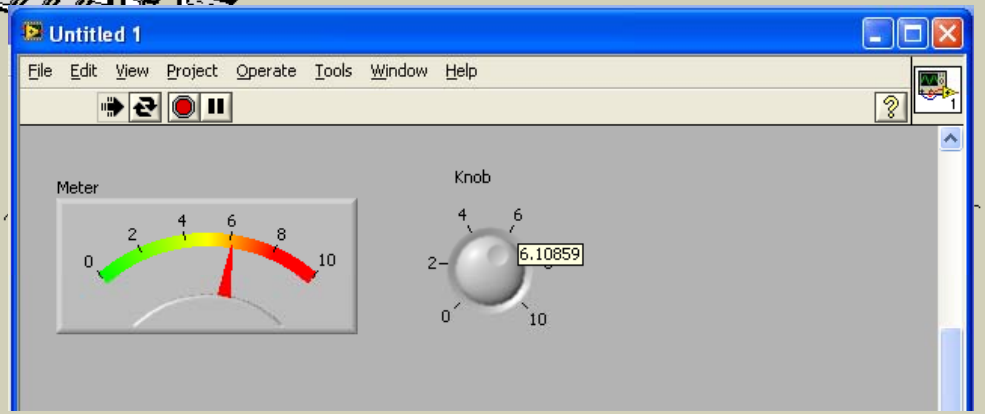




2.3 Controlos numéricos





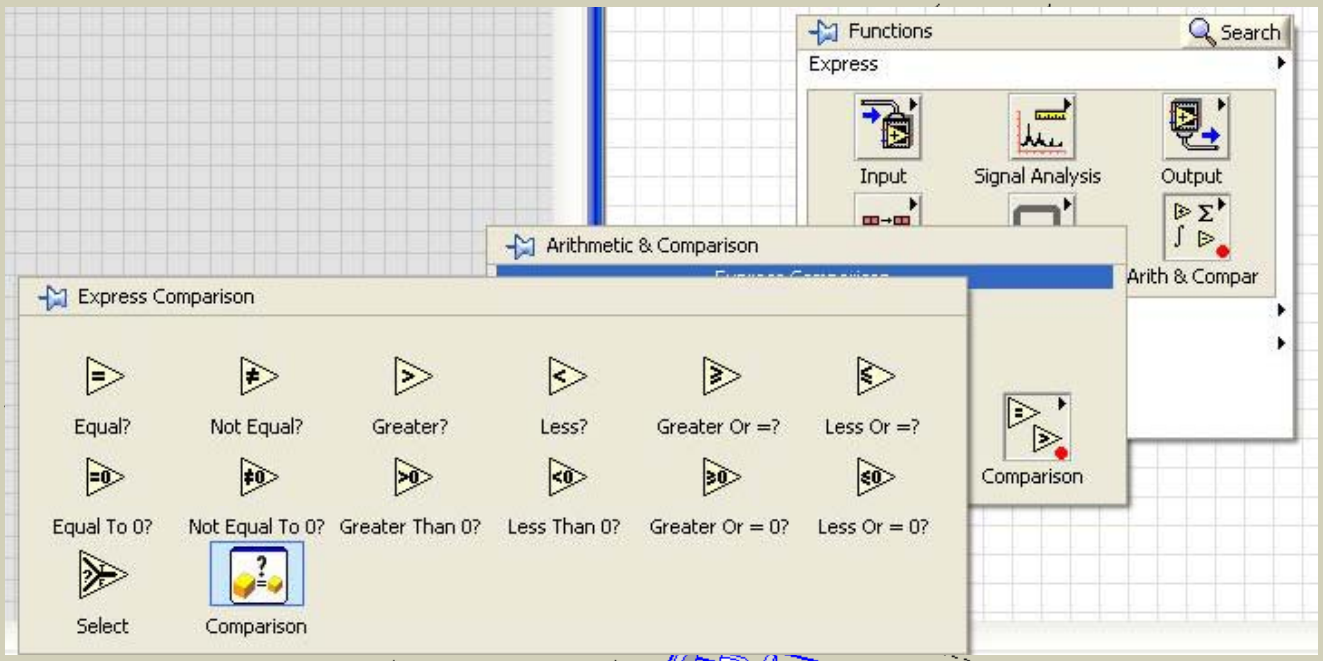


1 – Executar a ligação entre os dois blocos na janela " Block Diagram"

2 – Executar Programa [Run Continuously]

3 – Com ajuda do "Rato" girar o botão rotativo "Knob" e observar o movimento do painel de medida.

2.4 Operações de Comparação

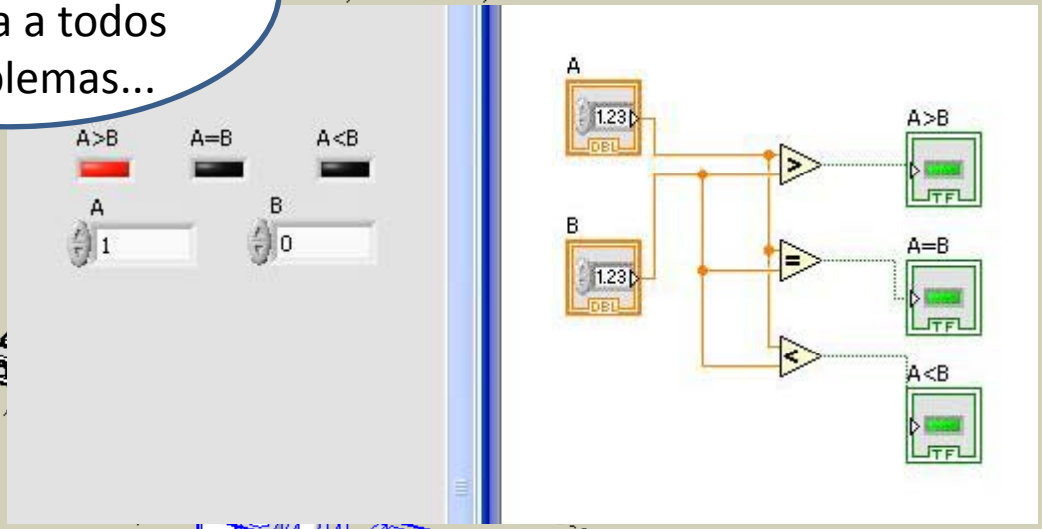


Traduz num valor lógico o resultado de comparação de dois valores numéricos.

EXEMPLO: O utilizador introduz dois valores, A e B. Se A for maior que B um LED vermelho é aceso, se A for menor que B um LED verde é aceso. Caso contrário um LED azul é aceso.

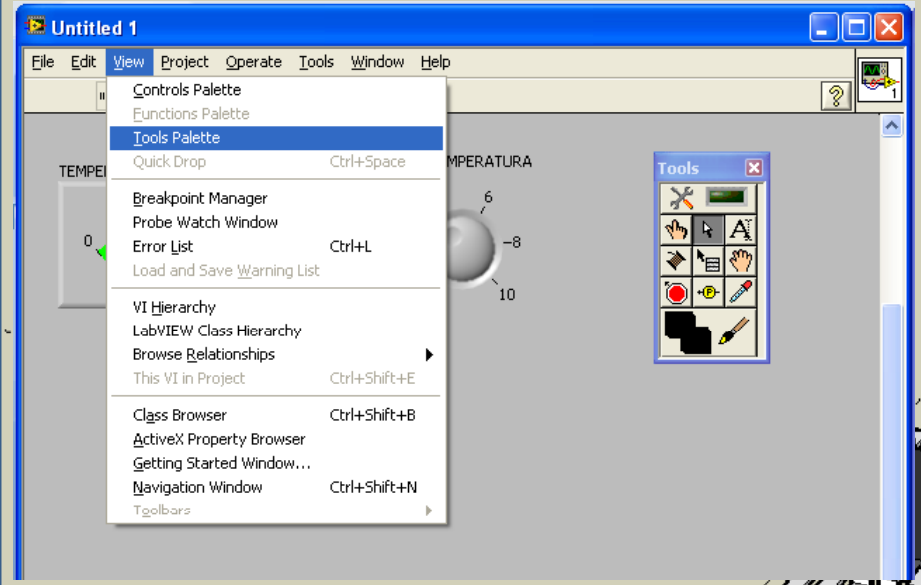
2.4 Operações de Comparação

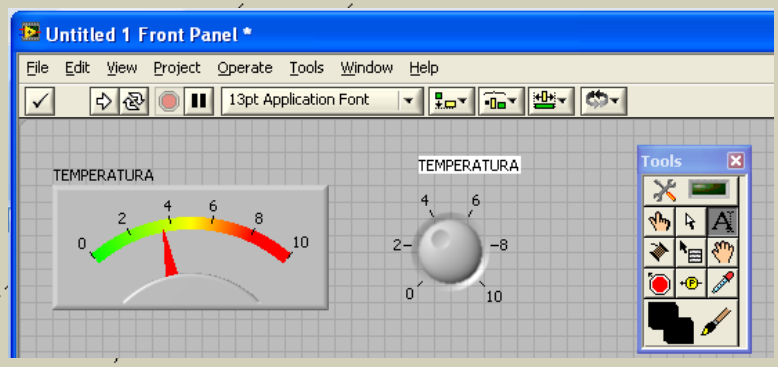
Penso que o LabView é a resposta a todos os problemas...



EXEMPLO: Alarme de Temperatura – Um LED deve acender se a temperatura aumentar acima de um valor pré-definido (SET-POINT). Esse limite deve ser definido por um objecto do tipo "pointer slider" e a temperatura do processo é simulada por um "knob"

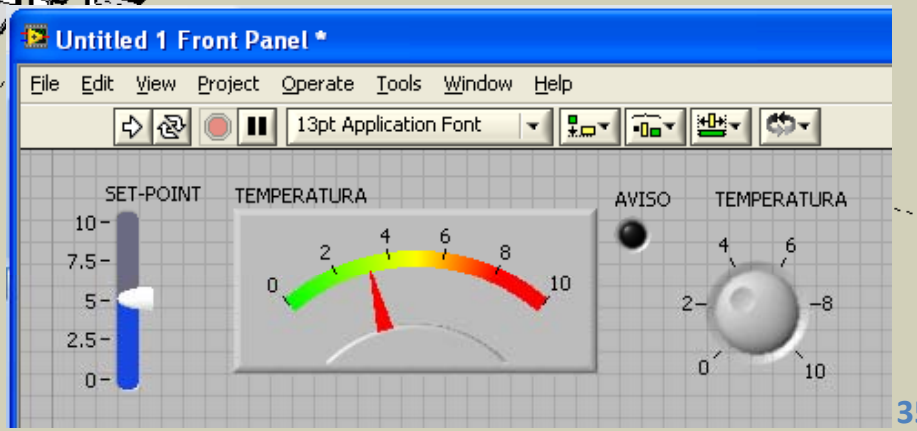
2.4 Operações de Comparação





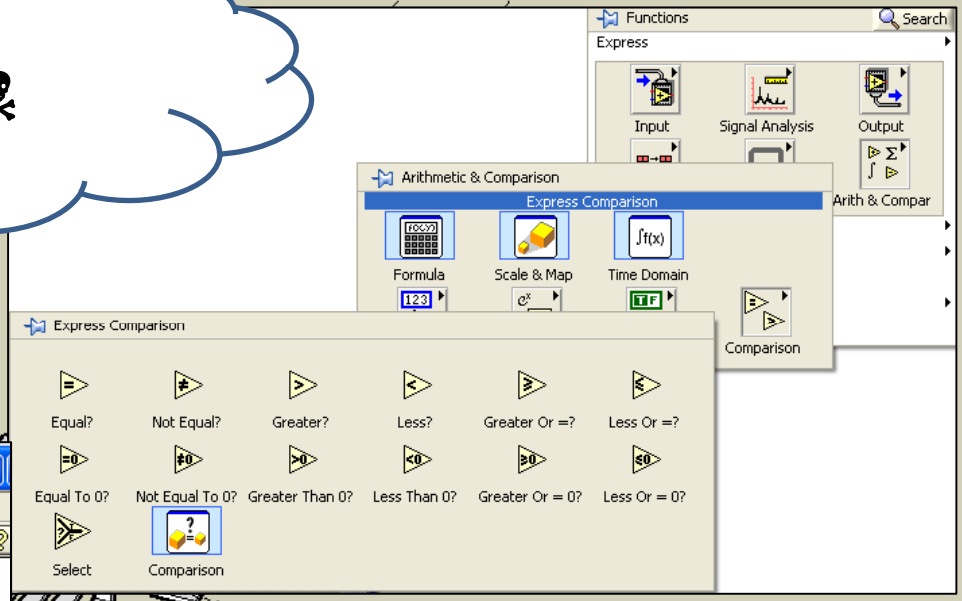
1 – Abrir caixa de ferramentas
2 – Seleccionar [Edit Text]
3 – Alterar o nome dos controlos.

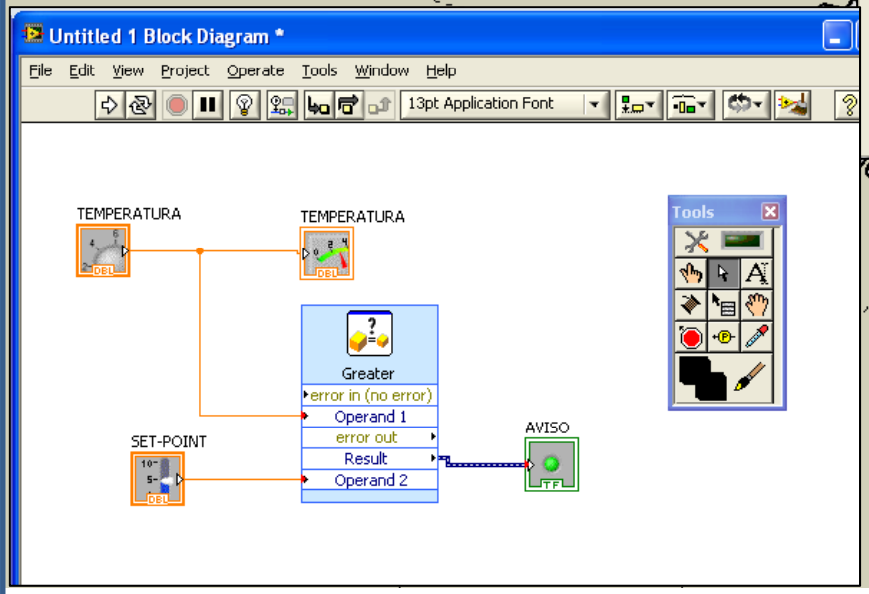
4 – Inserir "Pointer SLIDER"
2 – Inserir "LED" [Round LED]
3 – Alterar cor do LED:
Apagado = Preto
Aceso = Vermelho



2.4 Operações de Comparação

?!☺☠



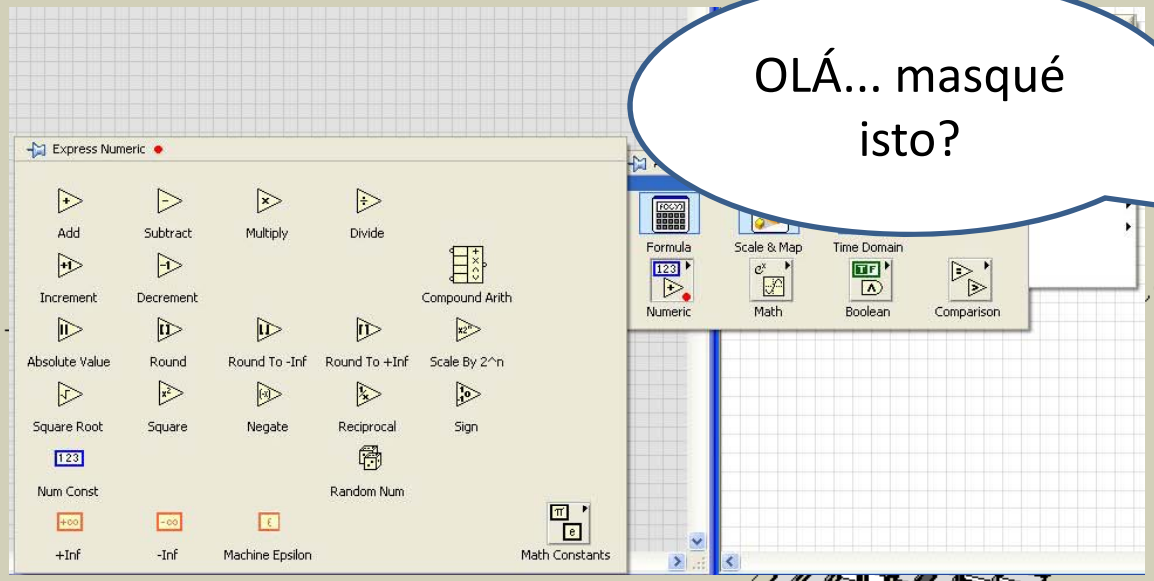


1 – Inserir COMPARISION no diagrama de blocos

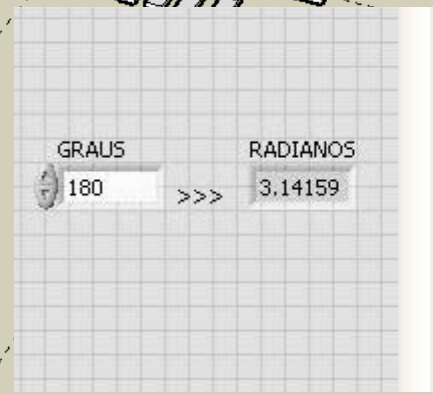
2 – Efectuar as seguintes ligações

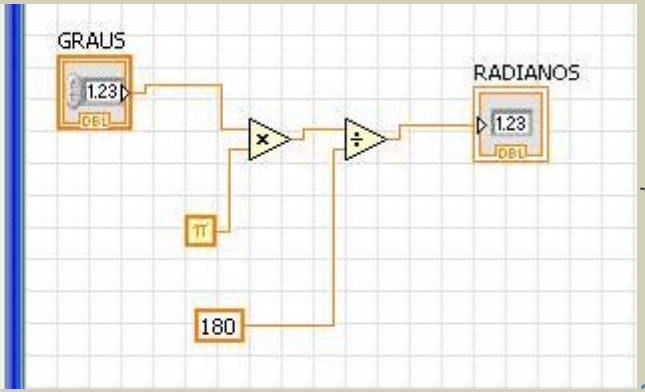
Nota: É necessário editar as propriedades do bloco COMPARISION de modo a admitir duas variáveis de entrada…

2.4 Operações Aritméticas



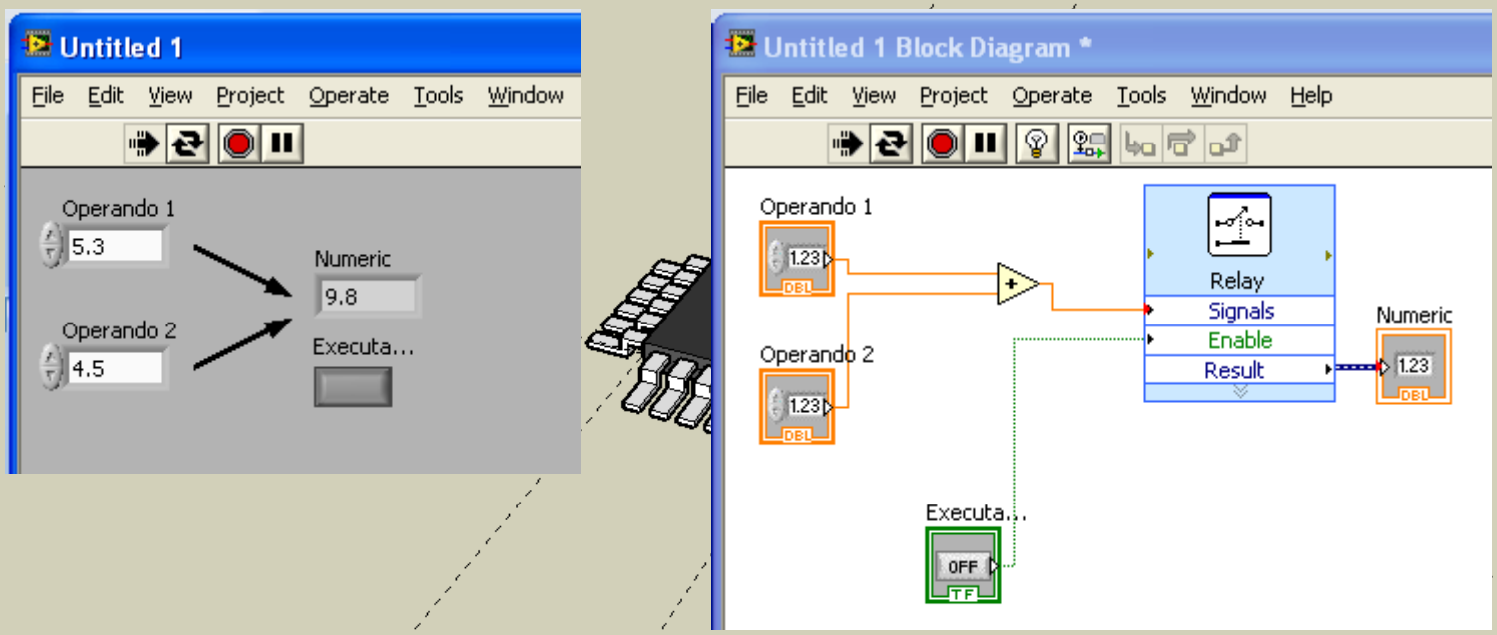
EXEMPLO: Conversão de um ângulo de graus para radianos.





2.4 Operações Aritméticas

- Máquina de somar simples



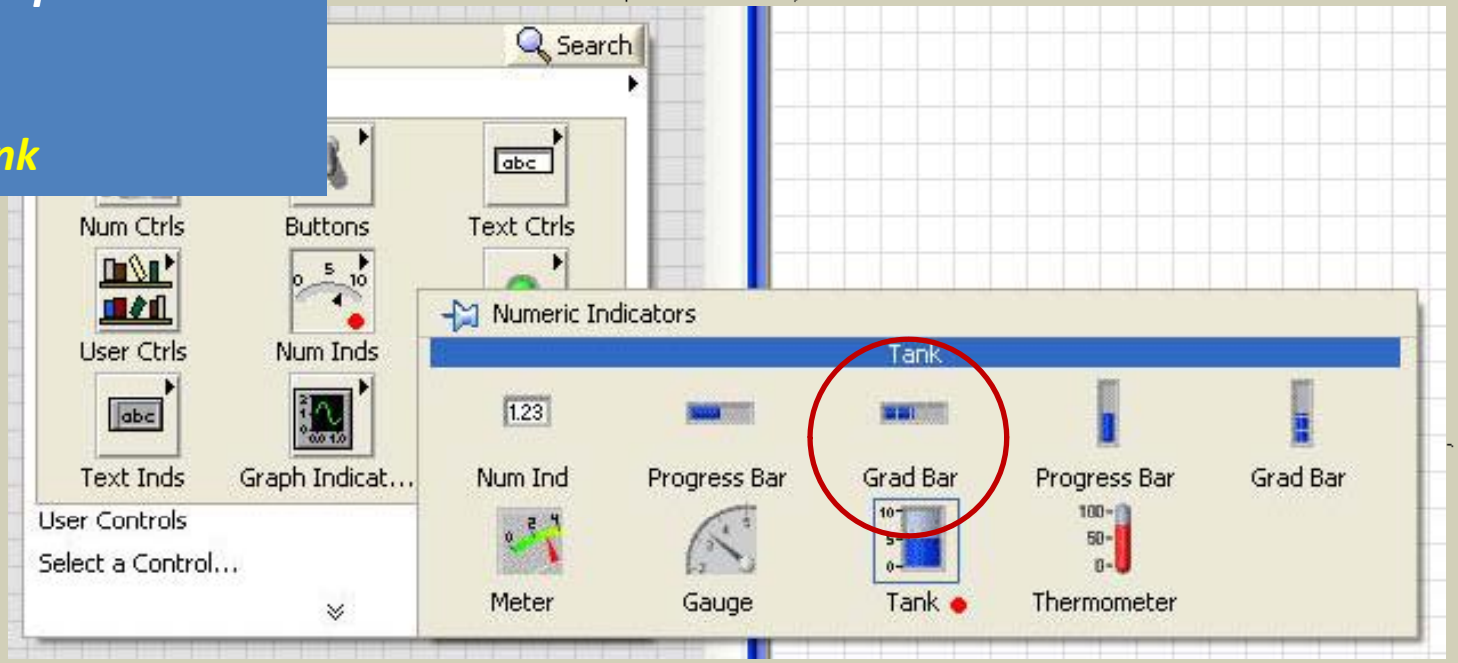
- A operação deve realizar-se sempre que o botão “Executa” é premido.

2.4 Operações Aritméticas

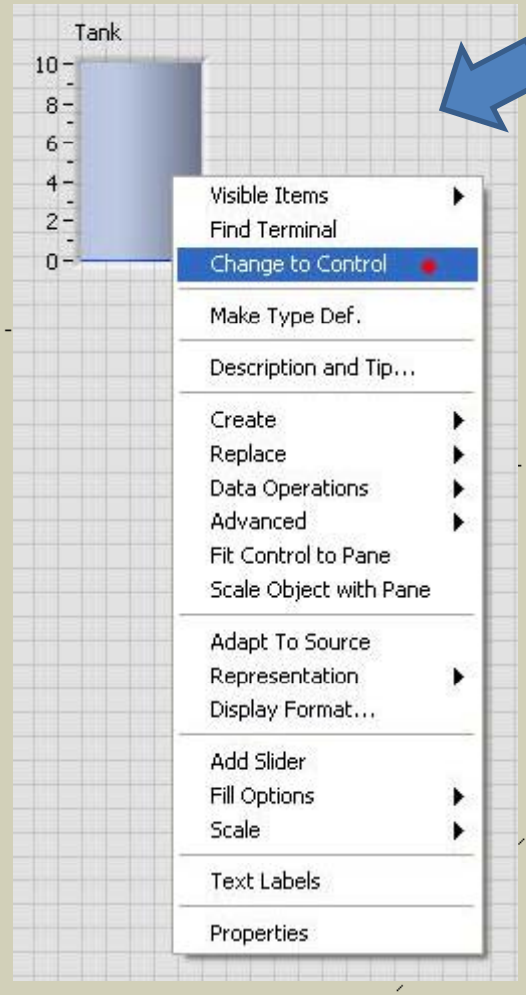
EXERCÍCIO: Cálculo do volume do fluído num tanque cilíndrico.
- O utilizador estabelece os parâmetros geométricos do tanque : altura e diâmetro
- Indicação de sobrecarga.

1 – Colocar o modelo de um tanque no *front panel*

Num Inds > ***Tank***

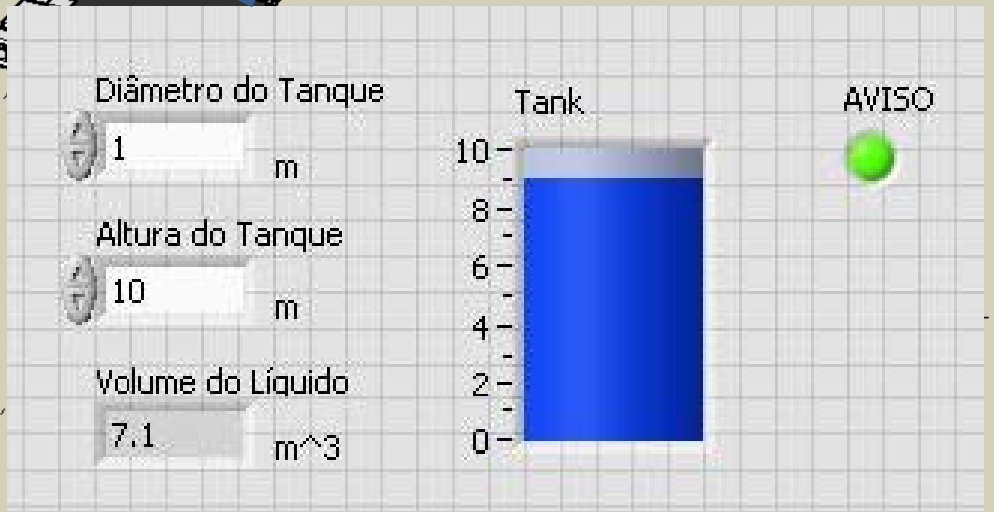


2.4 Operações Aritméticas



2 – Alterar o comportamento do tanque de Indicador para Controlo.

3 – Completar o resto da interface gráfica com os seguintes objectos...



2.4 Operações Aritméticas

4 – As seguintes condições devem verificar-se:

i) O diâmetro e altura do tanque são definidos pelo utilizador;

ii) Admite-se que o valor da altura do líquido é fornecido por um sensor;

iii) Se o nível do fluído estiver acima de ¾ da altura do tanque o LED deve acender

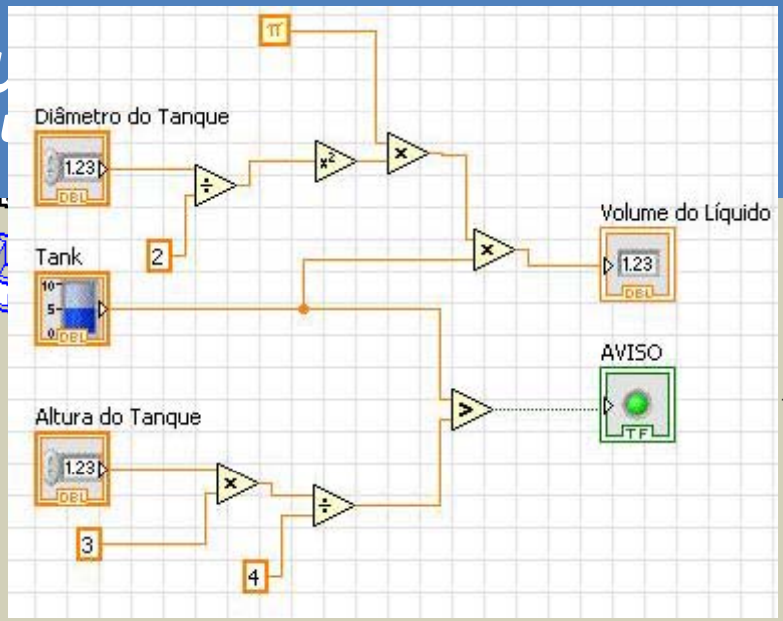
iv) O valor do volume do fluído deve ser apresentado no indicador numérico.

NOTA:

Volume do líquido = Altu

Onde r se refere ao raio do tanq

Solução ...



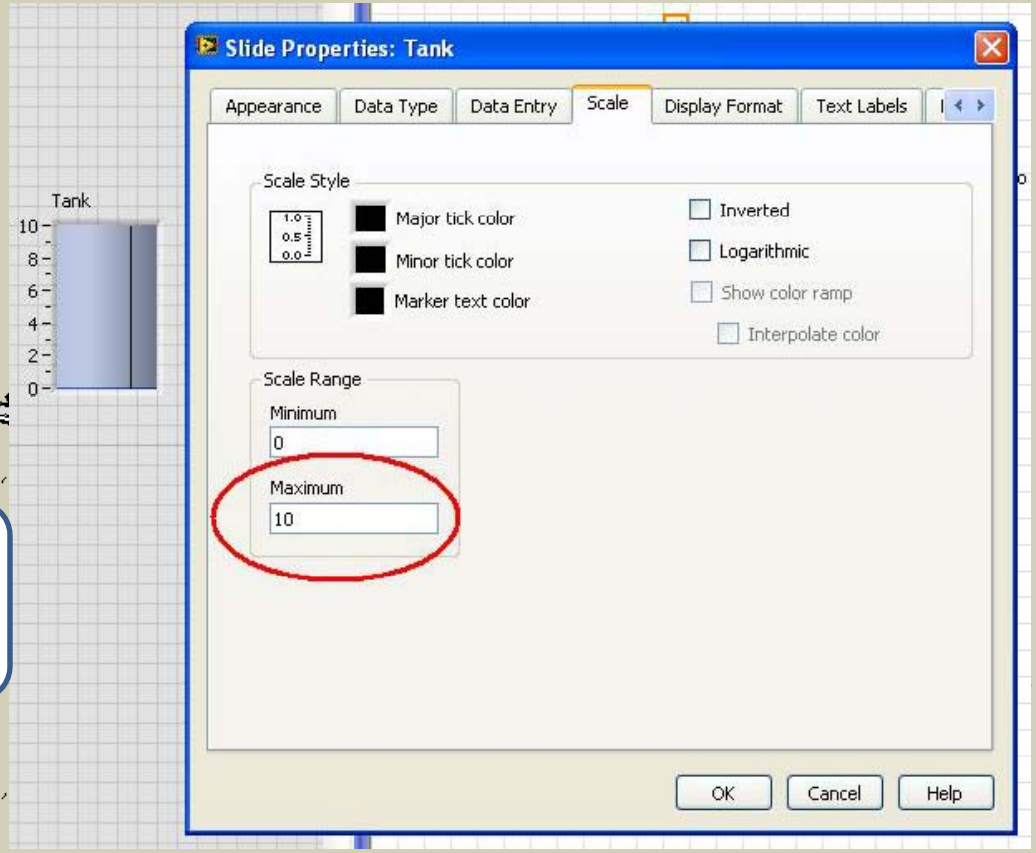
2.4 Operações Aritméticas

5 – A altura do líquido nunca pode ser superior à altura do tanque

É necessário alterar o valor máximo do tanque em função do valor da altura do tanque introduzido pelo utilizador.

COMO FAZER ISSO?

Já estou a ficar baralhada!

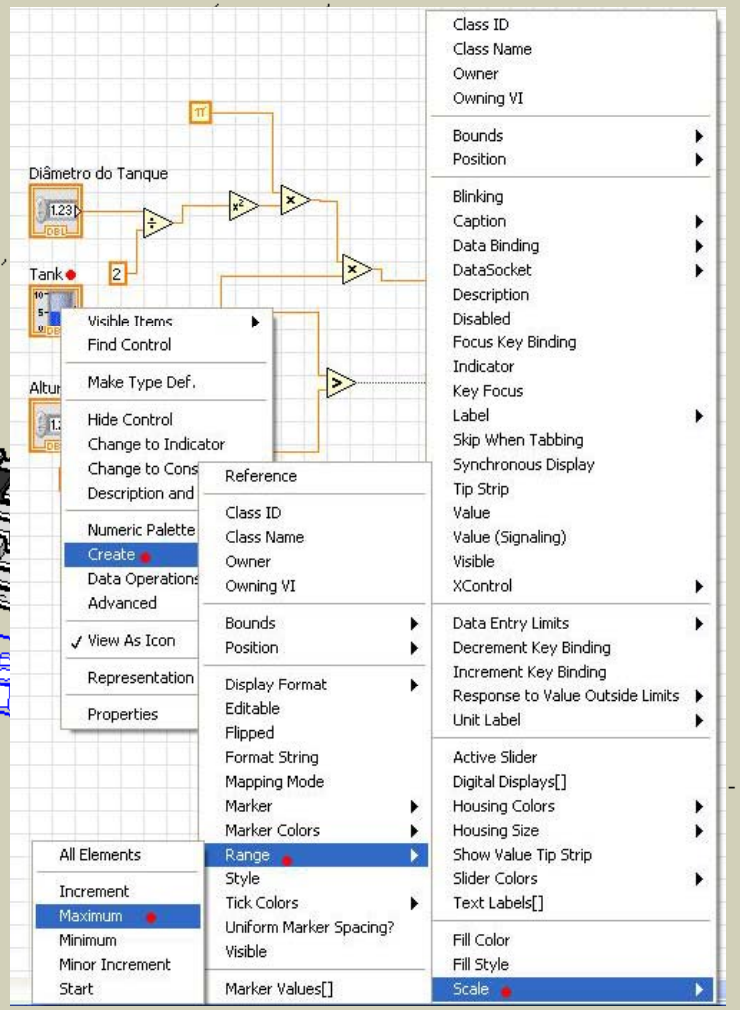


2.4 Operações Aritméticas

Qualquer propriedade de qualquer objecto pode ser alterada dinamicamente. (enquanto o programa está a ser executado).

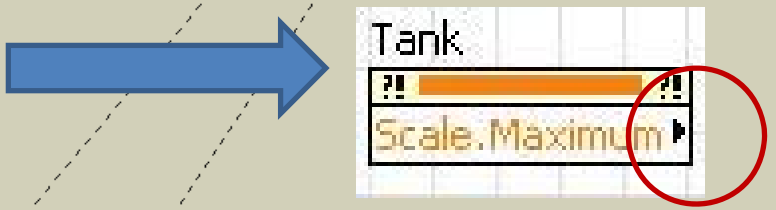
"click" com o botão direito sobre o objecto cujo comportamento se pretende alterar (neste caso o bloco Tank no *Block Diagram*)

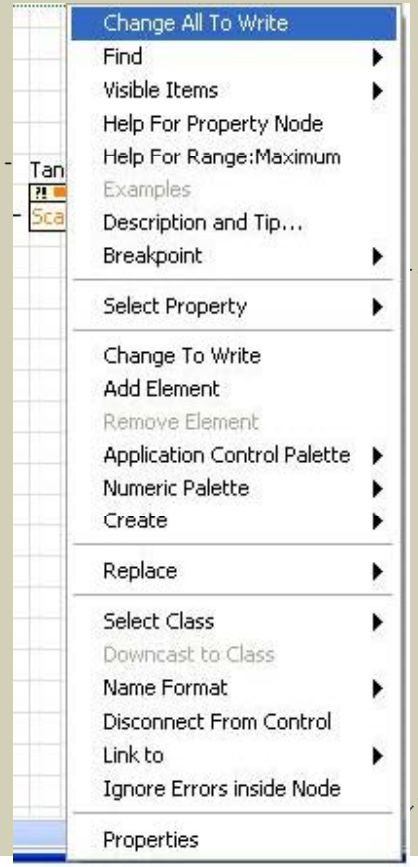
No "pop-up" menu seleccionar:
CREATE> PROPERTY NODE> SCALE> RANGE> MAXIMUM



2.4 Operações Aritméticas

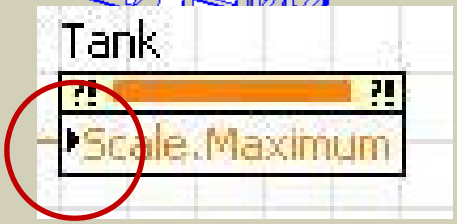
Um novo objecto é criado no Block Diagram.
Por defeito aparece no modo READ (LEITURA)





Como pretendemos alterar o valor desse objecto é necessário alterar o modo para WRITE (ESCRITA)

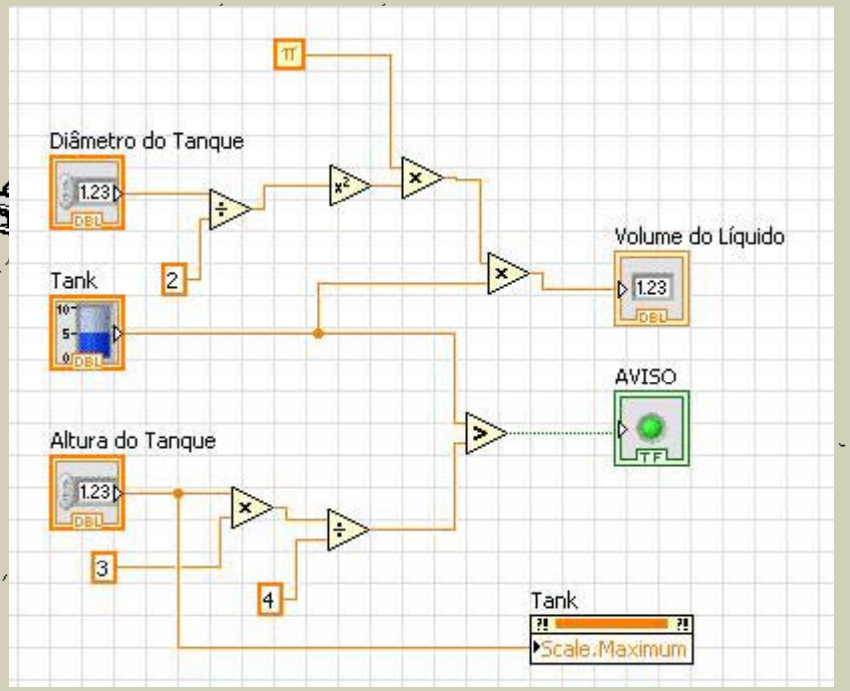
O procedimento é simples: "click" com o botão direito do rato sobre o objecto e seleccionar "All to Write"



2.4 Operações Aritméticas

Ya! Ficou bem fixe...

Associando o valor introduzido pelo utilizador em "Altura do Tanque" na propriedade *Maximum Scale* do objecto **Tank** obtém-se o seguinte diagrama de blocos!



2.4 Exercícios

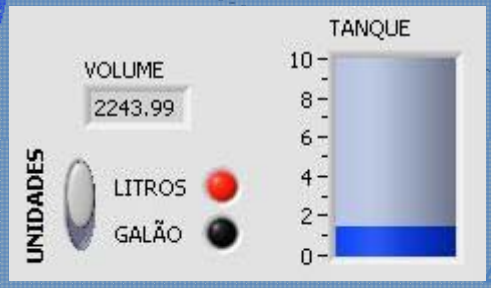
EX 8: ***Desenhe um programa para LabView que execute as seguintes funções:***

- *Soma dois números e apresenta o resultado;*
- *Multiplica os mesmos dois números e apresenta o resultado;*
- *Compara os dois resultados e acende um LED verde caso os valores sejam idênticos.*

EX 9: ***Pretende-se um programa para LabView que calcule o volume de um tanque (cilíndrico) em dois sistemas distintos de unidades: em litros ou em galão (sistema Americano)****. Para isso o programa deve possuir um botão que permita seleccionar entre essas duas unidades.*

NOTAS:

- *1 litro = 0.227 Galão e 1 Galão = 4.405 litros*
- *1 dm^3 = 1 litro*
- *Considere o diâmetro do Tanque constante e igual a 1m*
- *A altura do líquido no tanque é dada em metros.*



2.5 Sequências de caracteres

Sequências de caracteres = STRINGS

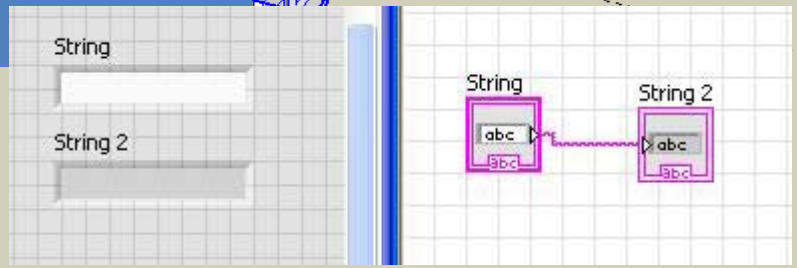
Podem ser usadas em várias situações:

- **Mensagens de texto;**
- **Passagem de dados numéricos como strings;**
- **Gravação de ficheiros de dados;**

Existem controlos e indicadores do tipo string.

Aceitam todo o tipo de caracteres ASCII (incluindo o ENTER)

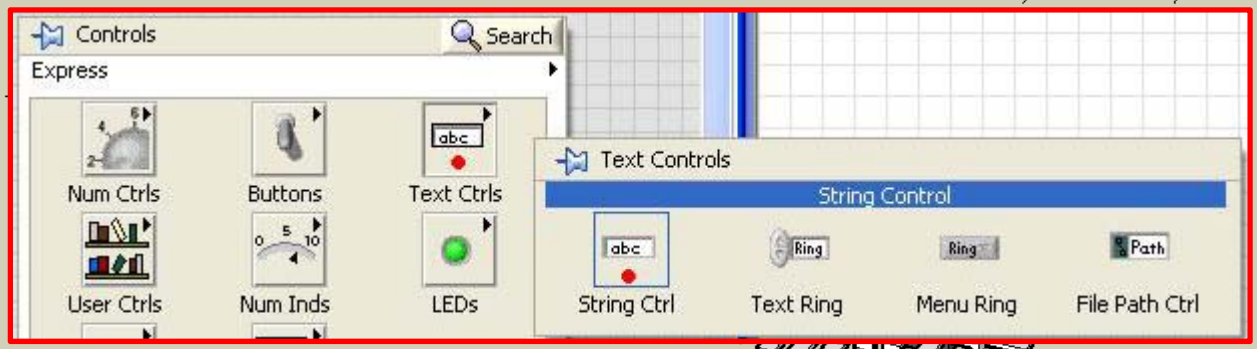
Os SINAIS e OBJECTOS no diagrama de blocos aparecem com a cor ROSA.

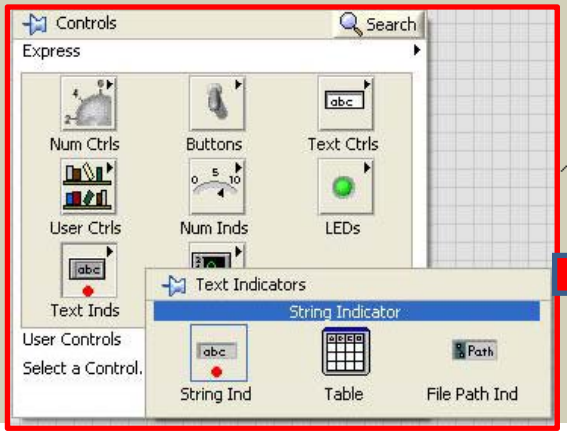


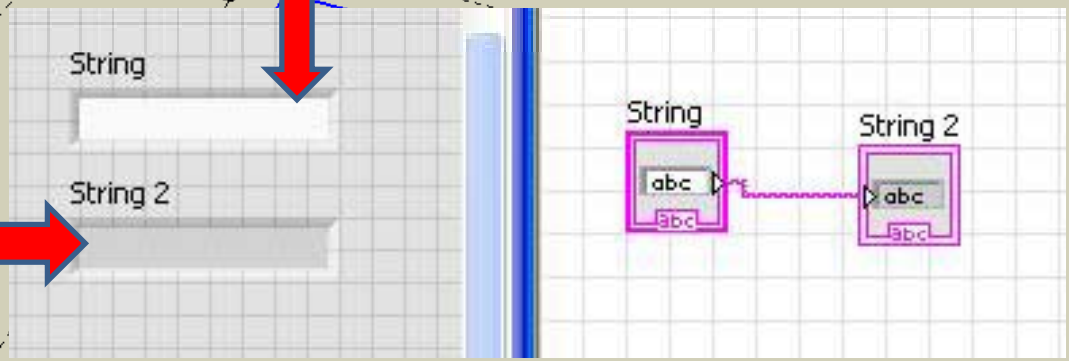
2.5 Controlos STRING

EXEMPLOS: um programa elementar e as suas variantes...

Coloque os objectos "**String Ctrl**" e "**String Ind**" no *Front Panel* e efectue a ligação apresentada em baixo...

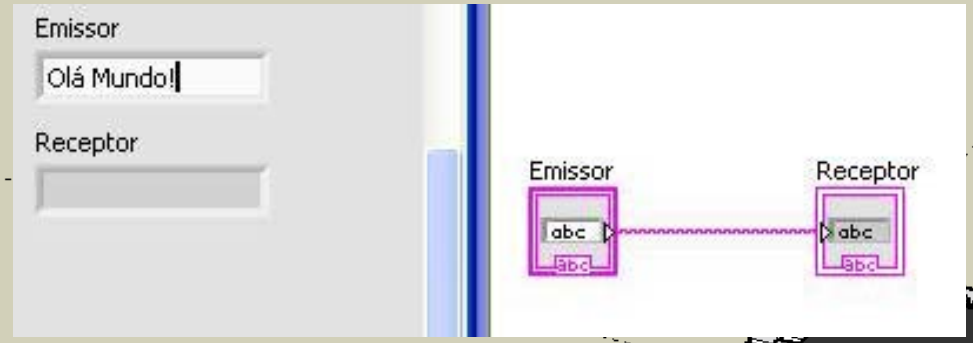






2.5 Controlos STRING

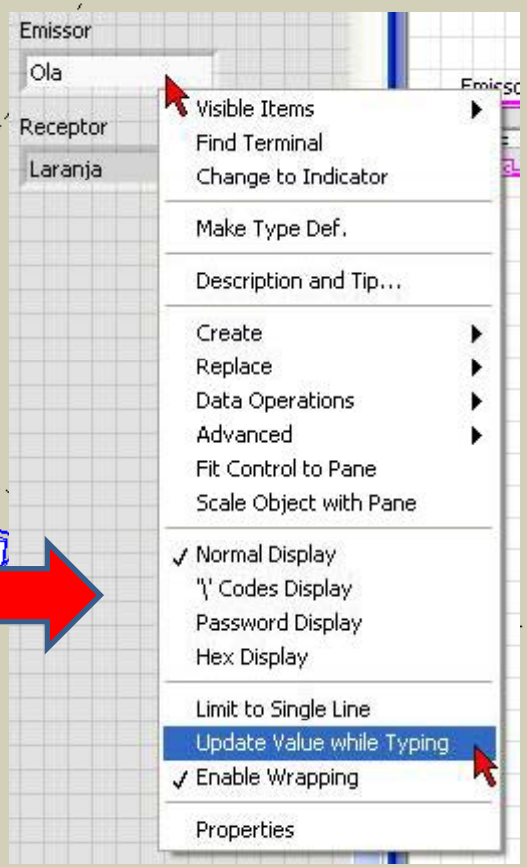
Execute o programa e escreva "Olá Mundo!" na caixa de controlo. Comente o que observa...



"clique" com o rato fora da caixa e veja o que acontece...

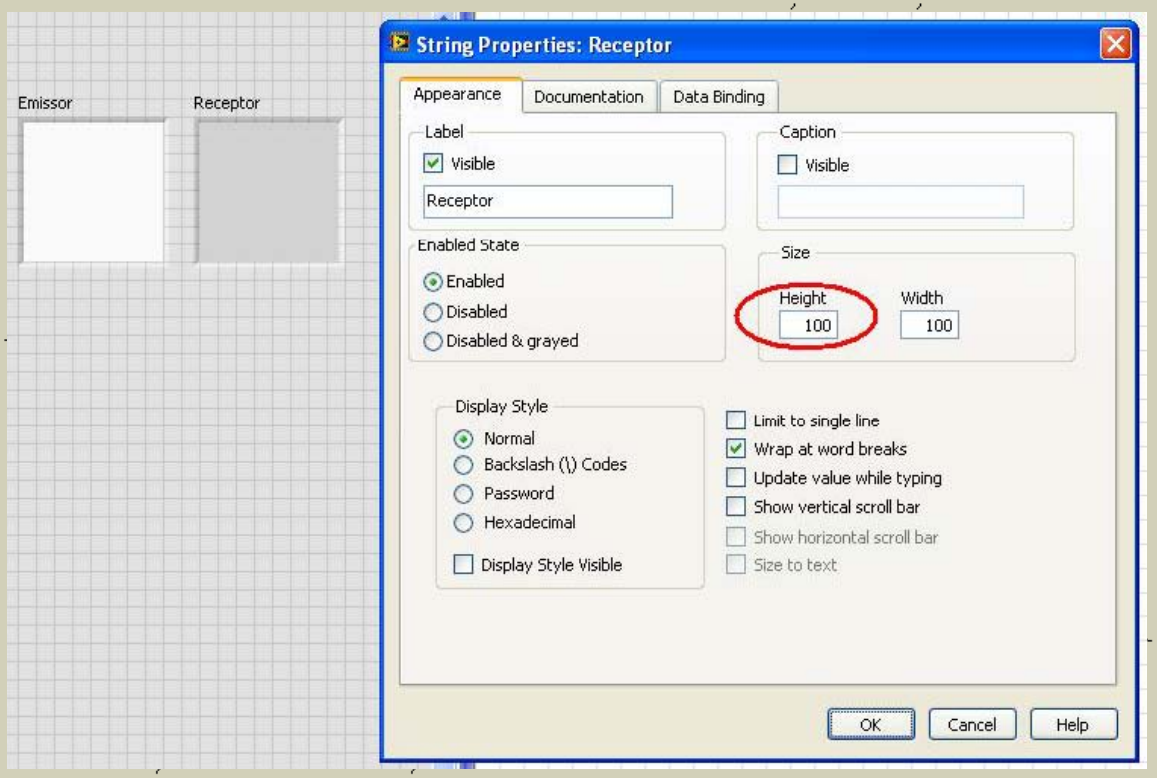
Modifique a seguinte característica da CAIXA de CONTROLO da String:

Execute o programa e volte a escrever "Olá Mundo!" . O que é que se pode concluir?



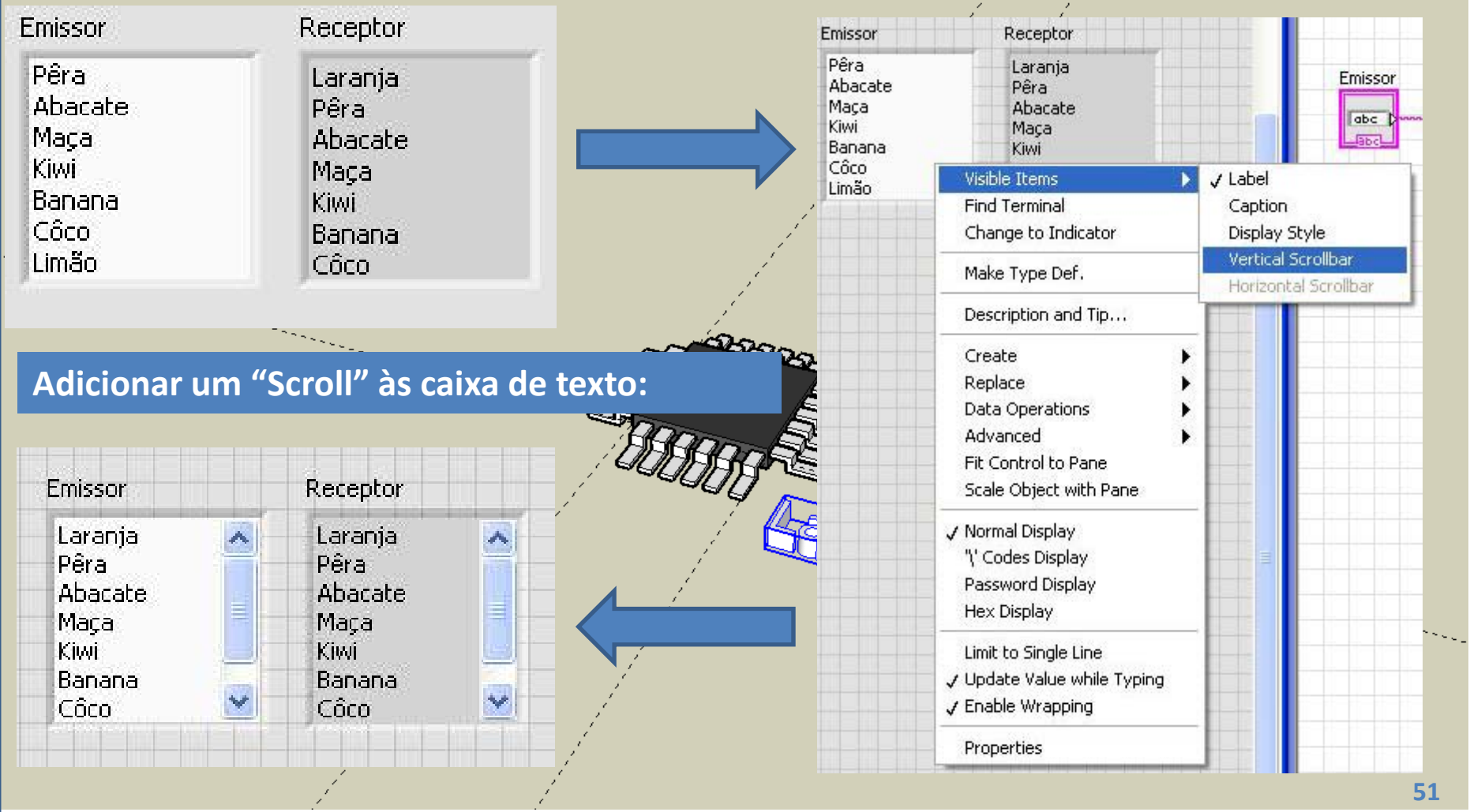
2.5 Controlos STRING

Altere a posição e dimensão dos objectos conforme se mostra na figura seguinte.



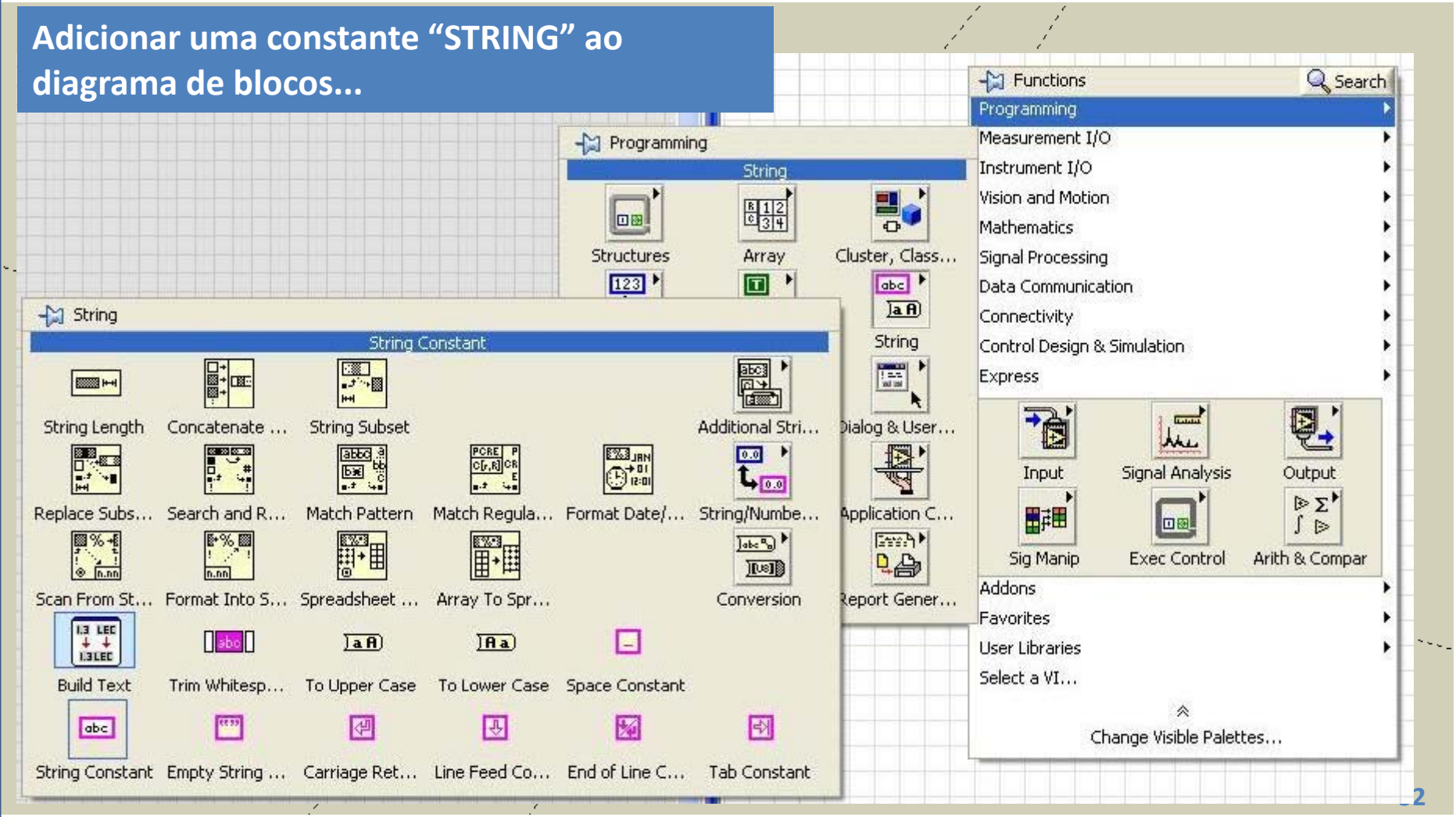
Preencha o "EMISSOR" com a seguinte lista de frutos (separados por ENTER) : Laranja, Pêra, Abacate, Maça, Kiwi, Banana, Côco e Limão.

2.5 Controlos STRING



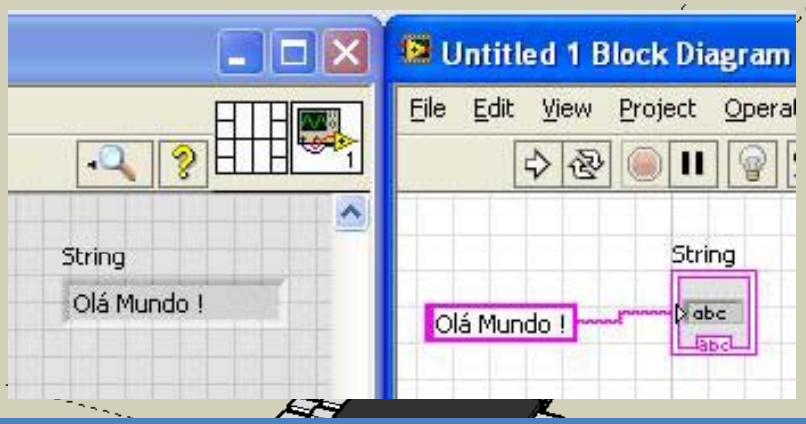
2.5 Constantes STRING

Adicionar uma constante "STRING" ao diagrama de blocos...



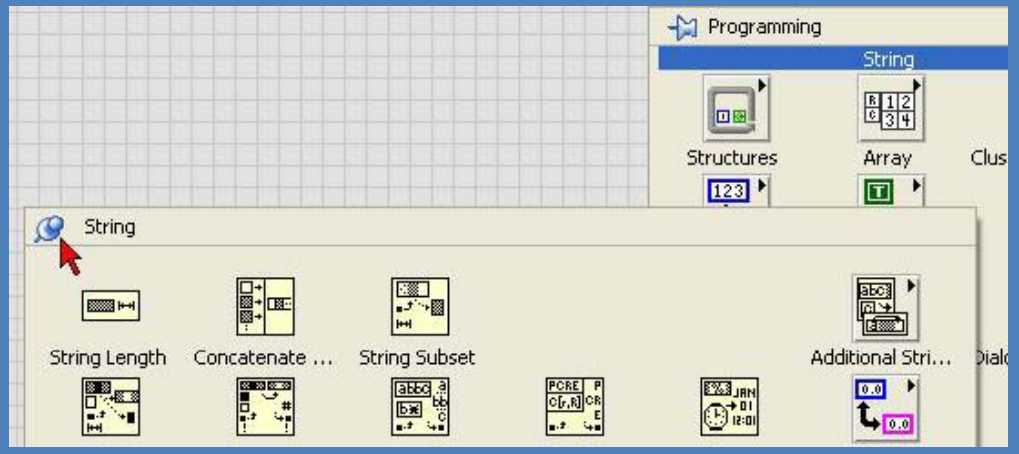
2.5 Constantes STRING

Adicionar um "String Ind" ao *Front Panel*, efectuar a ligação...Executar o programa!

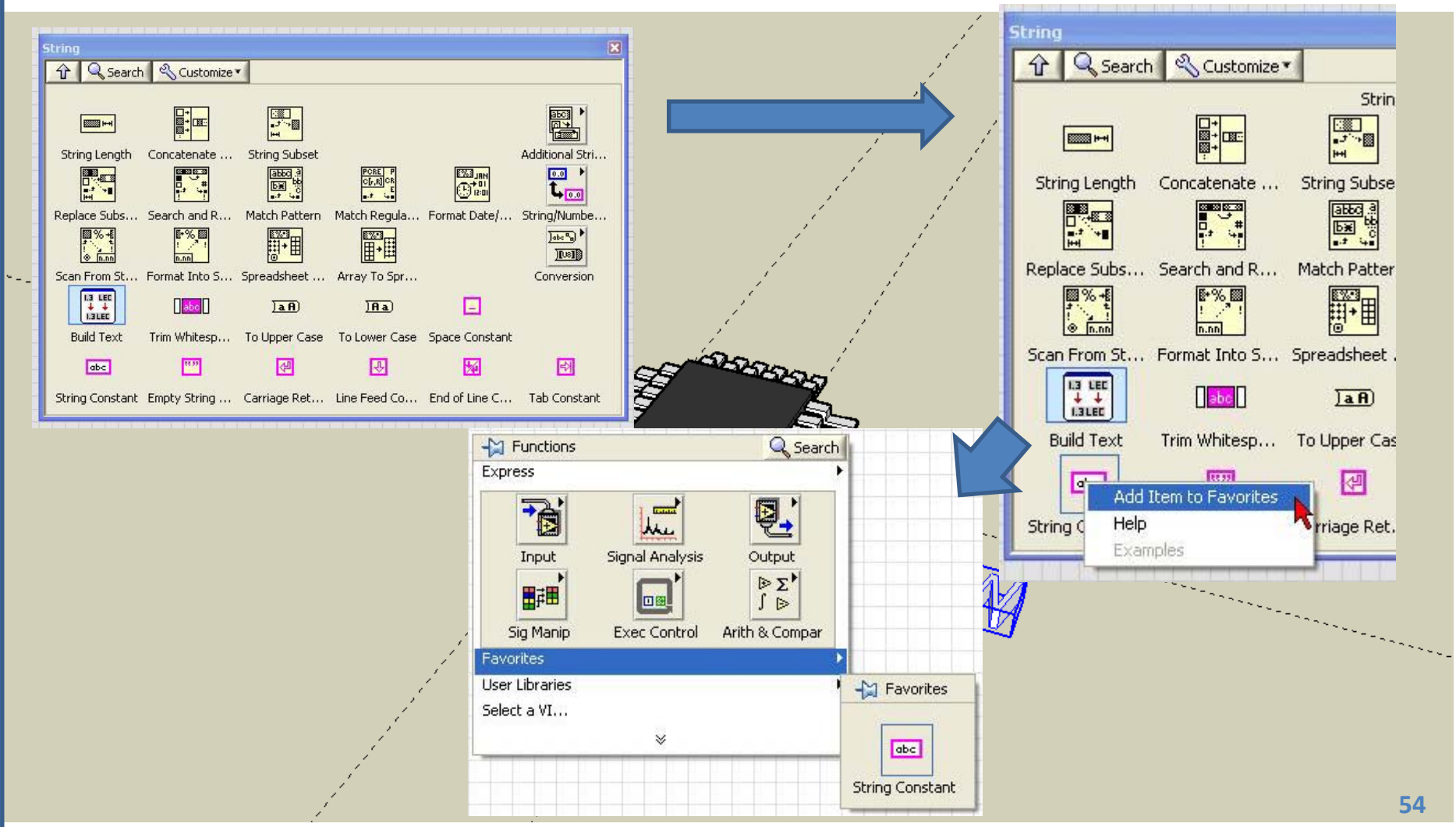


Alguns atalhos !!!!

"clique" no pin para manter a paleta activa...



2.5 Constantes STRING

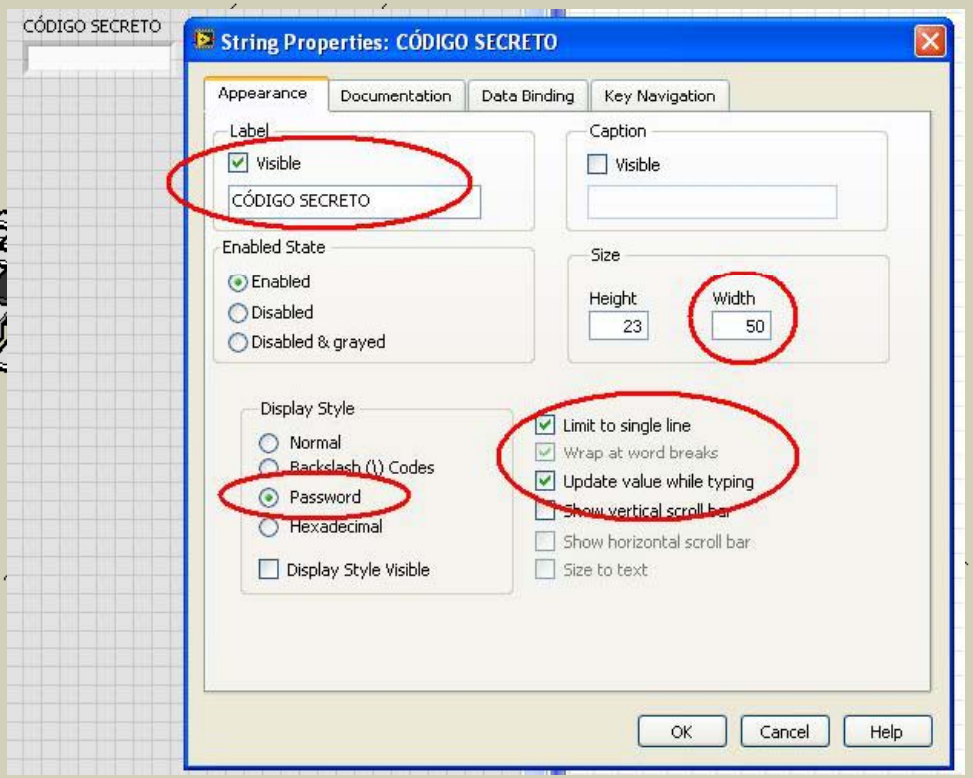


2.5 Constantes STRING

EXEMPLO ***O código secreto.***

Desenhar um programa que acende um LED se o utilizador acertar na palavra código (password) composta por 4 caracteres.

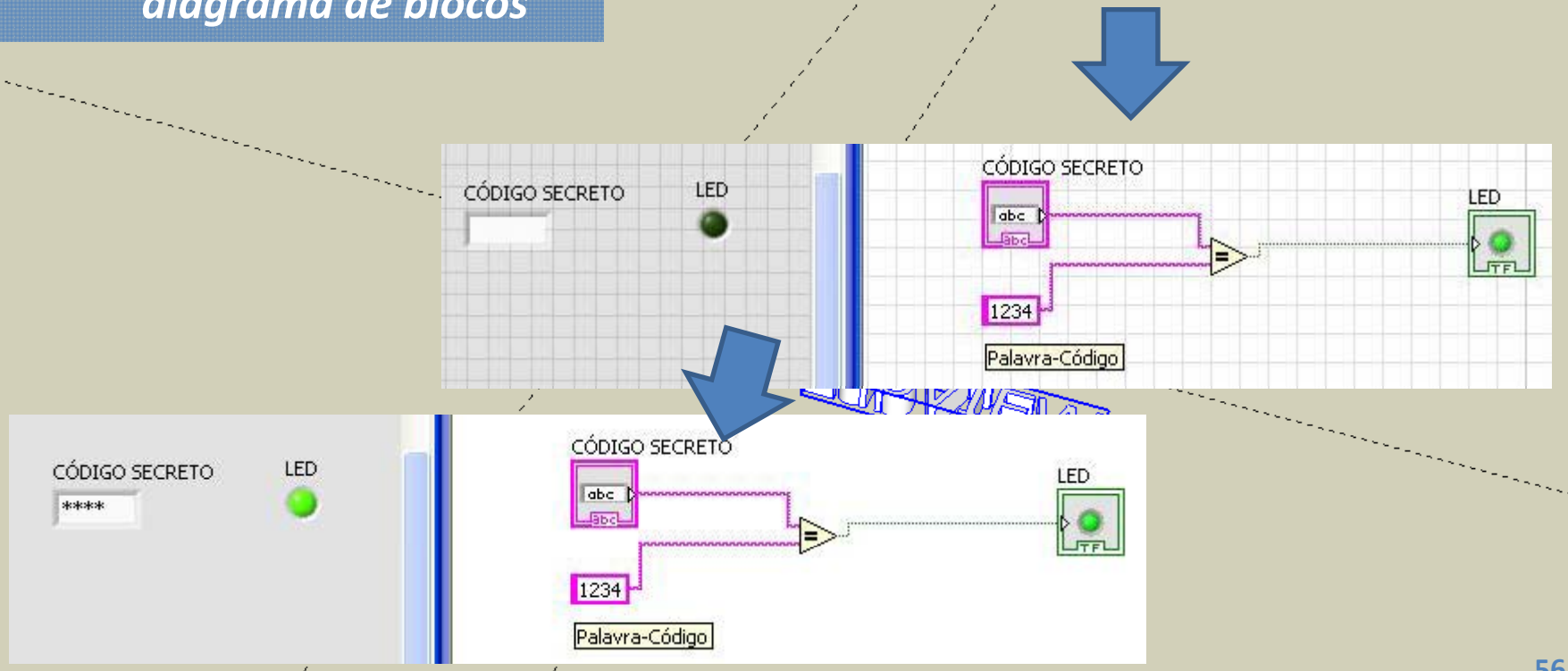
1º ***Colocar um objecto "String Ctrl" e alterar as propriedades de acordo com a figura ao lado...***



2.5 Constantes STRING

2º *Adicionar um LED ao painel frontal e um bloco comparador no diagrama de blocos*

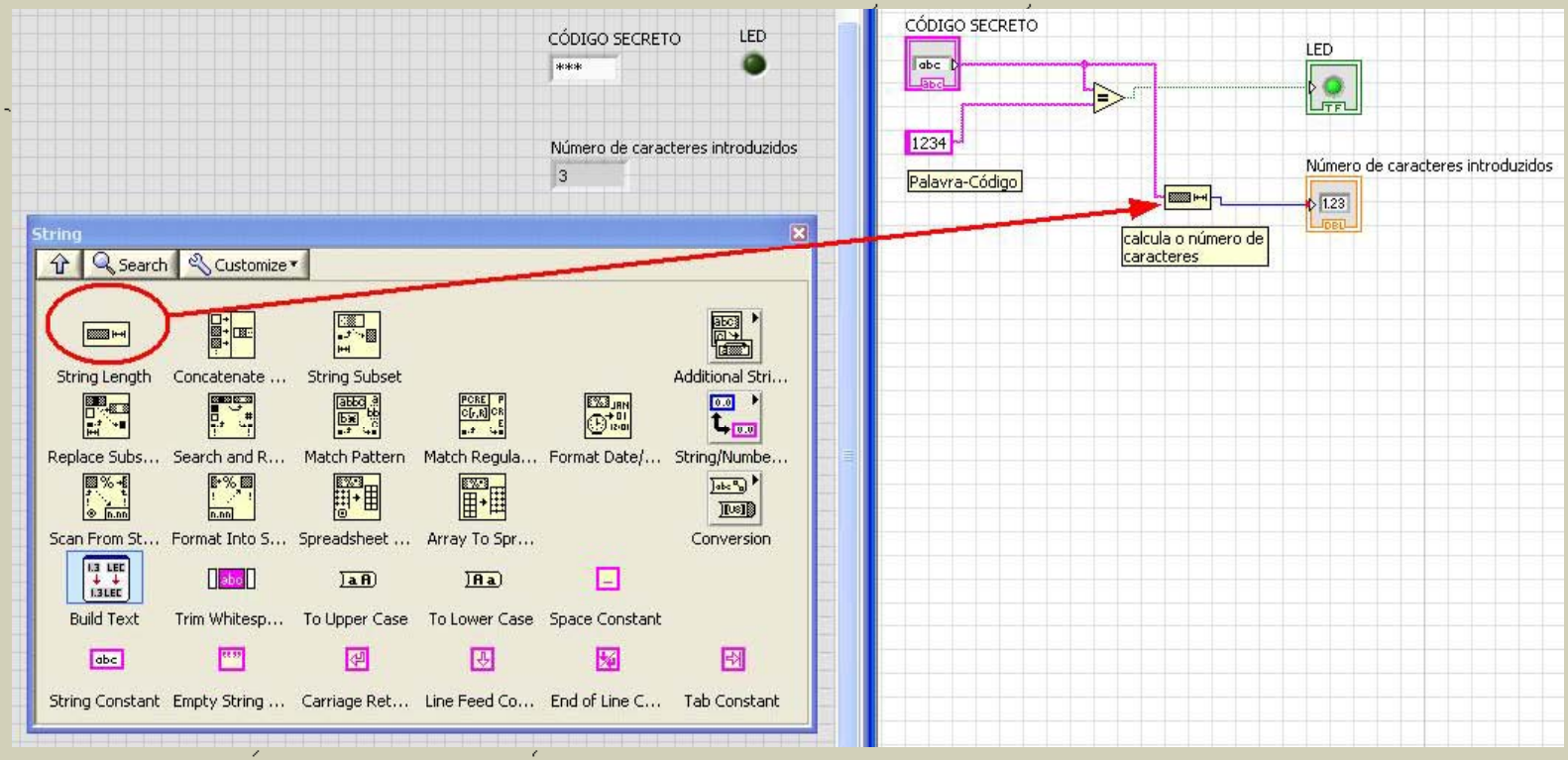
3º *Efectuar as seguintes ligações e executar o programa*



2.5 Constantes STRING

EXEMPLO ***O código secreto - Parte II.***

Adicione um indicador numérico que apresente o número de caracteres introduzidos.



2.5 Exercícios

***EX 10*: O código secreto.**

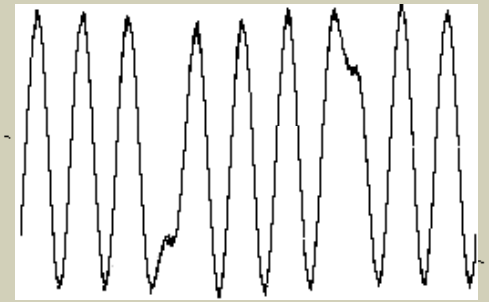
Considere a seguinte interface para uma aplicação para introdução de password:

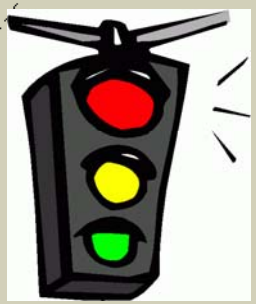
O LED "ACESSO" deve ficar verde se o código introduzido for o correcto.

O LED "ERRO" deve ficar vermelho no caso do utilizador introduzir mais do que quatro caracteres.

3.0 Sinais e Gráficos

Um SINAL é algo que transporta INFORMAÇÃO



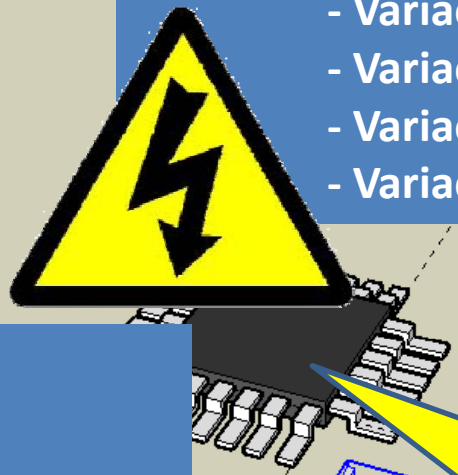


SINAIS

3.1 Sinais e Gráficos

Em instrumentação electrónica esse sinal é normalmente eléctrico:

- Variação de tensão;
- Variação de corrente;
- Variação de resistência;
- Variação de frequência, fase, etc.



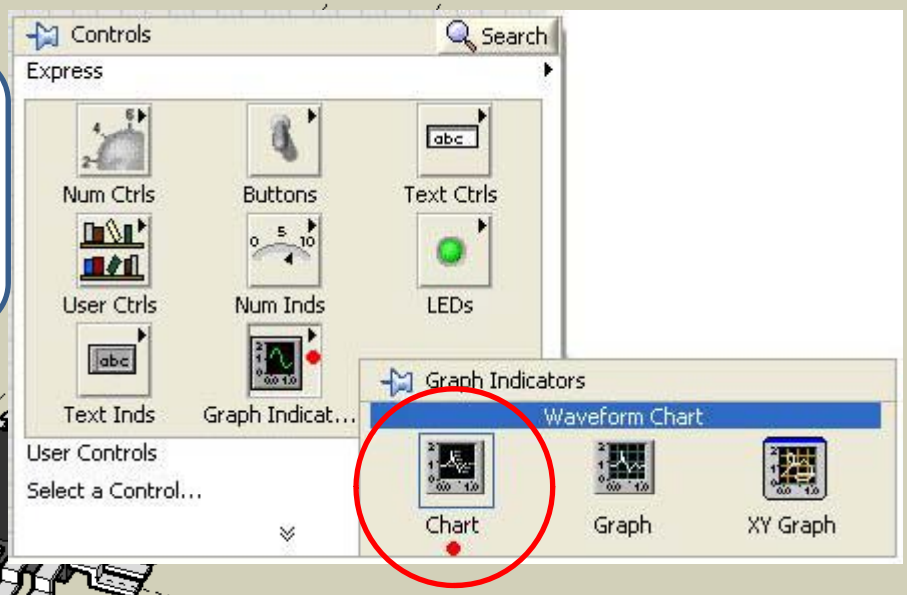
Tipos elementares de sinais:

- Ruído;
- Sinais periódicos;
  - . Sinusoidal;
  - . Triangular;
  - . Rectangular

DEMO com GERADOR DE SINAIS

3.2 **Gráficos:** *charts*

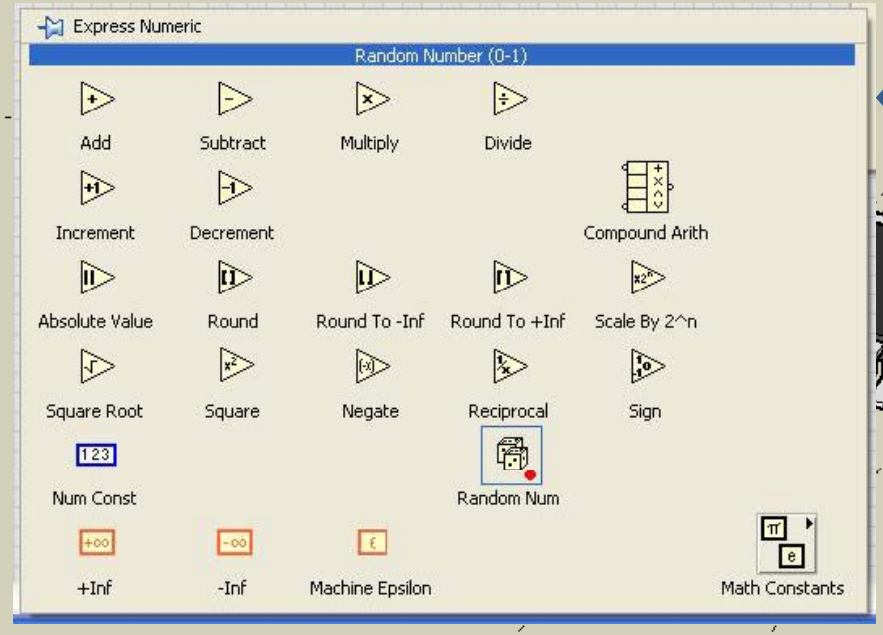
Chart?... Então não é *charter*



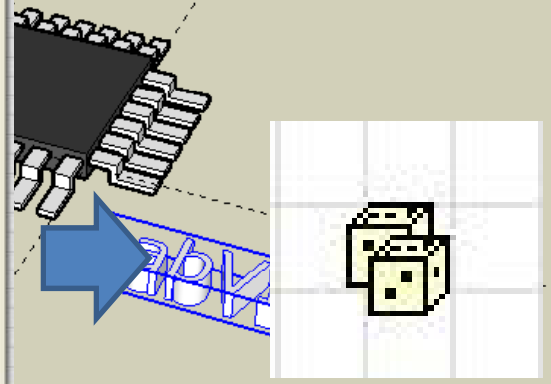
- **Um gráfico do tipo chart armazena e mostra um conjunto de valores.**
- **Esses pontos são guardados internamente num buffer (memória).**

3.2 Gráficos: *charts*

EXEMPLO: Gerador de ruído branco...



1º ***coloque o objecto "Random Num" no diagrama de blocos.***

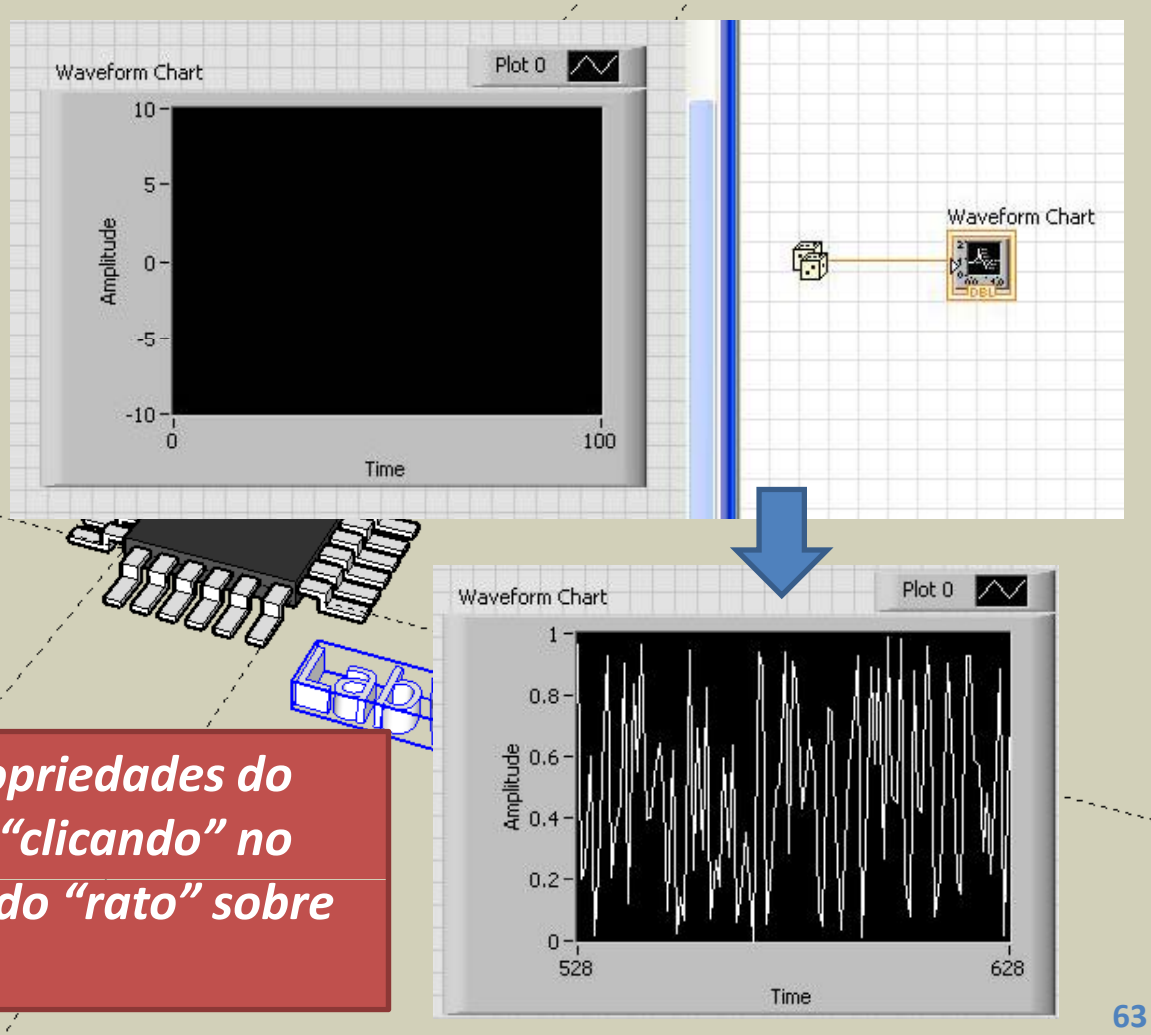


Este objecto gera um valor aleatório entre 0 e 1

3.3 Gráficos: *charts*

2º *Adicione um objecto "chart" à interface gráfica.*

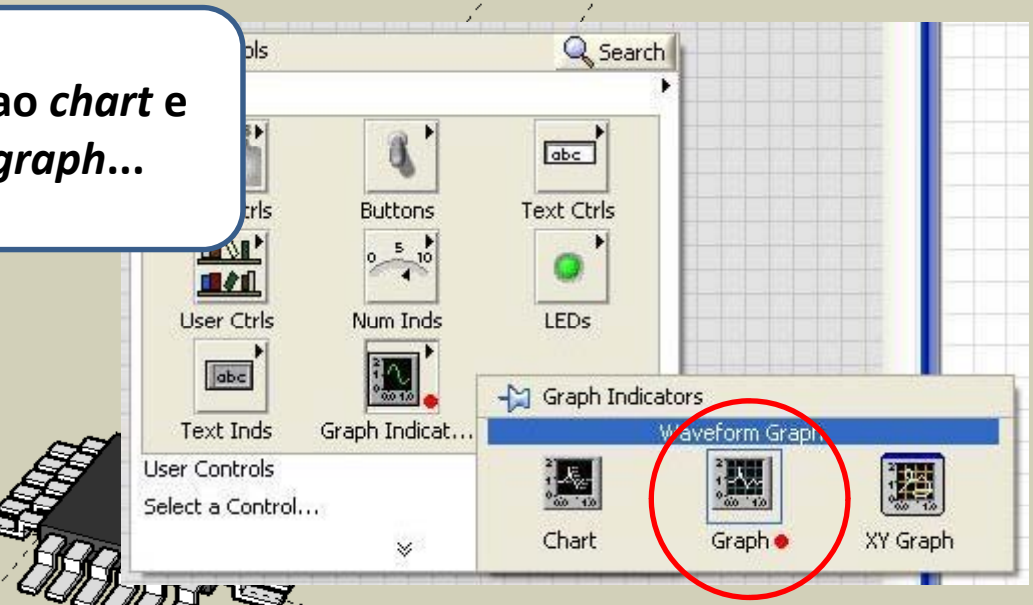
3º *Efectue a ligação entre os dois componentes e execute o programa…*



SUGESTÃO: *Explore as propriedades do objecto chart "clicando" no botão direito do "rato" sobre o objecto.*

3.3 **Gráficos:** graphs

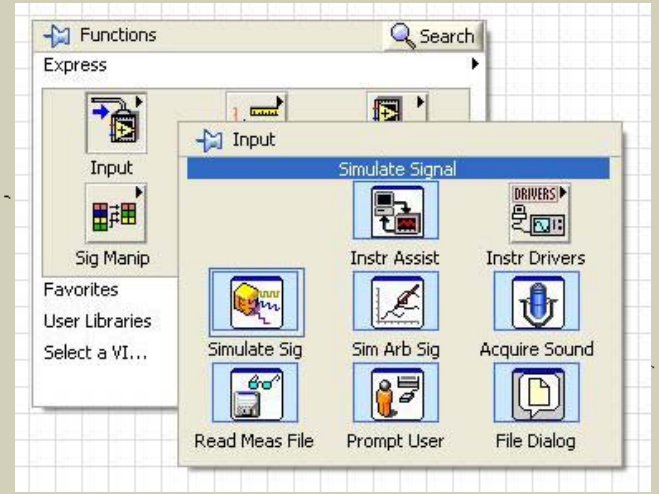
Dou nota 20 ao *chart* e nota 20 ao *graph*...

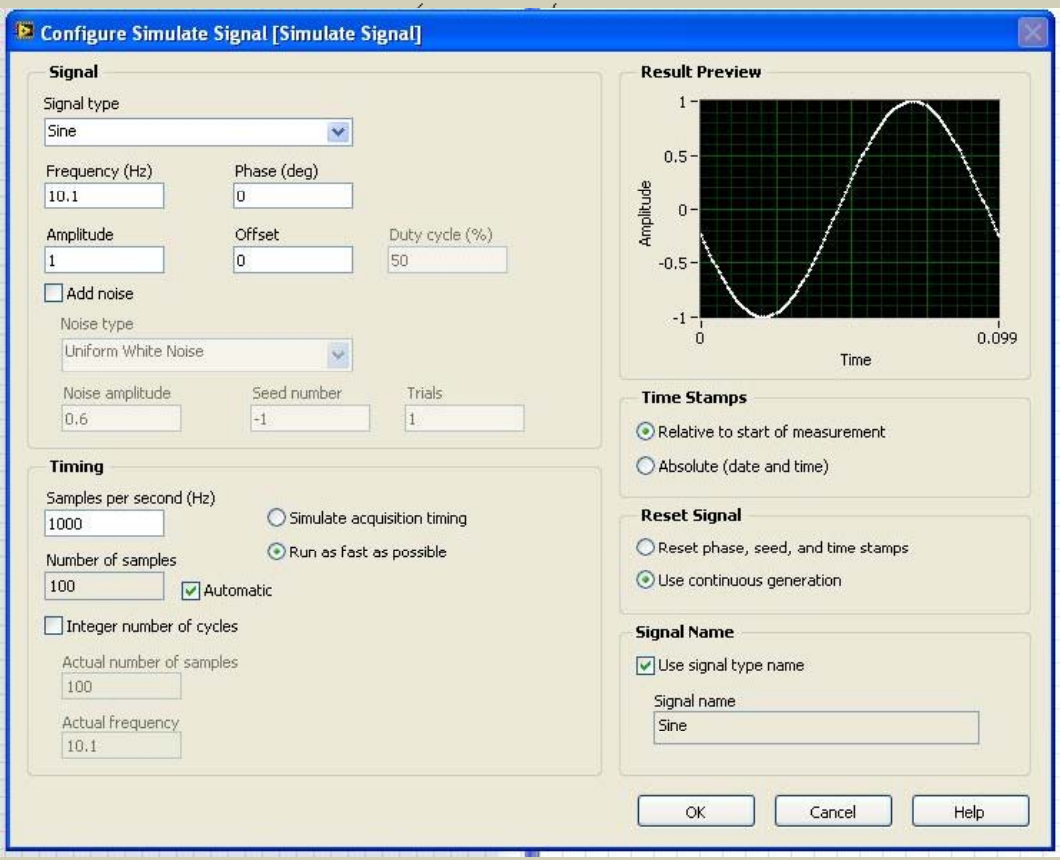


- Um gráfico do tipo ***graph*** requer a informação seja introduzida como **vector**.
- O **vector** funciona como buffer (memória).

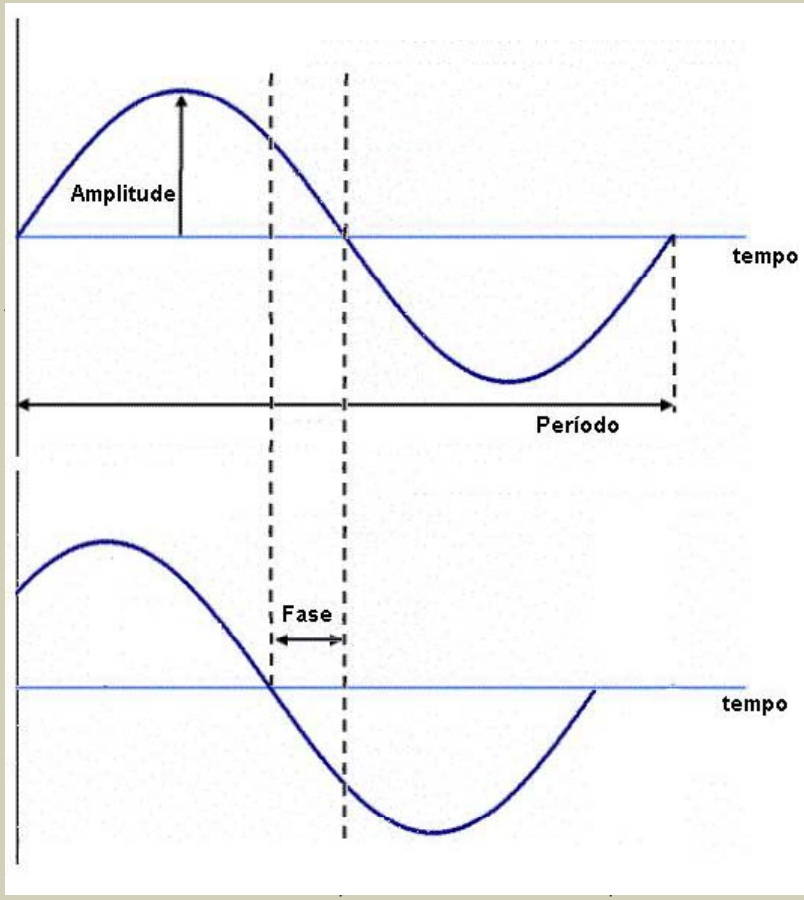
3.4 Gerador de Sinais

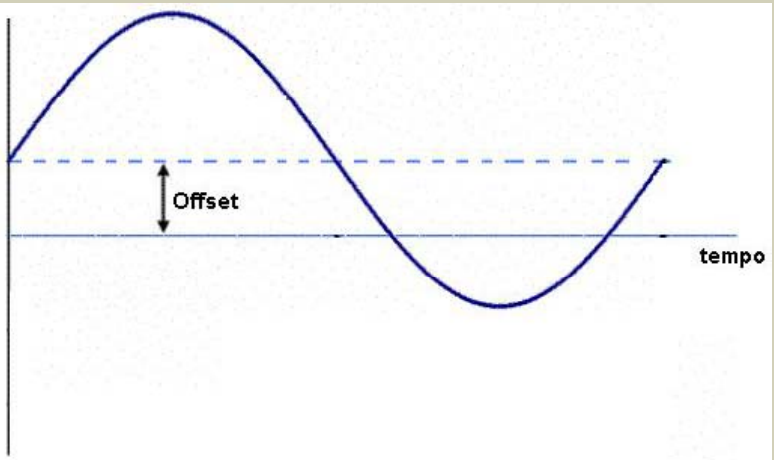
Sinais periódicos podem ser gerados facilmente com o bloco "Simulate Sig"





3.4 Gerador de Sinais



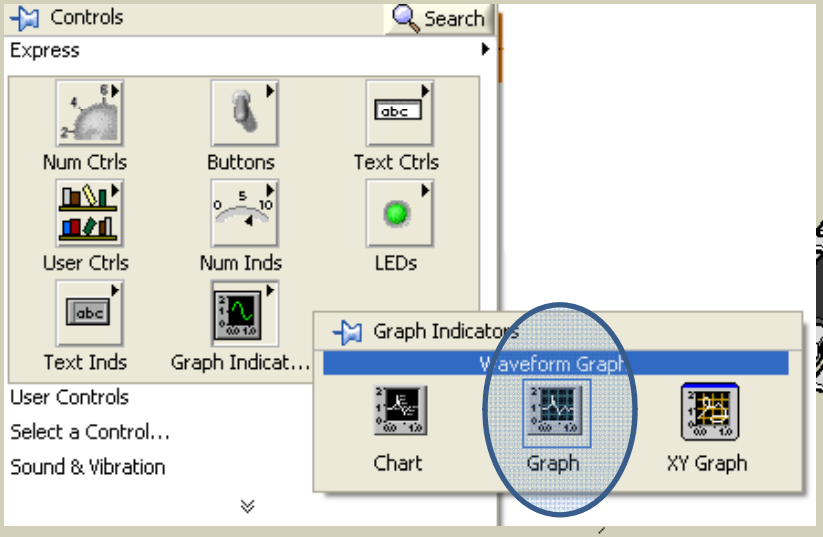


A frequência (f) não é mais do que o inverso do período (T):

$$f = \frac{1}{T}$$

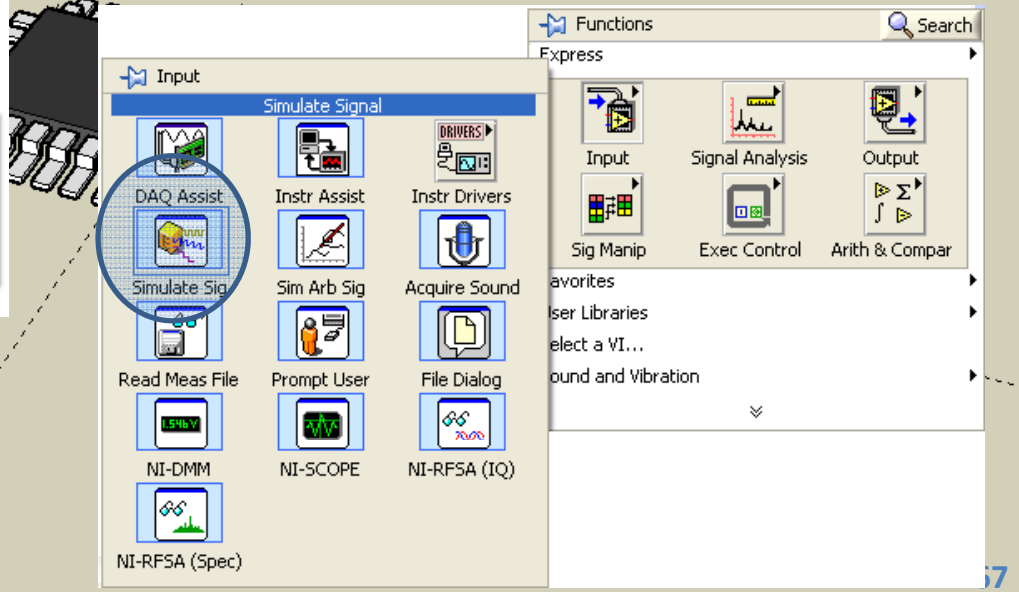
3.4 Gerador de Sinais

Realizar um programa que apresente, num gráfico, um sinal sinusoidal contaminado com ruído branco. A relação sinal/ruído deve poder ser alterada através de um "knob"



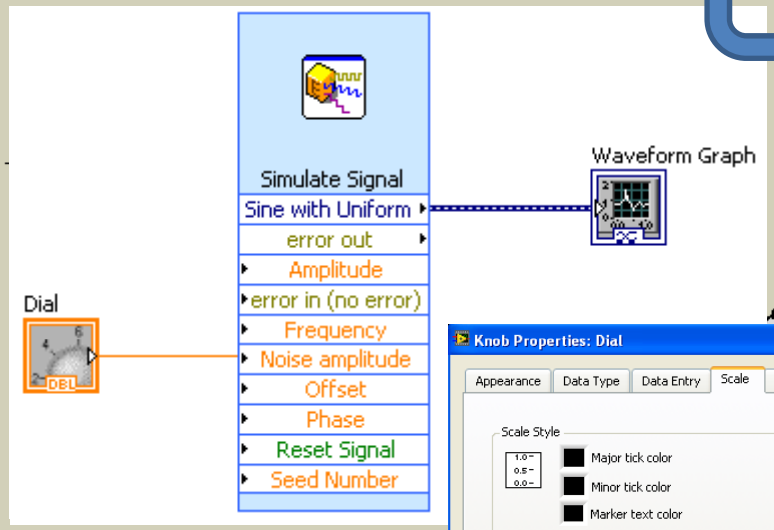
1 – Inserir Gráfico
2 – Inserir "KNOB"

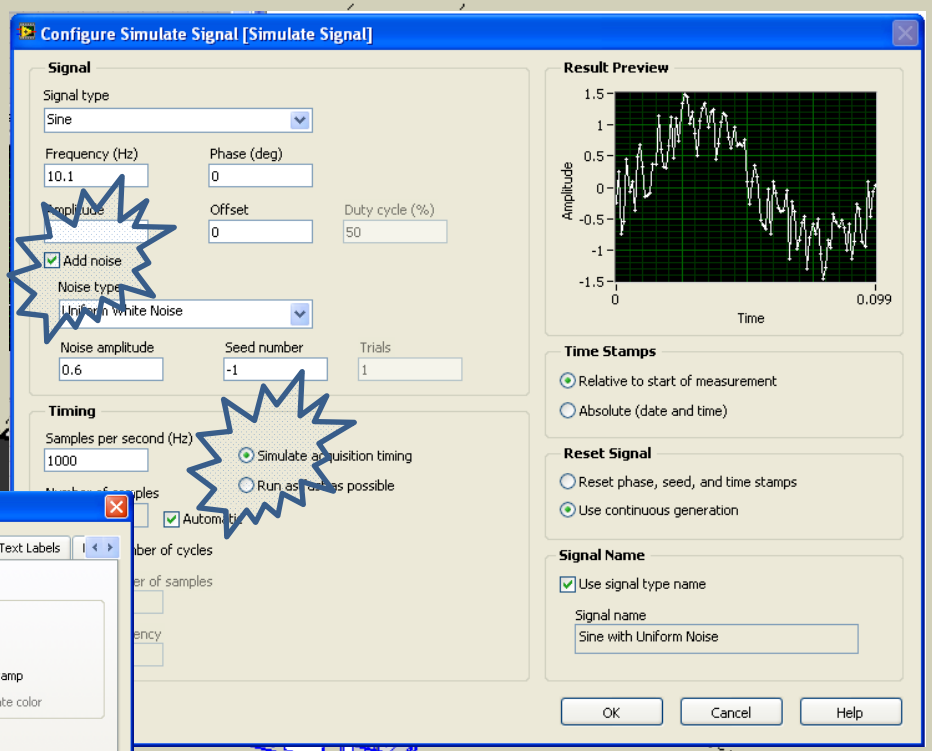
3 – Inserir SINAL

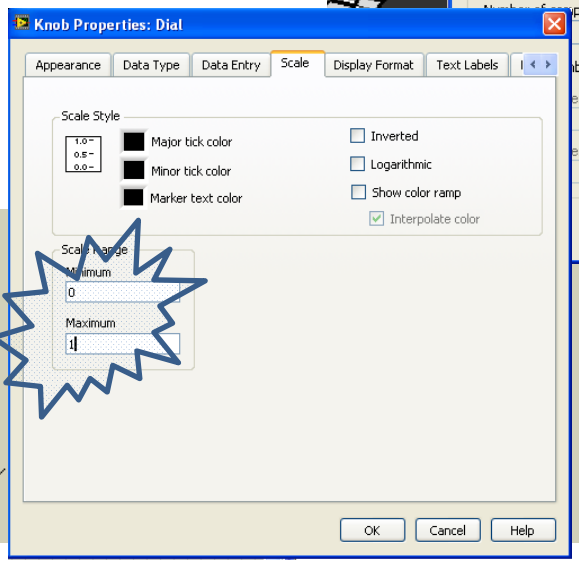


3.4 Gerador de Sinais

4 – Alterar propriedades do SIGNAL





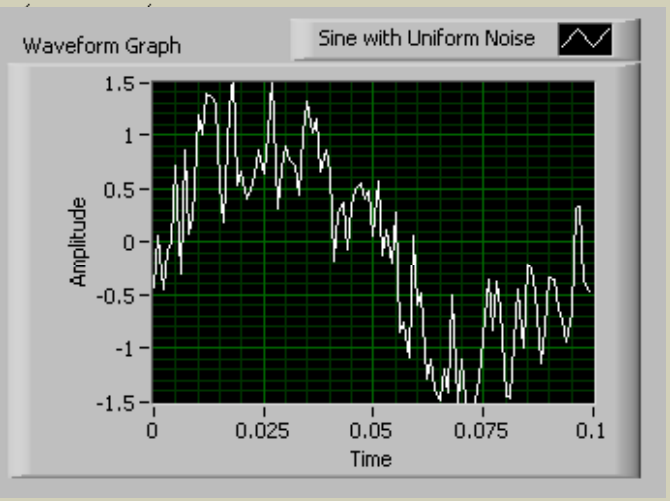


5– Alterar propriedade “Maximum” e “Minimum” do ‘KNOB’

6 – Efectuar as ligações…

3.4 Gerador de Sinais

7 – Simular: Girar o "KNOB" e verificar a evolução da forma de onda.



Exercício:

Repita o exercício anterior adicionando botões para controlar a amplitude e frequência do sinal.

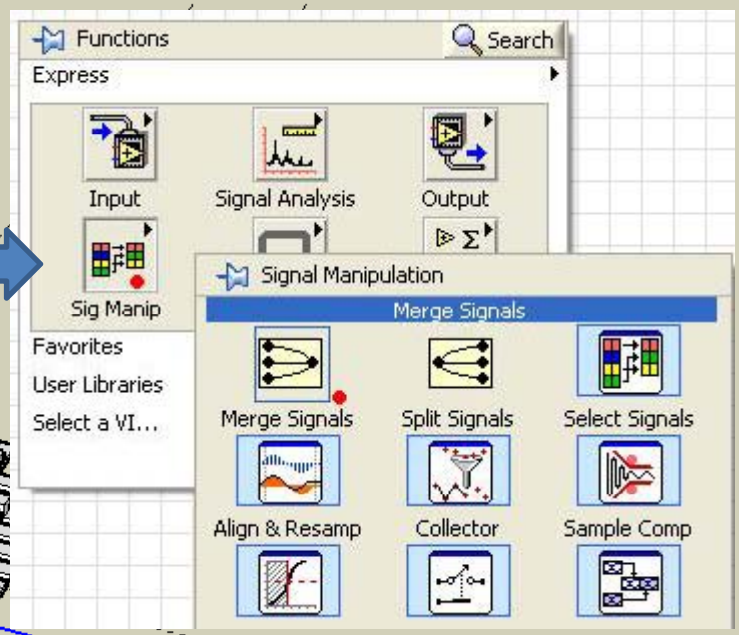
Na mesma janela de visualização é possível observar mais do que um traço *simultâneamente*.

COMO?

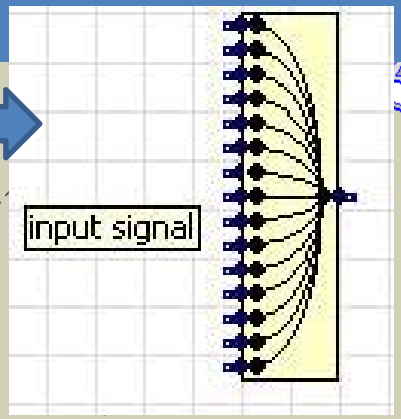
Através de um "multiplexador"

3.5 *Mux.* de Sinais

Para "*juntar*" sinais num mesmo gráfico deve aceder à paleta "Sig Manip" e seleccionar o objecto "Merge Signals"

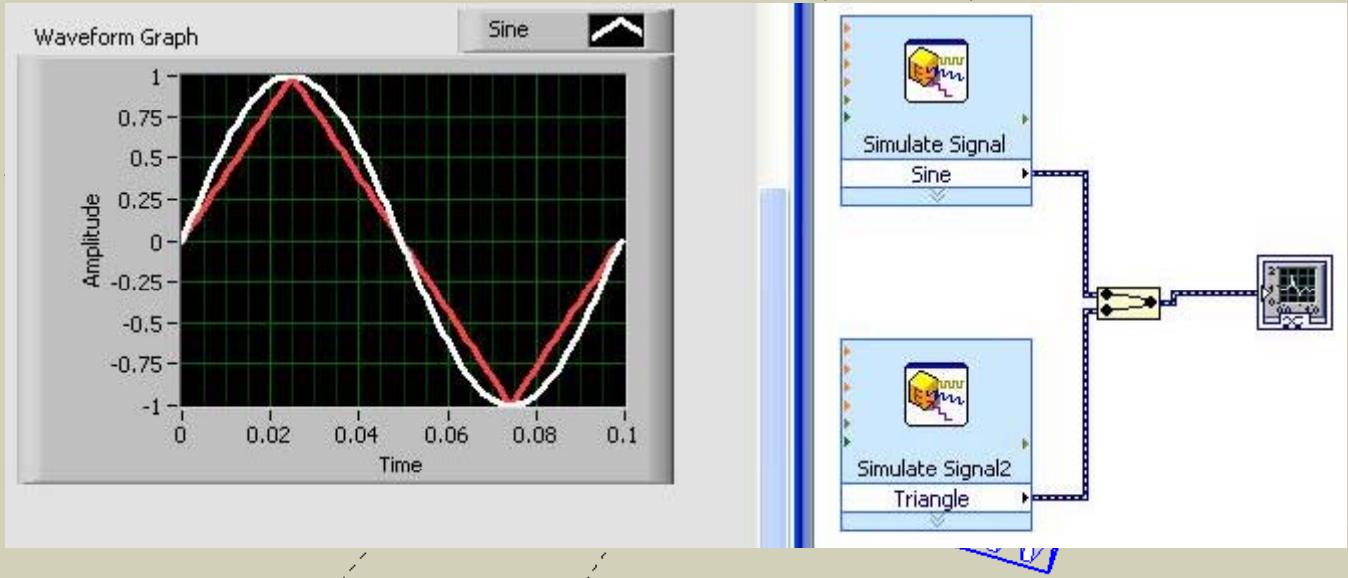


Utilizando o "rato", e redimensionando o objecto "Merge Signals" no diagrama de blocos, extende-se o número de sinais que podem ser agregados...



3.5 *Mux.* de Sinais

EXEMPLO: Apresentar na mesma janela uma onda sinusoidal e uma onda triangular com igual amplitude e frequência.



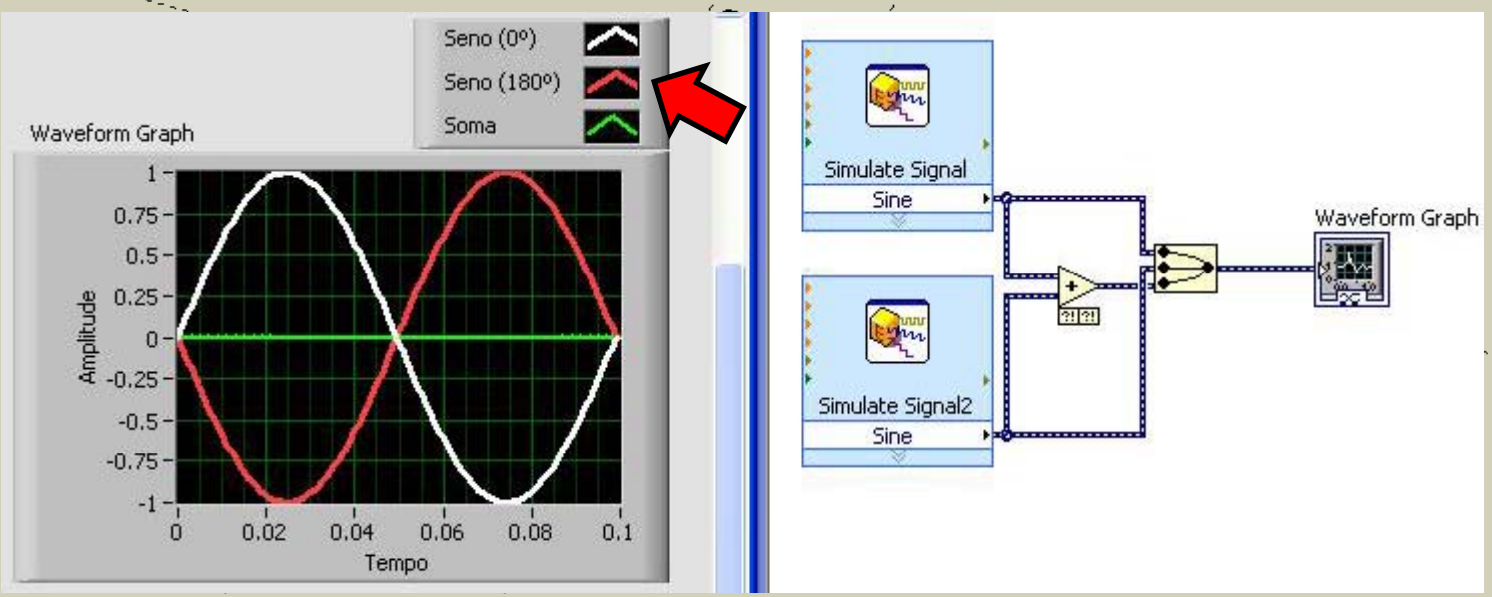
EXERCÍCIO: Adicione ao gráfico do exemplo anterior uma onda quadrada cujo ciclo de trabalho (dutty-cycle) possa ser alterado por um "knob".

3.6 *Aritmética com* Sinais

É possível efectuar operações algébricas sobre sinais:

- **Soma, subtracção, multiplicação e divisão;**
- **Funções transcendentes (logaritmos, exponenciais, etc.)**

EXEMPLO: Apresentar o gráfico da soma de duas sinusóides com diferença de fase igual a 180º. Devem ser apresentados os traçados das funções individuais.

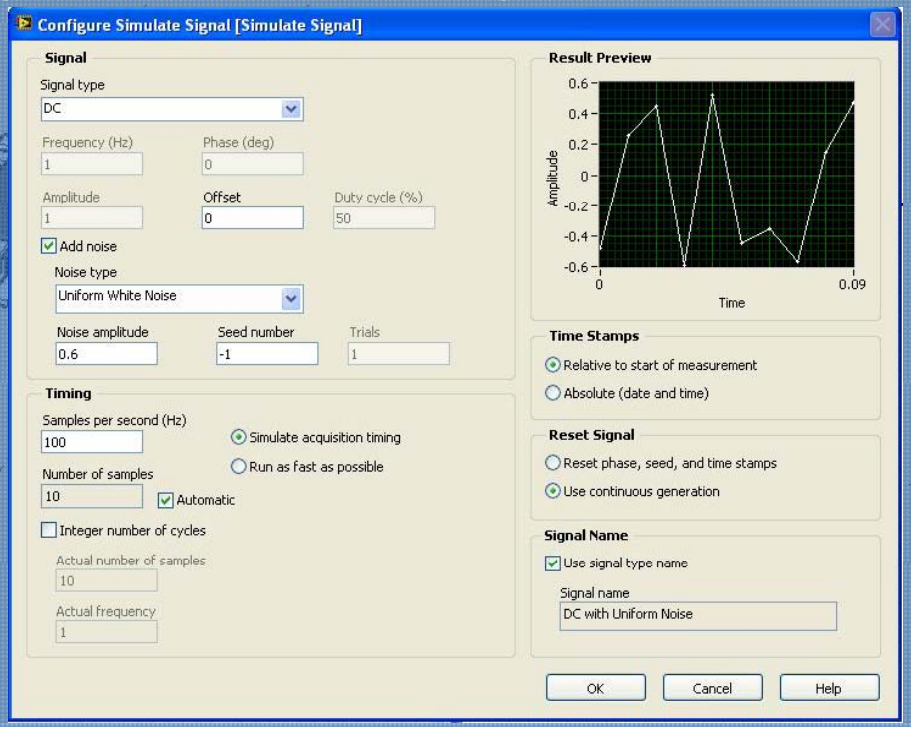


3.7 Exercícios

***EX 11**: Faça um programa para LabView que seja capaz de apresentar a soma de uma sinusóide com o seu harmónico de segunda ordem (sinal igual mas com o dobro da frequência).*

***EX 12**: Apresente num gráfico o aspecto de um sinal do tipo ruído branco com valor médio nulo.*

*A configuração desse sinal é apresentado na figura ao lado. Deve ser adicionado um **LED** que deve acender sempre que a <u>amplitude</u> do sinal for maior que 0.5 ou menor que -0.5.*



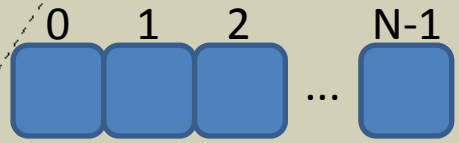
4.0 Arranjos (arrays)

Até ao momento apenas se trabalhou com números escalares.

Em LabView um arranjo (array) consiste numa colecção de elementos todos do **mesmo** tipo.

Um array pode ter uma ou mais **dimensões**.

Os **índices** são numerados de **0** a **N-1** onde **N** se refere ao número de elementos do array.



Num **arranjo** os elementos são acedidos pelos seus **índices**.

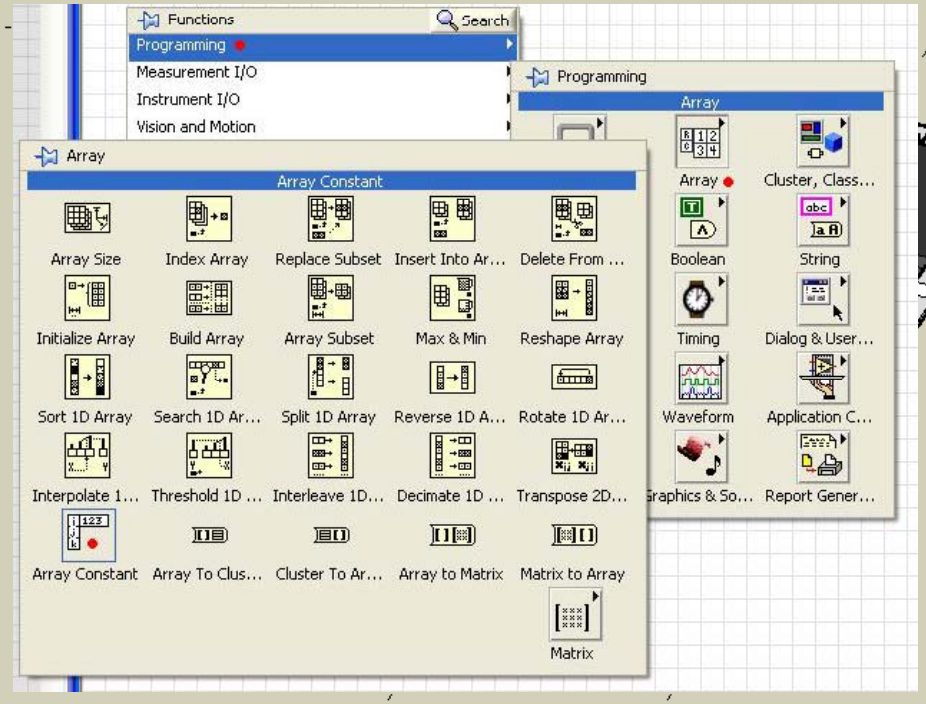
Que complicado... Valha-me D. Afonso Henriques!

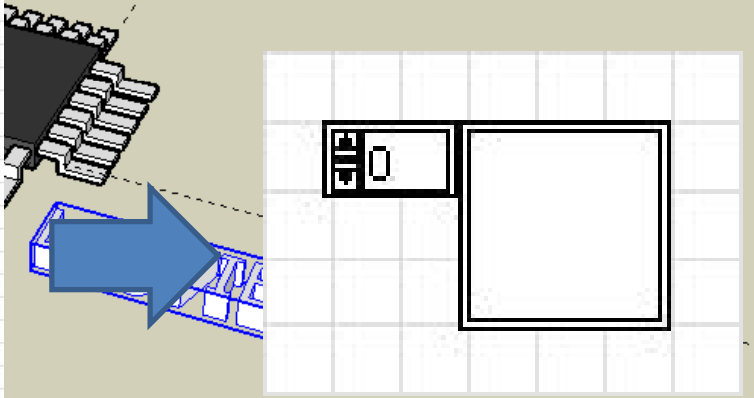
Formas de onda são normalmente armazenadas em arrays.

4.0 Arranjos (arrays)

Quando se cria um arranjo é necessário atribuir-lhe um tipo de dados. Assim devem efectuar-se dois passos:

1º Colocar o objecto ARRAY no diagrama de blocos



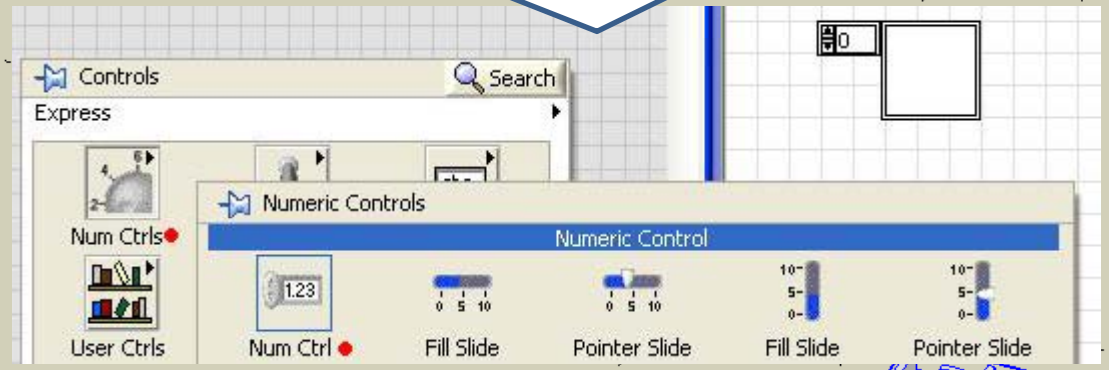


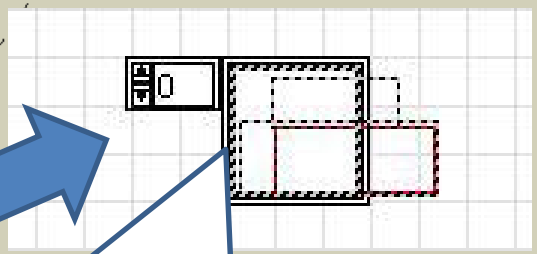
4.0 Arranjos (arrays)

2º Atribuir-lhe um tipo de DADOS:

Numérico:

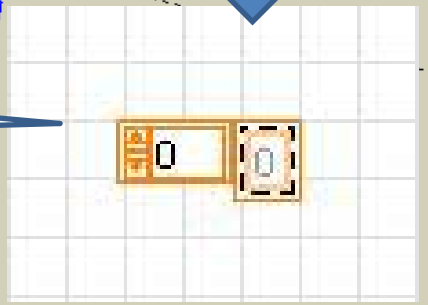
(a) Seleccionar controlo <u>numérico</u>



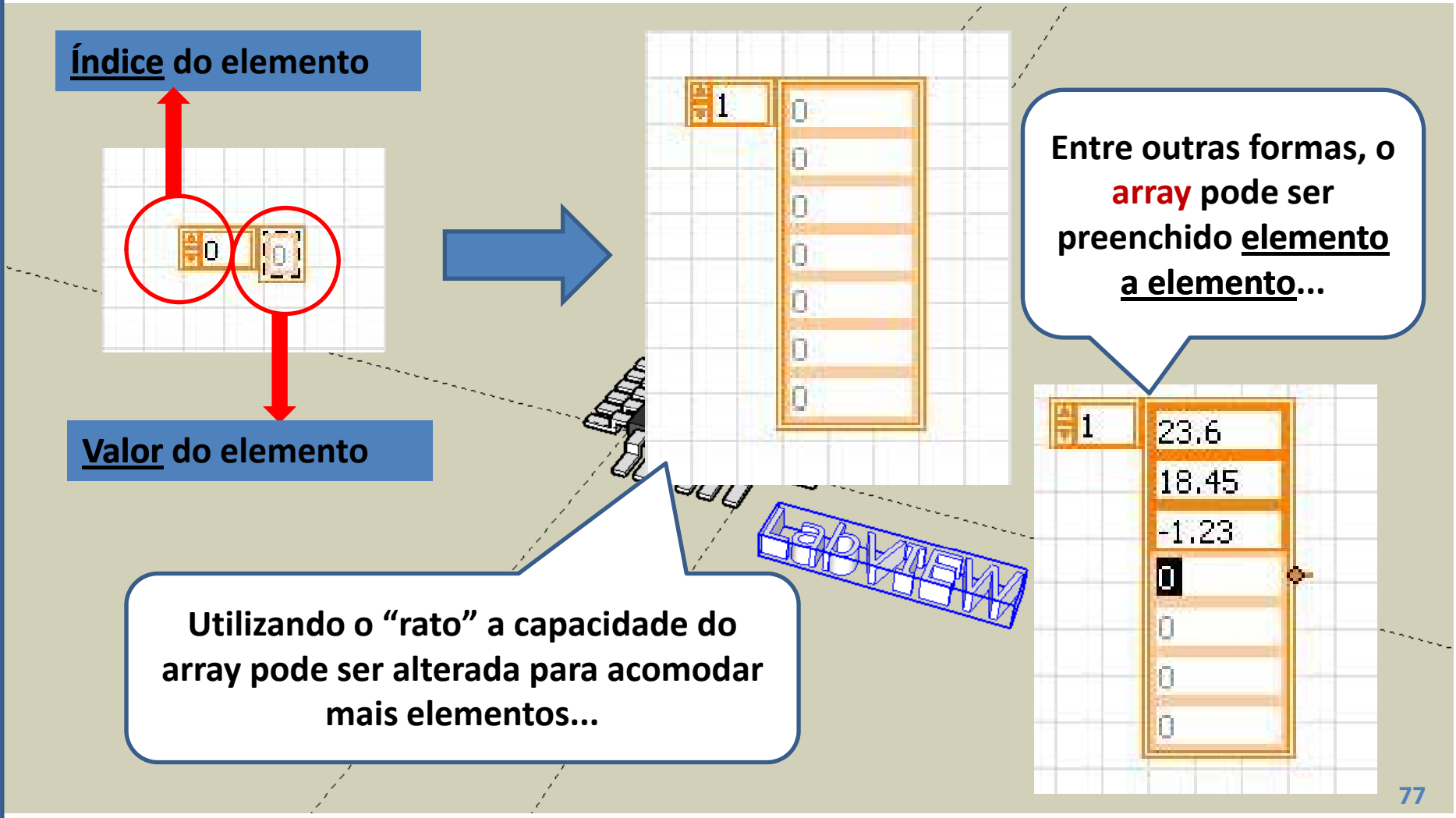


(b) "arrastá-lo" para <u>dentro</u> do objecto ARRAY

(c) Libertar o botão do "rato" e ... *volilá*



4.0 Arranjos (arrays)



4.0 Arranjos (arrays)

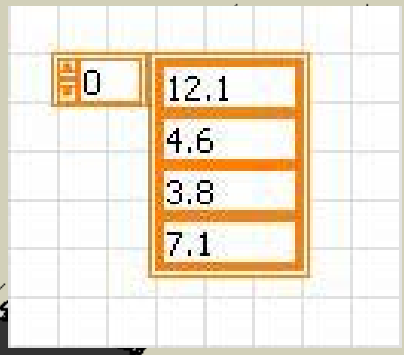
EXEMPLO: Apresentação do conteúdo de um vector num gráfico.

Afinal isto não tem nada que saber...

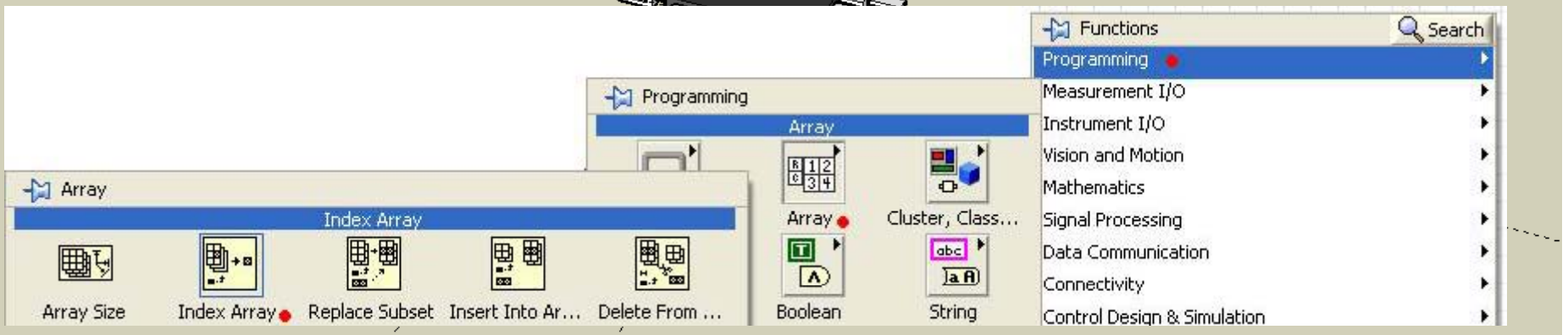
4.0 Arranjos (arrays)

EXEMPLO: Aceder a elementos num vector.

Introduza o seguinte ARRAY no diagrama de blocos...

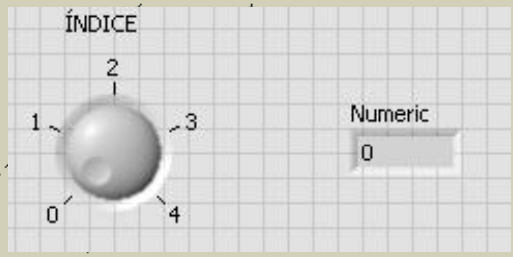


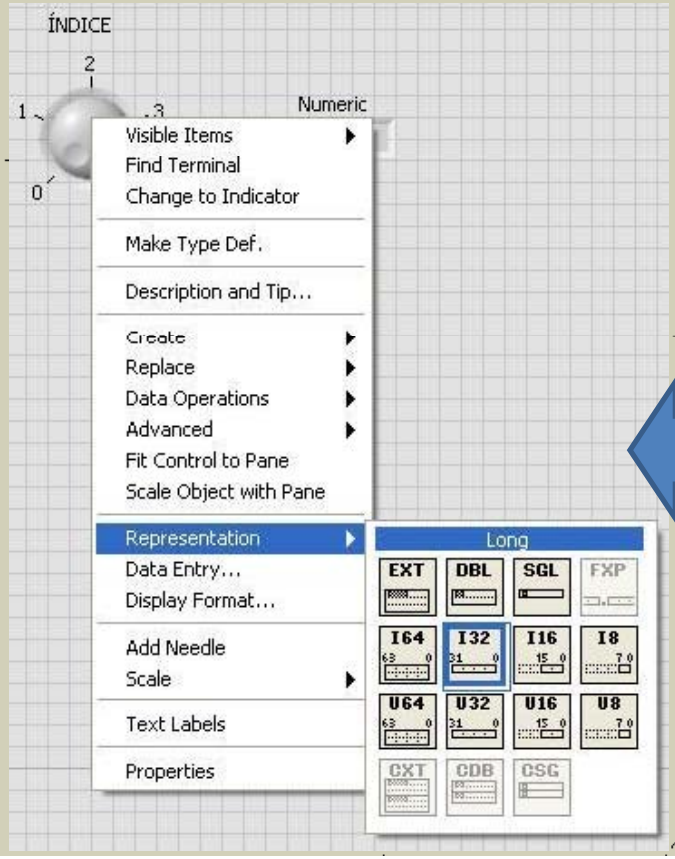
Acrescente o objecto "**Index Array**"



4.0 Arranjos (arrays)

Ao **Front Panel** acrescente um "**knob**" e um "**Numeric Ind.**"





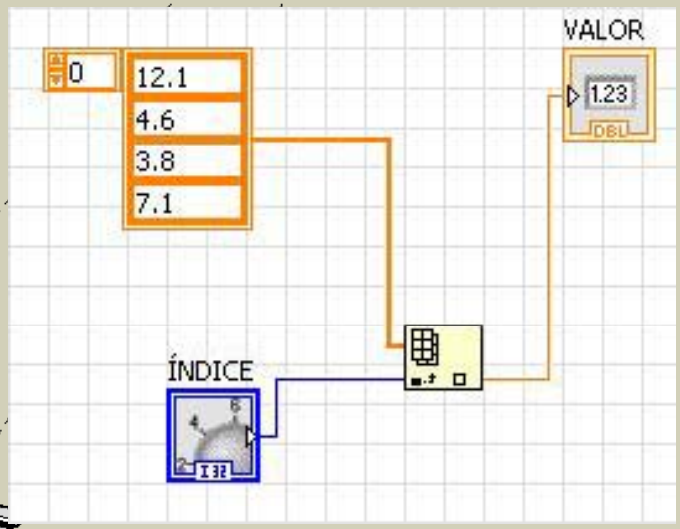
O "knob" será responsável por definir o índice do elemento a aceder e o valor do elemento apontado por esse índice irá ser apresentado no "Numeric Indicator".

Como o índice deve ser um número inteiro vamos alterar o tipo de representação do "knob" para Inteiro 32 bits.

4.0 Arranjos (arrays)

Efectuar as seguintes ligações e testar.....

Questão: O que acontece quando se tenta aceder a um elemento do vector que ainda não foi definido?



Elementos que estão fora-de-jogo?

6.1 ARRAY de STRINGS

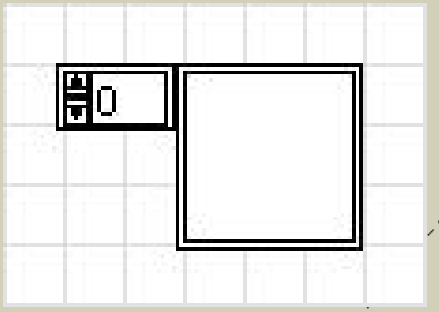
String:

É possível definir ARRAYS de outro tipo de dados ou até objectos.

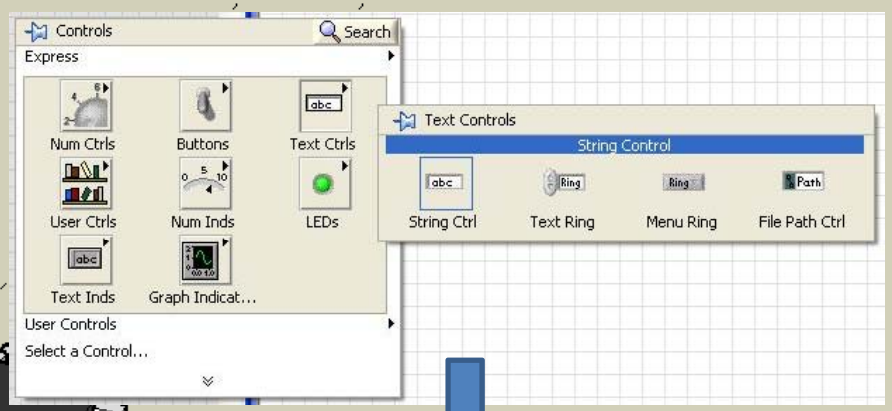
Um tipo muito útil é o ARRAY de STRINGS.

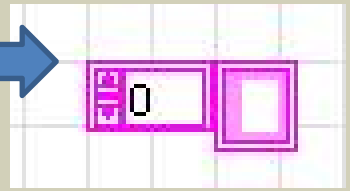
Criar um array de strings é semelhante à criação de um array numérico:

1º Colocar o objecto ARRAY no diagrama de blocos



2º Atribuir-lhe(1) o objecto "String Ctrl"





3º Preencher as posições pretendidas com caracteres alfanuméricos.

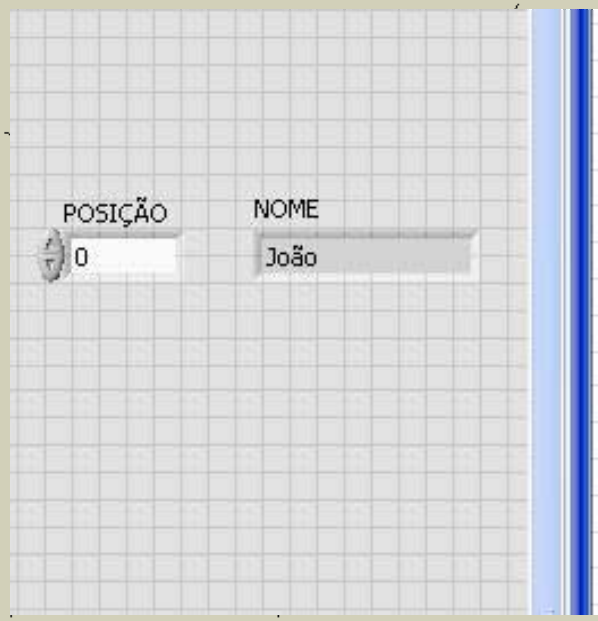
(1) "drag and drop" sobre o objecto ARRAY no diagrama de blocos.

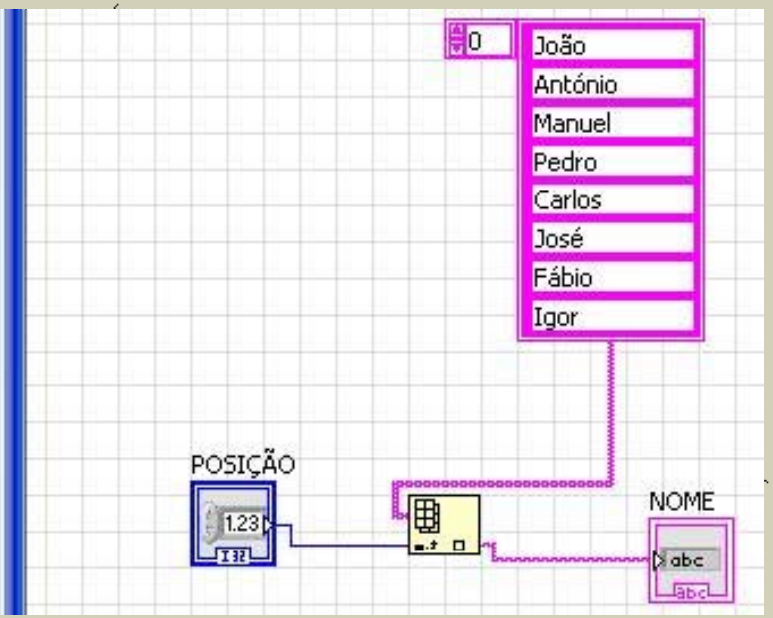
4.1 ARRAY de STRINGS

EXEMPLO:

1º Introduzir uma sequência de 8 nomes masculinos.

2º completar o programa de acordo com o esquema em baixo...





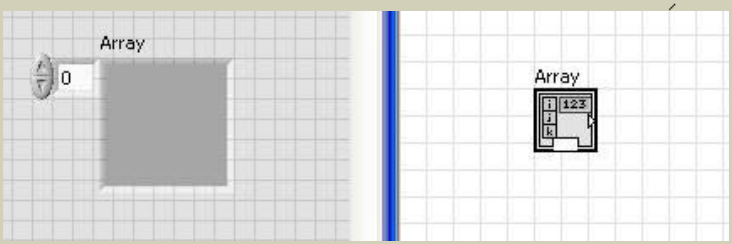
4.2 ARRAYS no FRONT PANEL

Até ao momento os ARRAYS foram considerados como CONSTANTES no diagrama de blocos.

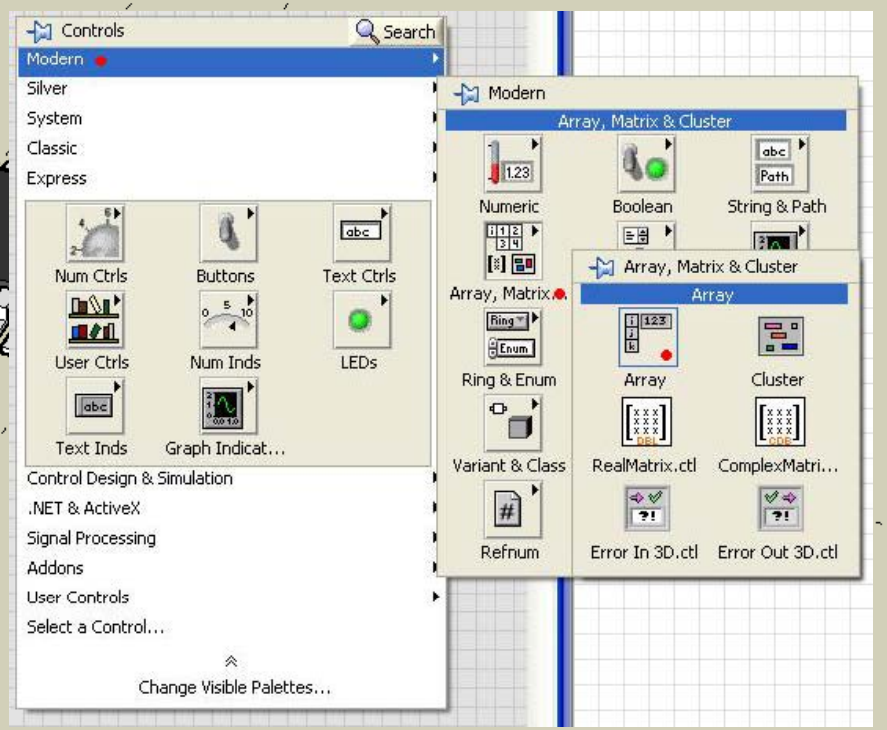
Alternativamente podem ser considerados OBJECTOS no FRONT PANEL

COMO?

No FRONT PANEL "clicar" com o botão direito do rato e aceder ao objecto ilustrado na figura ao lado...

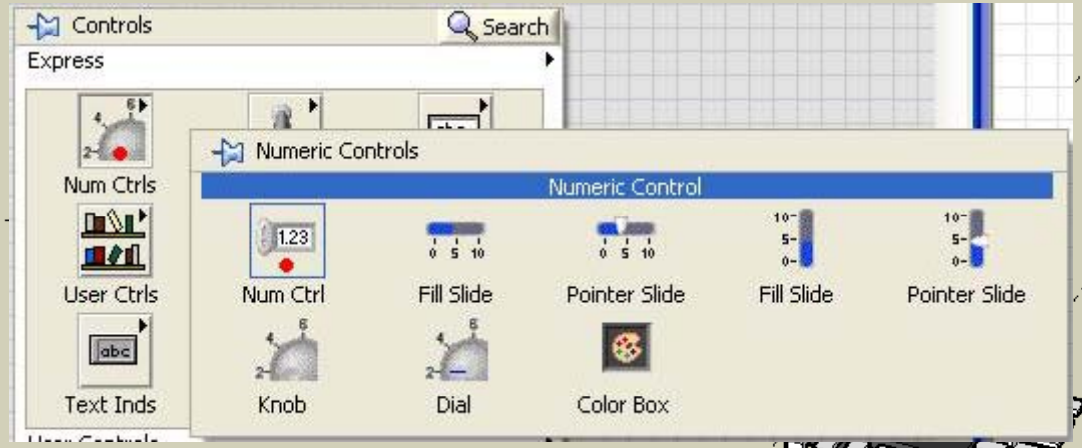


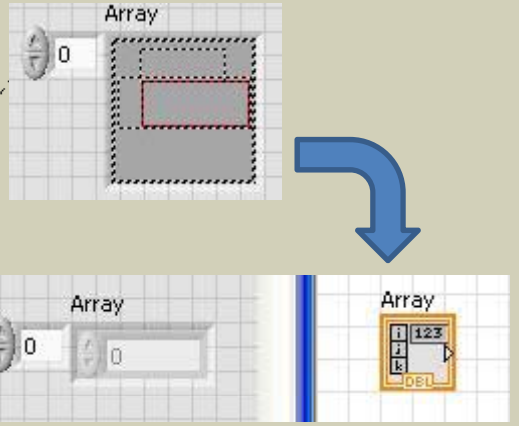
A cor PRETO indica que o ARRAY deve ser inicializado!



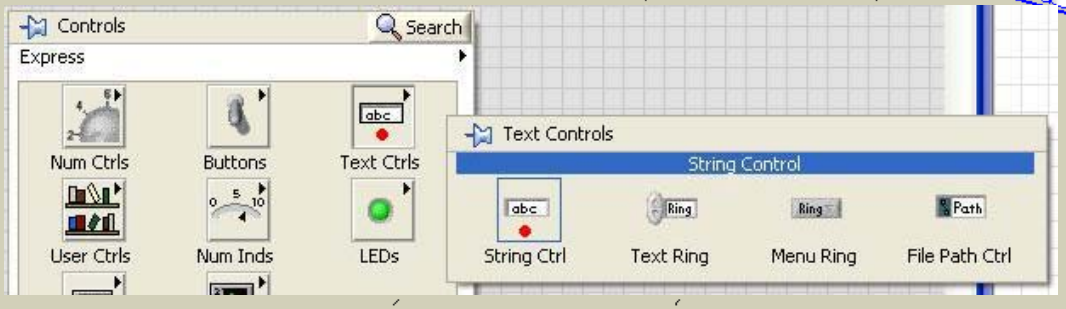
4.2 ARRAYS no FRONT PANEL

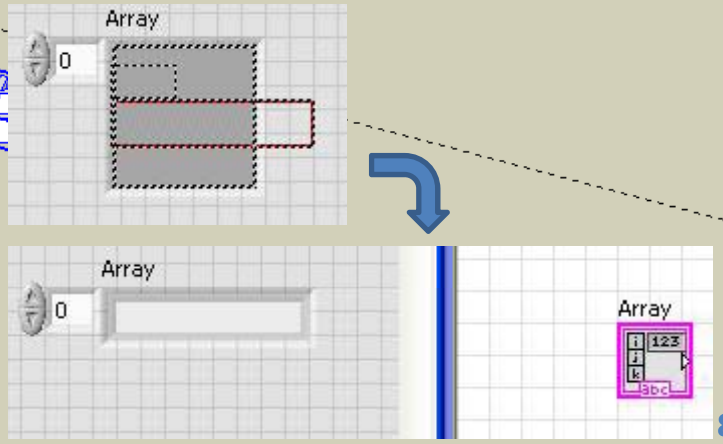
Inicialização com valores numéricos...





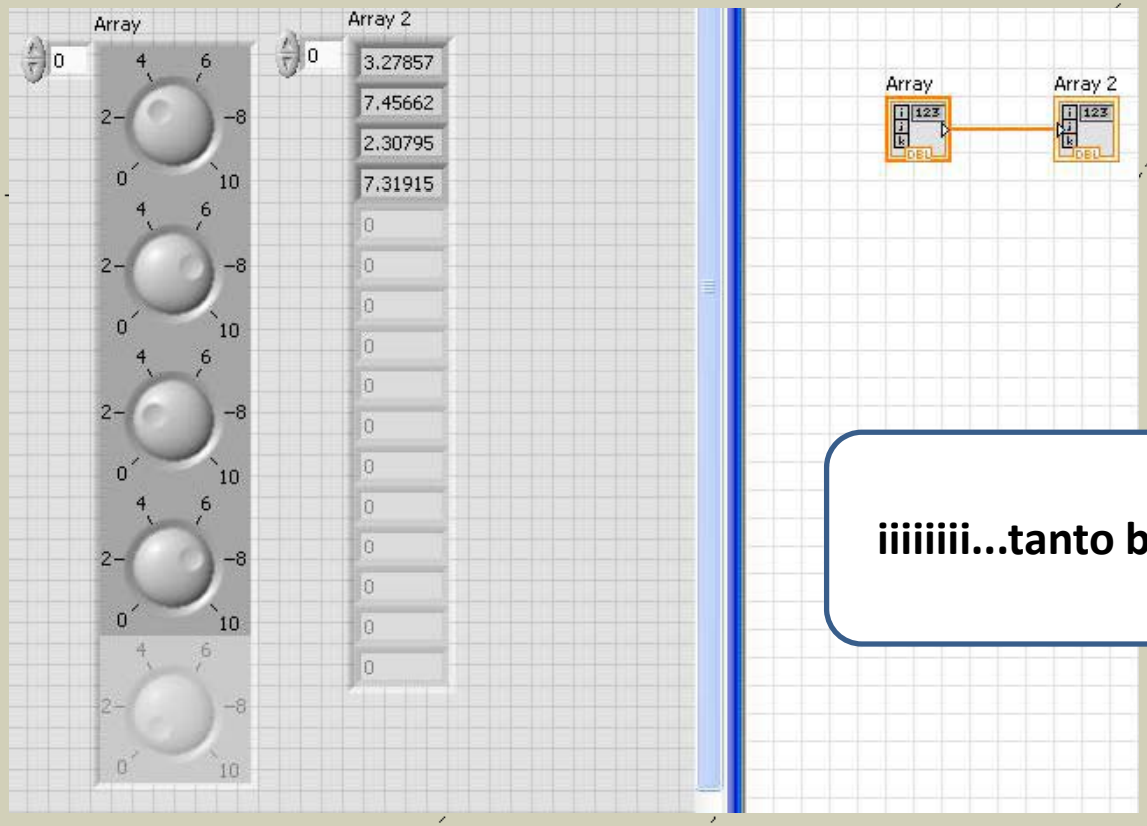
Inicialização com strings...



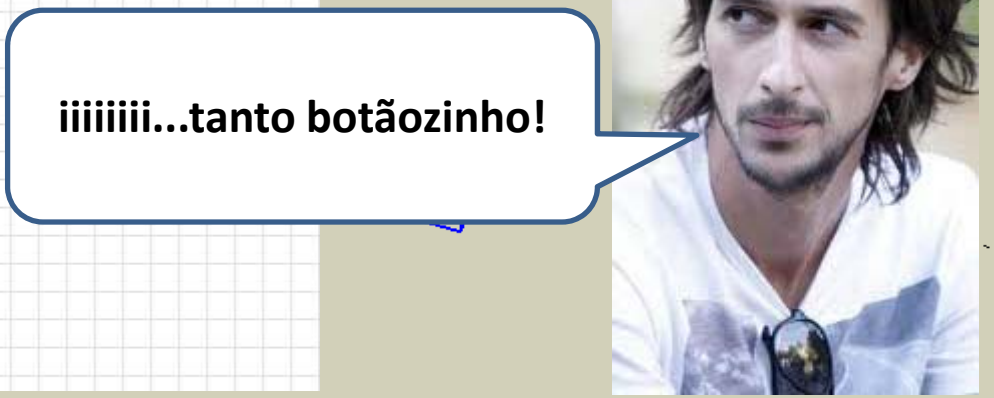


4.2 ARRAYS no FRONT PANEL

Inicialização com outros objectos...

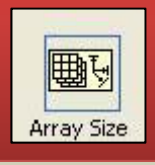


EXEMPLO: Implemente e teste o seguinte programa...

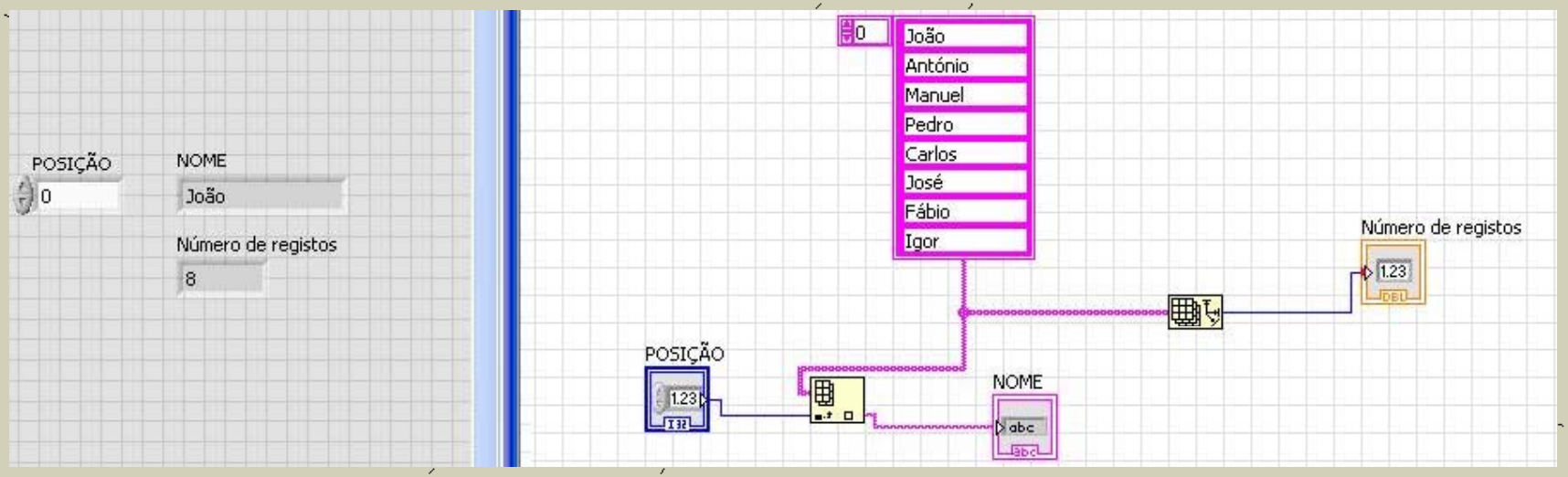


4.3 ARRAYS: operações

Algumas operações elementares que podem ser realizadas sobre ARRAYS:

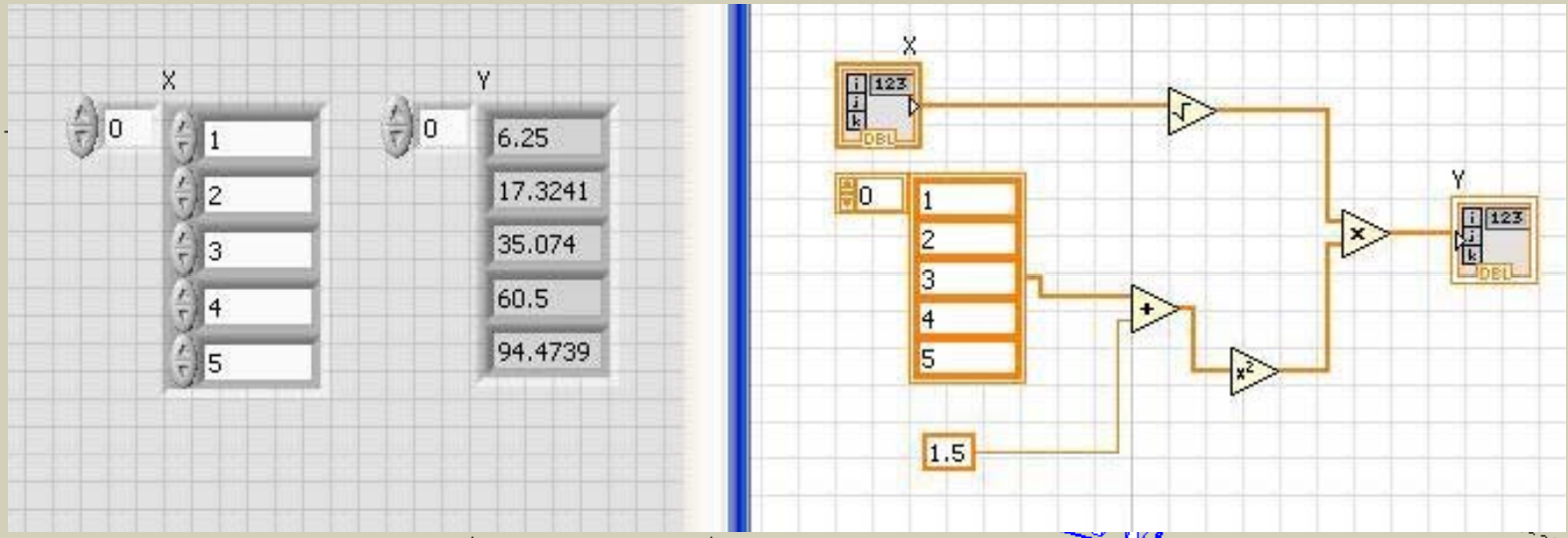


NÚMERO DE ELEMENTOS: retorna o número de elementos de um ARRAY



4.3 ARRAYS: operações

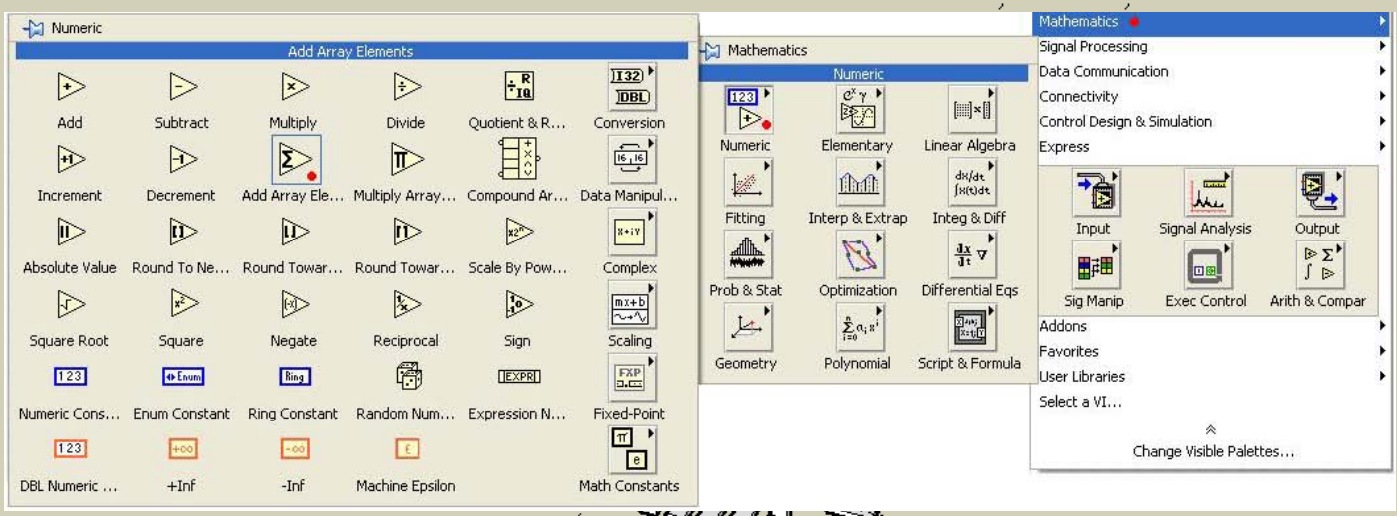
ARITMÉTICA ENTRE ARRAYS NUMÉRICOS



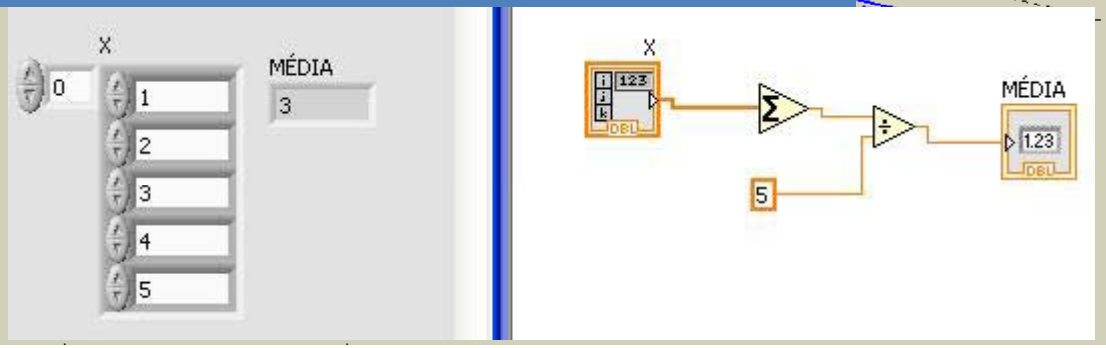
Exemplo: Calcular o valor médio de um vector

4.3 ARRAYS: operações

1º Aceder à operação "ADD ARRAY ELEMENTS"



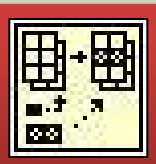
2º Efectuar as ligações...



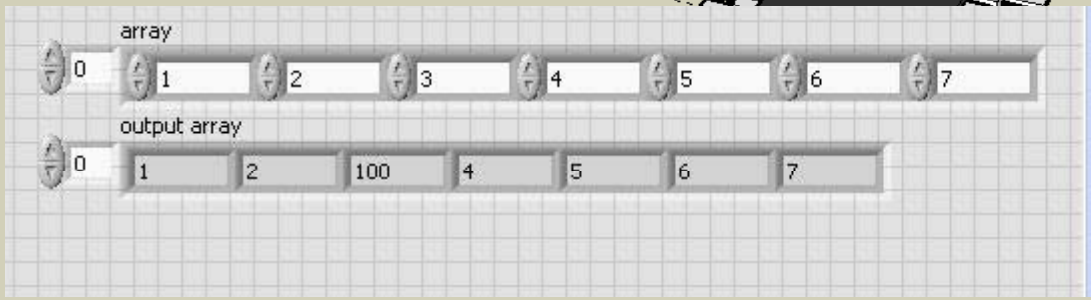
4.3 ARRAYS: operações

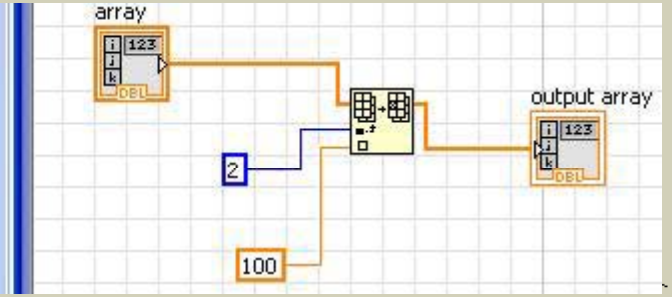
EXERCÍCIO: Modifique o programa anterior de modo a poder ser calculado automaticamente a média de um ARRAY de dimensão arbitrária

EXERCÍCIO: Desenhe um programa capaz de calcular o desvio padrão de um ARRAY



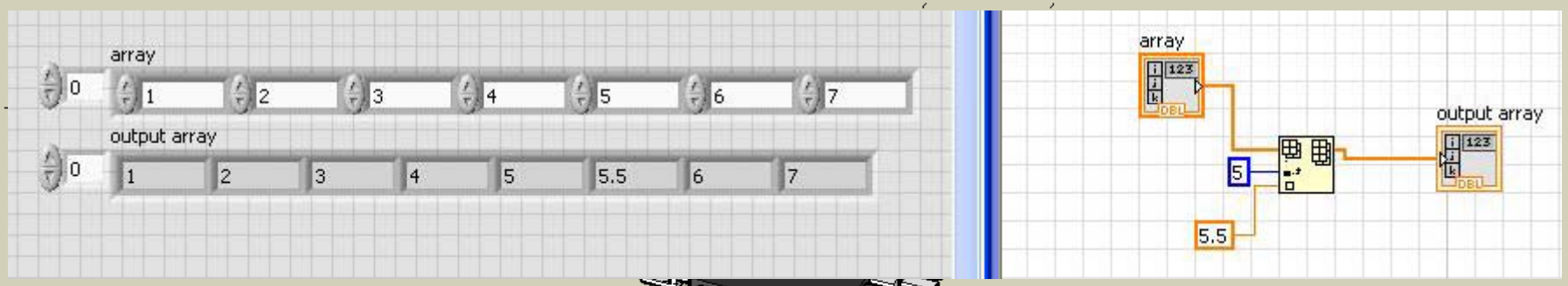
Alterar um elemento



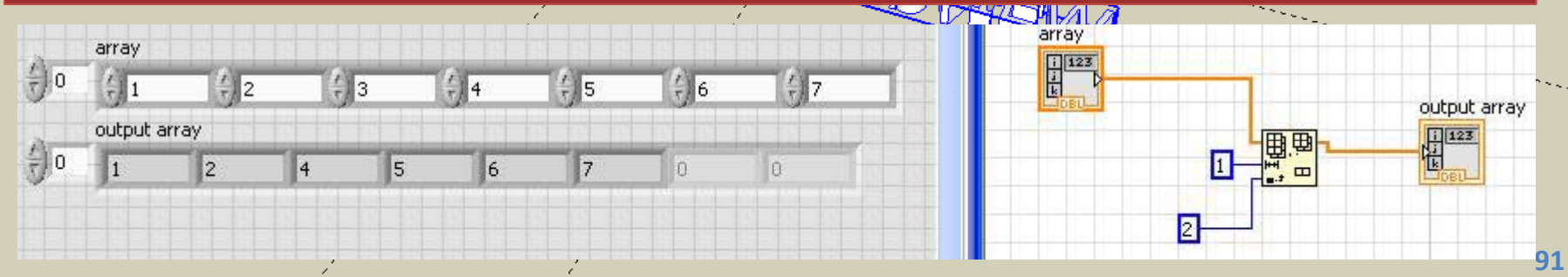


4.3 ARRAYS: operações

Adicionar um elemento



Remover um elemento



5.0 SUBVI's

O carácter modular do LabVIEW é potenciado com o conceito de SUBVI

Um SUBVI é um VI (instrumento virtual) utilizado "dentro" de outro VI

A utilização de SUBVI permite:

- **Tornar mais fácil a leitura e a depuração de um programa;**
- **Possibilita a reutilização de elementos de código já desenvolvidos.**

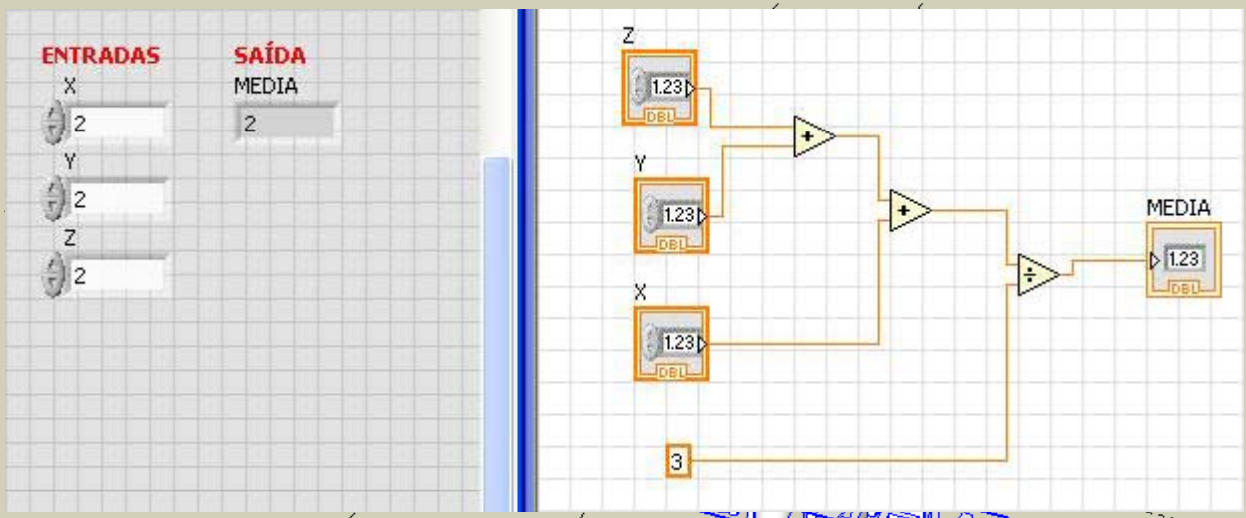
Será que vou conseguir entender esta estória de SUBVI's?

VI's podem ser "aninhados" em número arbitrário (VI's dentro de VI's que já envolvem outros VI's ...)

5.1 Construção de **SUBVI's**

EXEMPLO: SUBVI capaz de calcular a média de três valores.

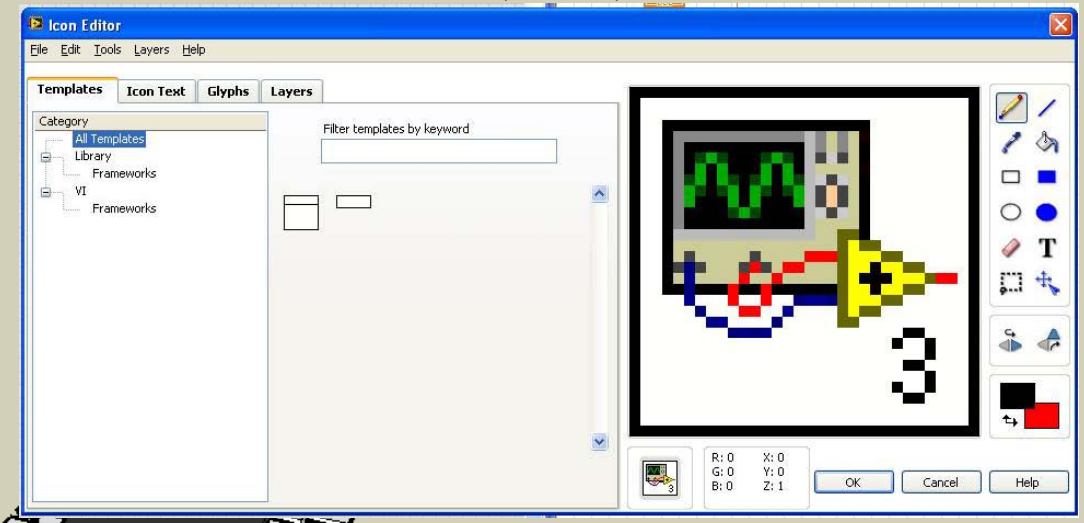
1º Construir o programa...



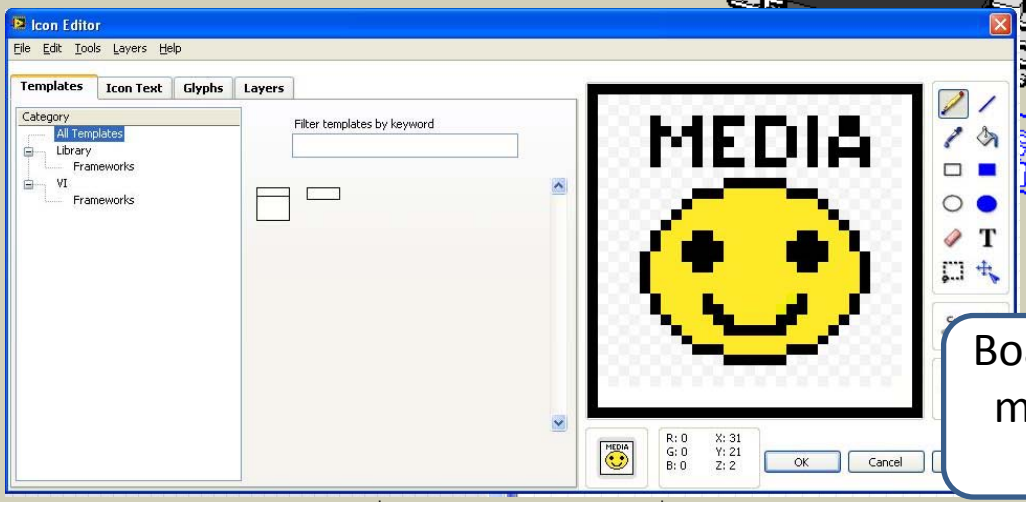
2º (opcional) Atribuir um ícone ao programa...

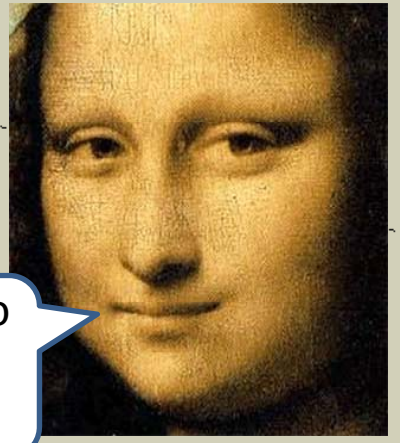
5.1 Construção de **SUBVI's**

Programa de desenho de ícones...



Uma proposta sorridente...





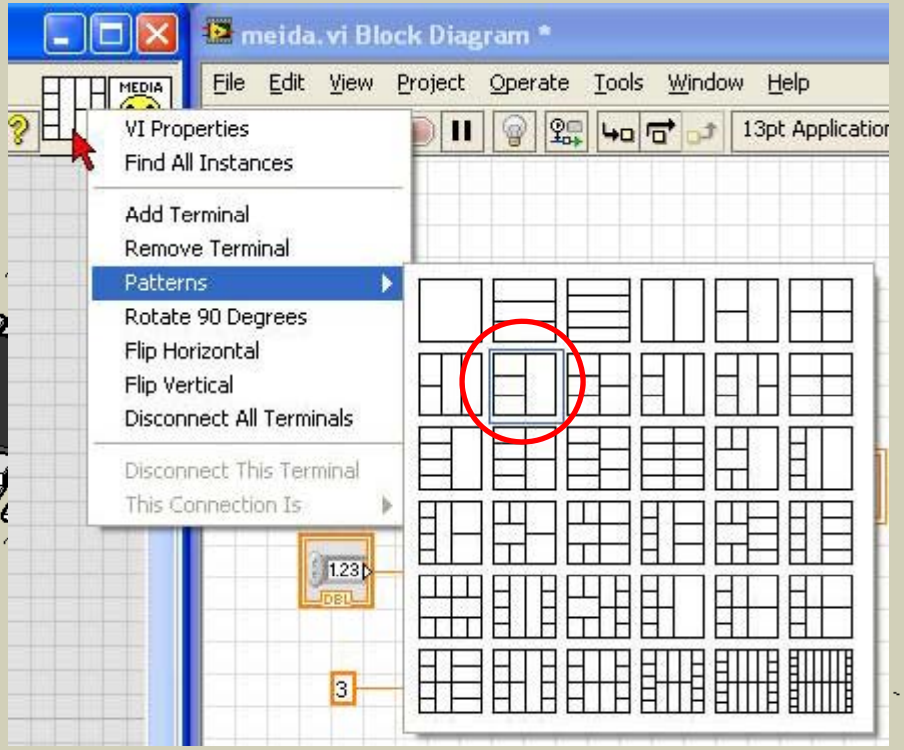
Boa tentativa...mas o meu sorriso é mais gracioso....

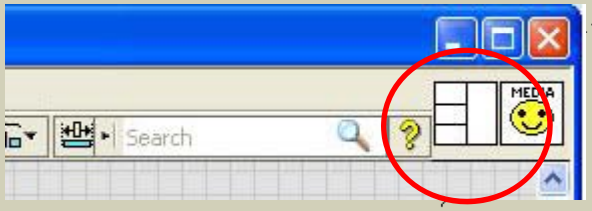
5.1 Construção de **SUBVI's**

3º atribuir sinais de entrada e saída...

Seleccionar no "connector pane" a topologia que mais se adapta ao programa.

Por convenção os terminais mais à esquerda são associados a "entradas" e os terminais mais à direita são associados às "saídas"



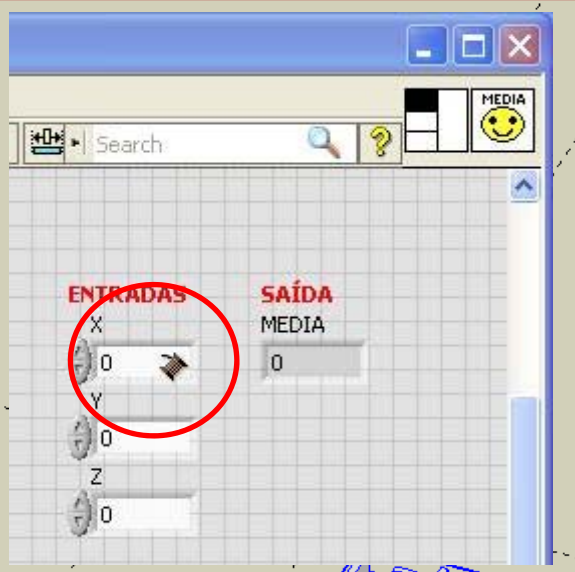


É normalmente considerada boa prática escolher um padrão de terminais que possua mais pares entrada/saída do que as necessárias de forma a contemplar possíveis alterações no SUBVI.

5.1 Construção de **SUBVI's**

Atribuir terminais aos "controlos" e "indicadores"

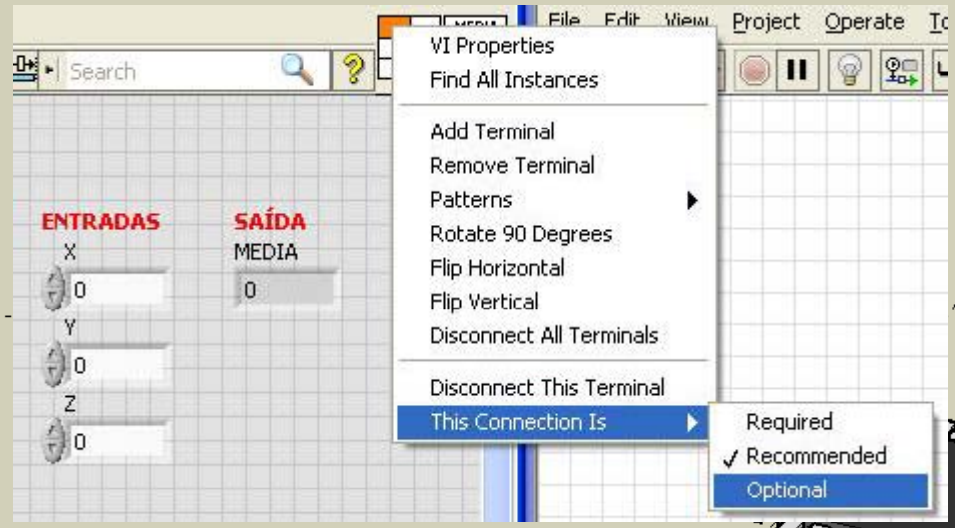
seleccionar no "**connector pane**" um dos terminais disponíveis (o terminal passa de "branco" para "preto")



seleccionar no "**Front Panel**" qual o indicador/controlo a associar a esse terminal

O terminal recentemente associado passa a apresentar a cor laranja.

5.1 Construção de **SUBVI's**



Cada terminal associado pode ter um de três modos distintos:

- **Necessário**
- **Recomendado**
- **Opcional**

Por defeito o valor é recomendado...

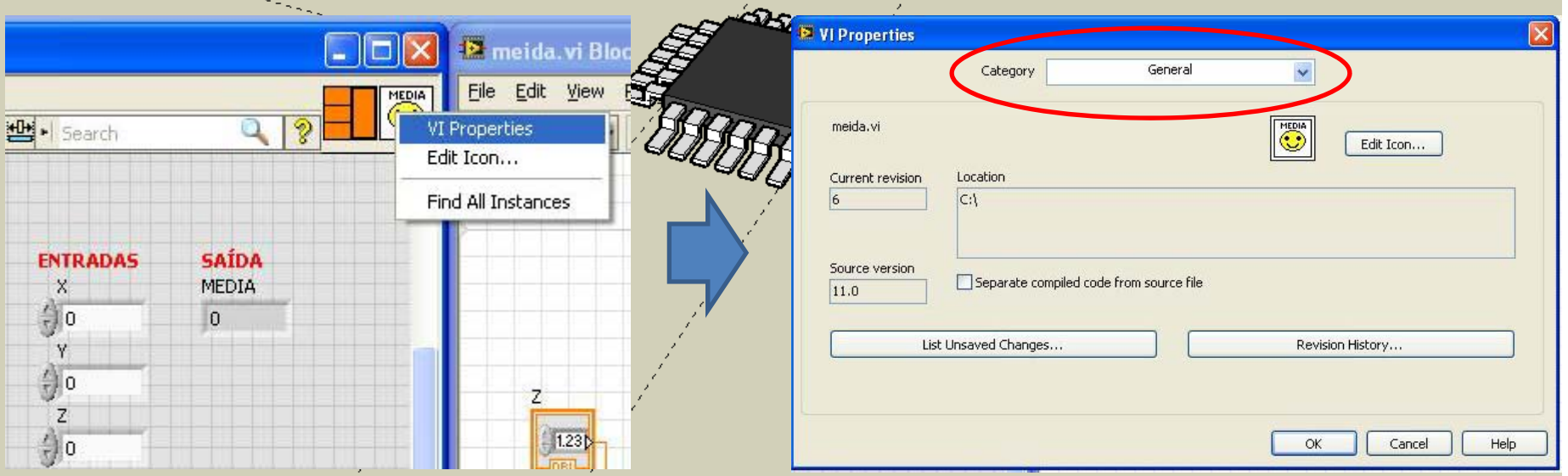
Aspecto do "connector pane" após todas as entradas e saídas serem associadas a um terminal...

5.1 Construção de **SUBVI's**

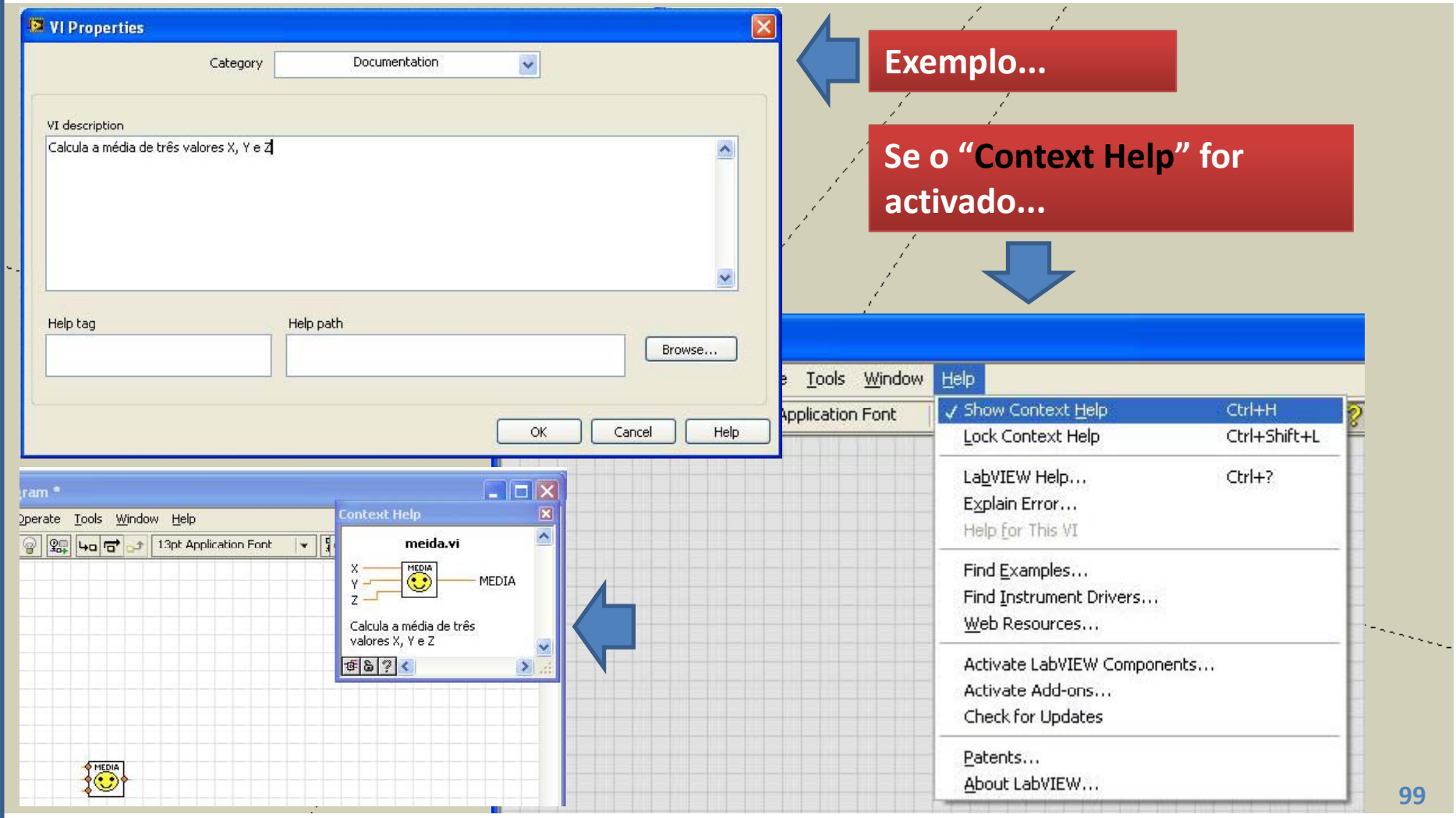
4º (opcional mas recomendado) documentar o programa ...

"click" no botão direito do rato sobre "connector pane" ou sobre o ícone e seleccionar "VI Properties"

Na janela "VI Properties" escolher em Category o item "Documentation"



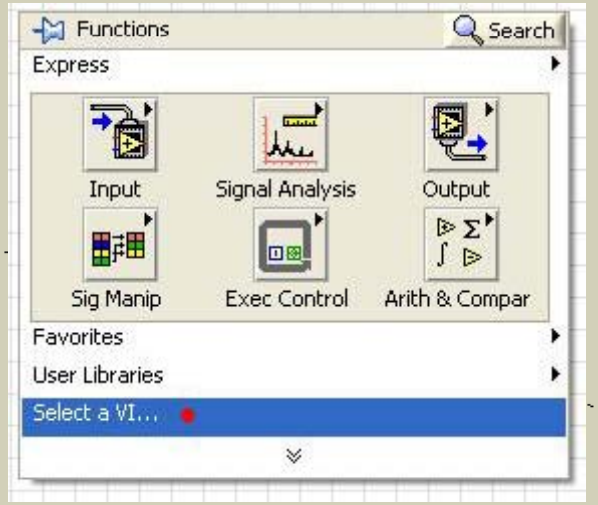
5.1 Construção de **SUBVI's**



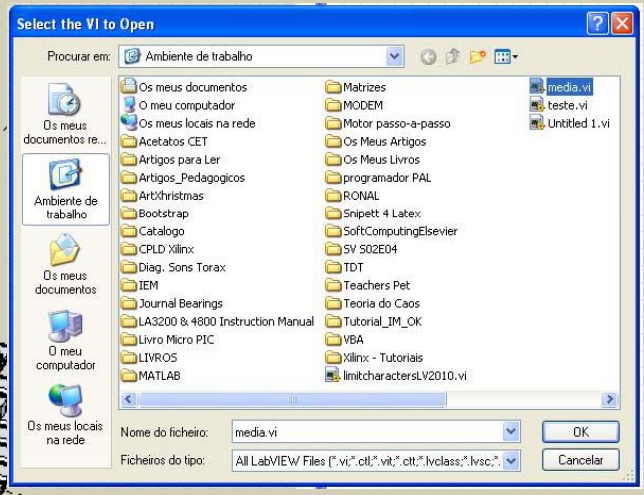
5.1 Construção de **SUBVI's**

Inserir e utilizar um SUBVI num VI:

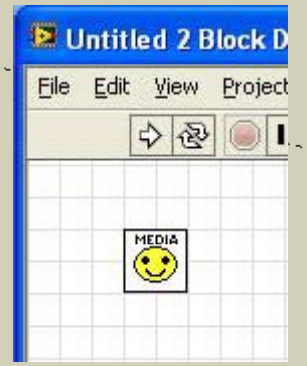
1º



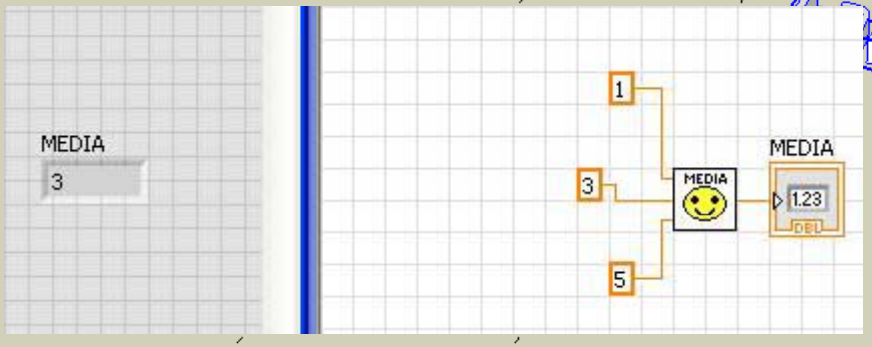
2º



3º



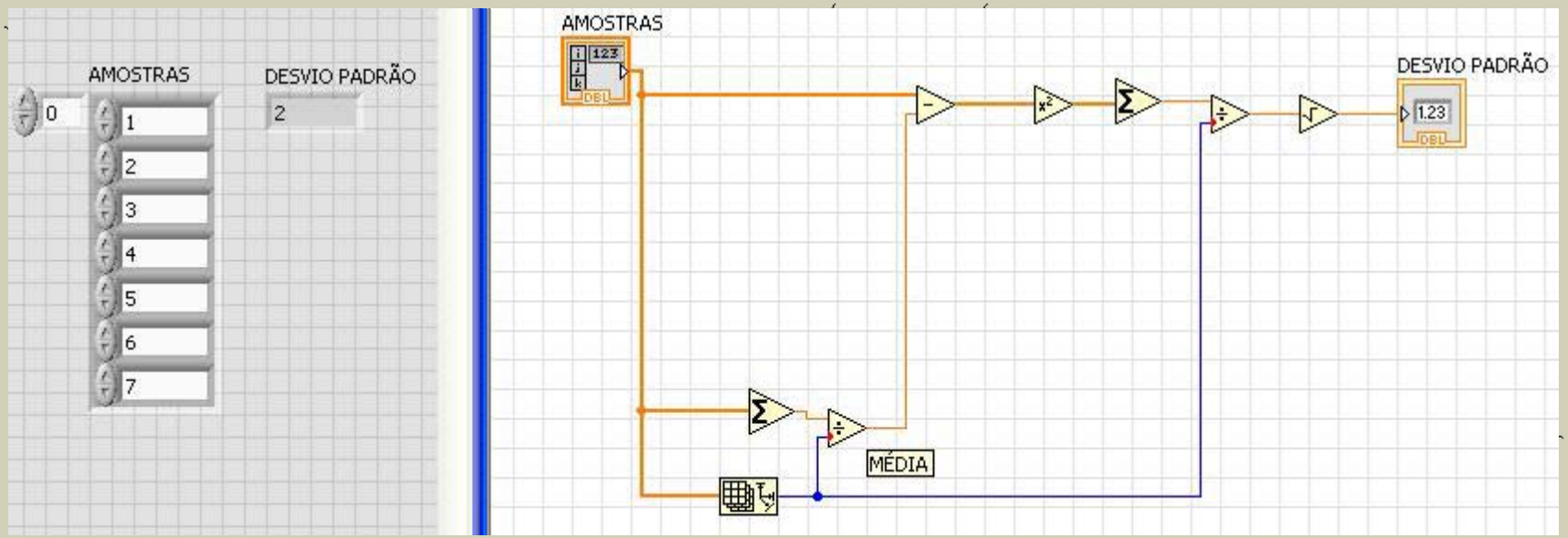
4º



5.1 Construção de **SUBVI's**

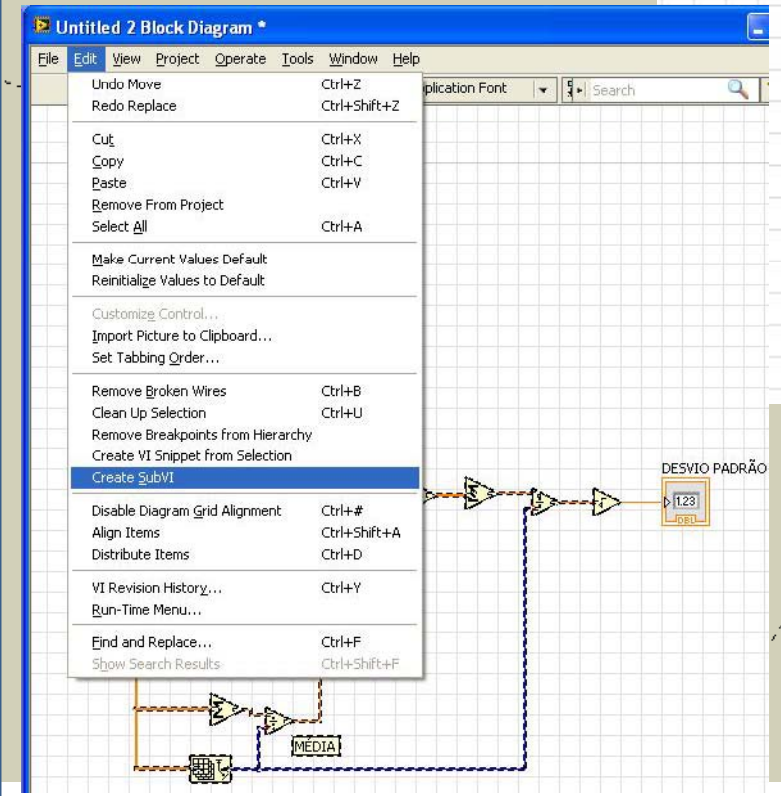
Criar um **SUBVI** a partir de um excerto de um digrama de blocos...

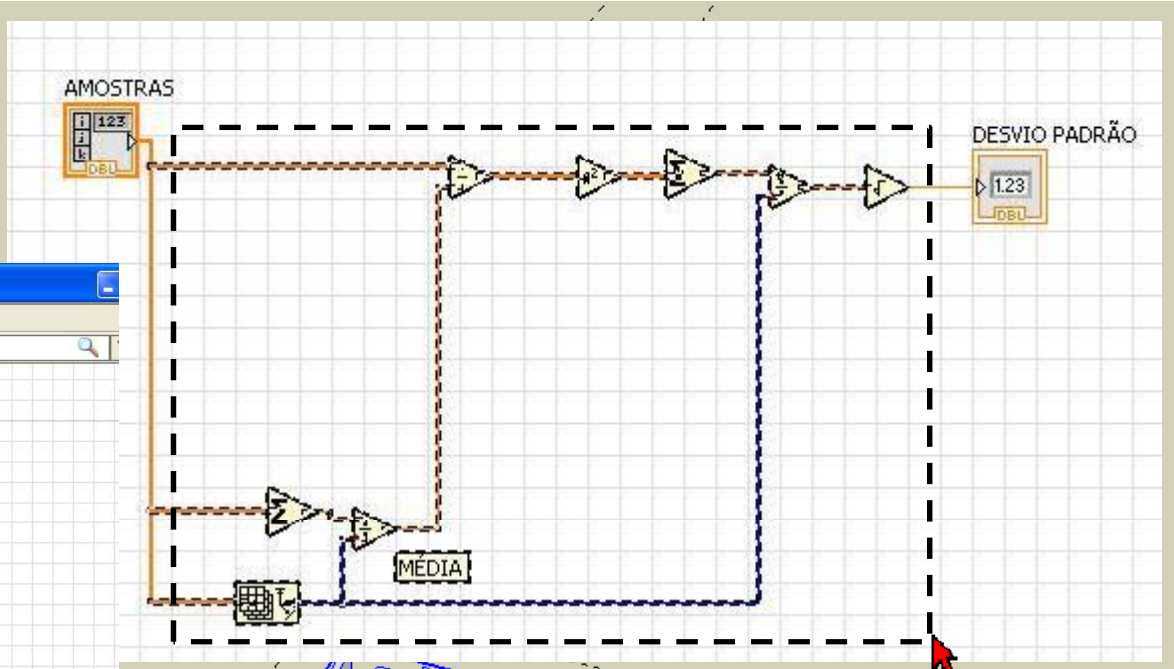
Considerar o seguinte VI...



5.1 Construção de **SUBVI's**

Seleccionar a fracção do diagrama de blocos que se pretende substituir.

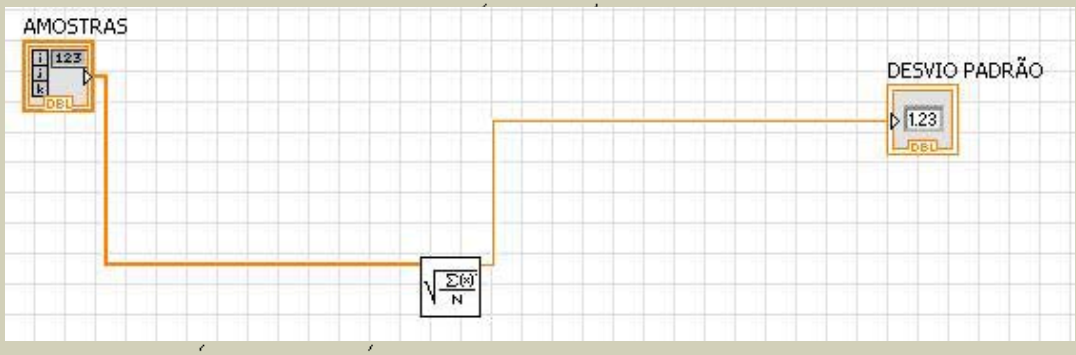




No menu "Edit" escolher "Create SubVI".

5.1 Construção de **SUBVI's**

Aspecto final já com novo ícone.



Deo Gratias... Estou sem palavras....

5.2 Exercícios

***EX 13**: Construa um SubVI capaz de calcular as quatro operações aritméticas elementares entre dois vectores (o produto e a divisão dizem respeito ao produto de Hadamard entre os dois vectores ou ente um dos vectores e outro obtido a partir do inverso dos elementos do segundo)*

***EX 14**: Utilize o SubVI desenhado anteriormente para calcular a seguinte sequência de operações entre os vectores V1, V2 e V3:*

*(V1+V2*V3)/(V4+1)*

***EX 15**: Construa um SubVI capaz de gerar um número **inteiro** aleatório entre 1 e 49*

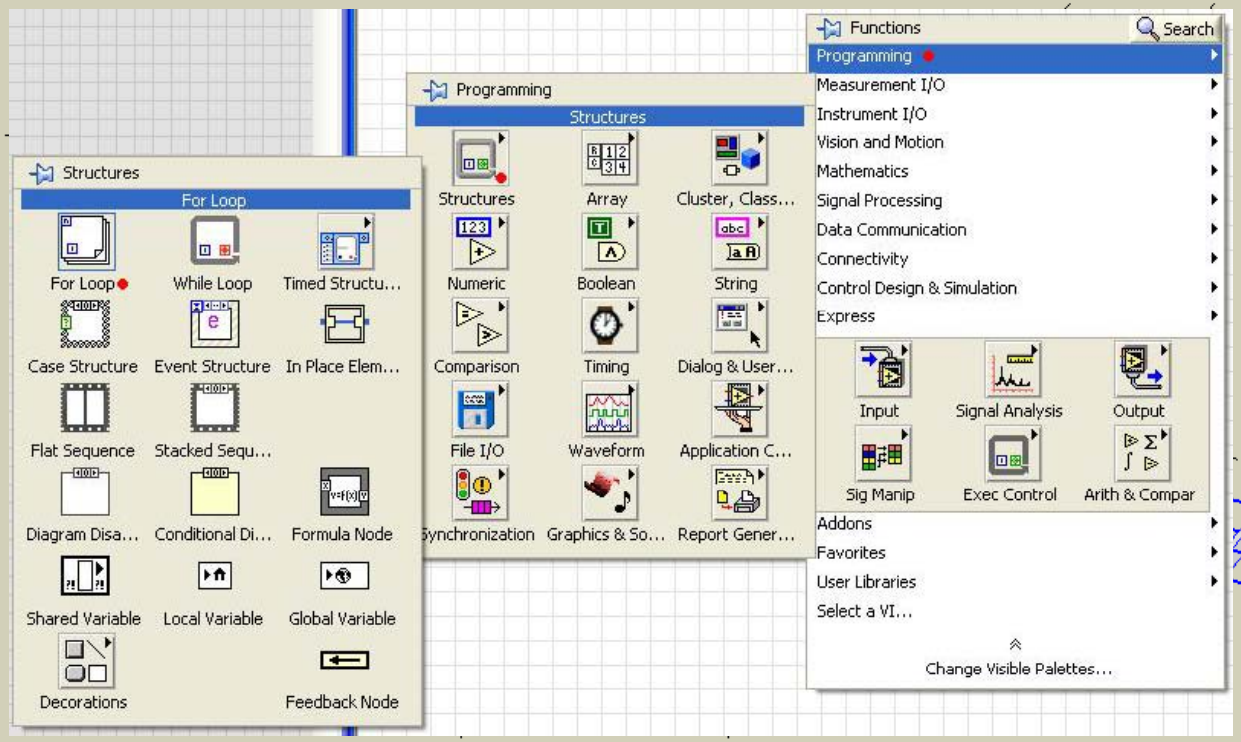
6.0 Estruturas de Controlo

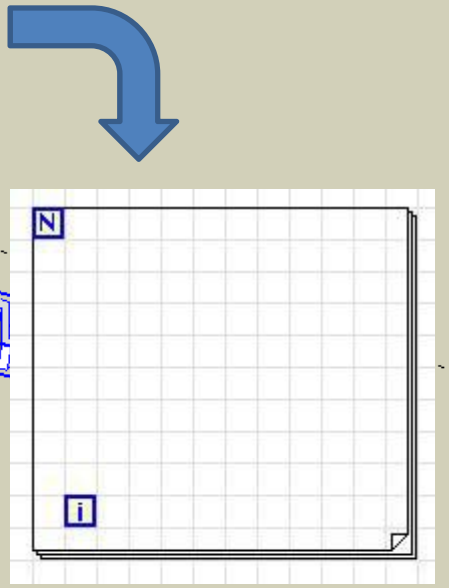
- Para além dos controlos, indicadores e VI, um diagrama de blocos pode também conter sequências para controlo de fluxo de dados.
- O LabVIEW suporta, entre outras, as seguintes estruturas de controlo:
  - Ciclos **FOR**
  - Ciclos **WHILE**
  - Estruturas condicionais **CASE**

Afinal o LabVIEW é que é o *SPECIAL ONE*

6.1 Ciclos **FOR**

Utilizam-se sempre que se conhece, à partida, o número de ciclos a executar.



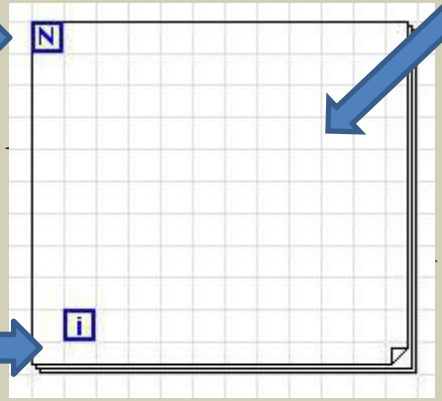


6.1 Ciclos **FOR**

Um ciclo **FOR** possui dois terminais:

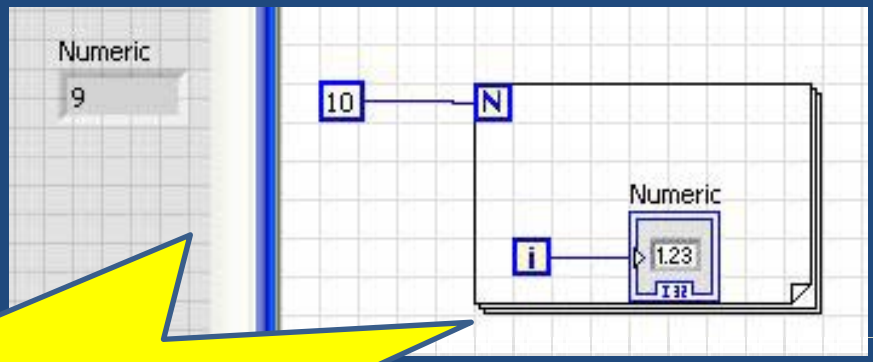
Terminal de contagem.

Código que deve ser executado repetidamente (subdiagrama).



Terminal de iteração.

Exemplo:

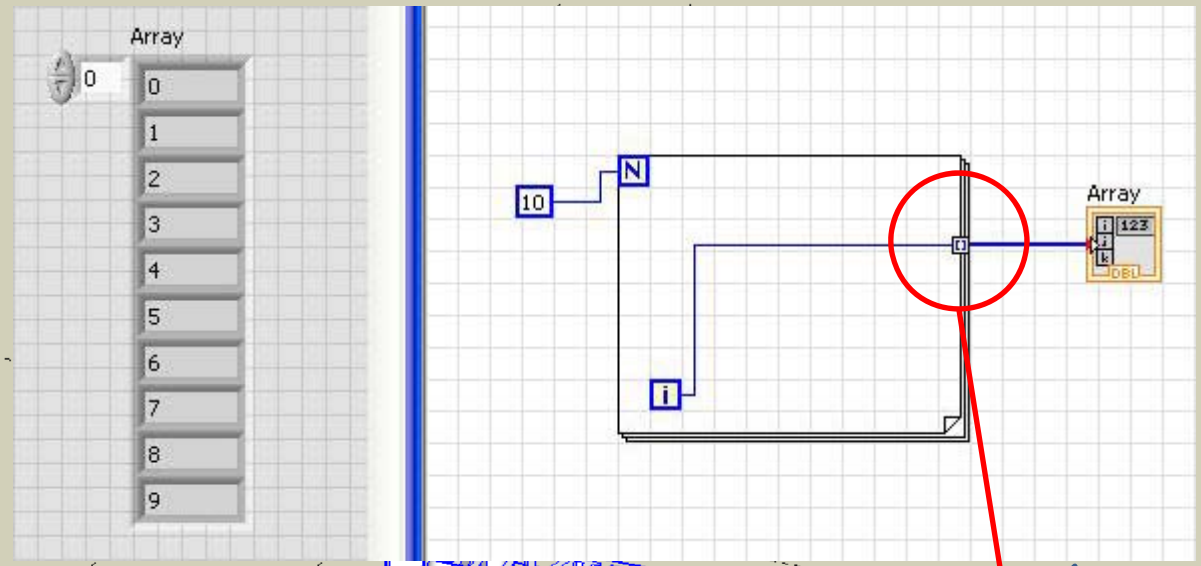


Execute com Highlight Execution ligado

6.1 Ciclos **FOR**: túneis

O ciclo FOR é frequentemente utilizado para inicializar ARRAYS.

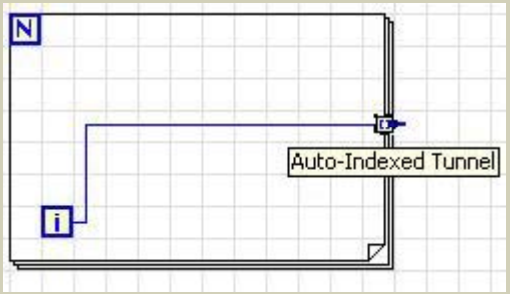
O programa indicado ao lado preenche um ARRAY com os números inteiros de 0 a 9.



Os túneis permitem a passagem de informação através do ciclo.

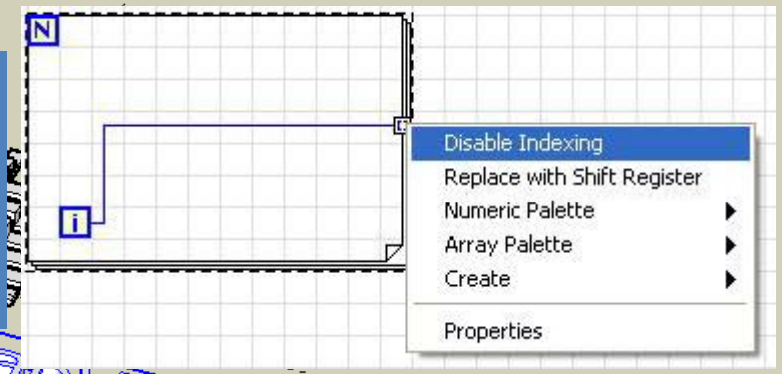
Um túnel pode possuir a indexação automática habilitada ou não.

6.1 Ciclos **FOR**: túneis

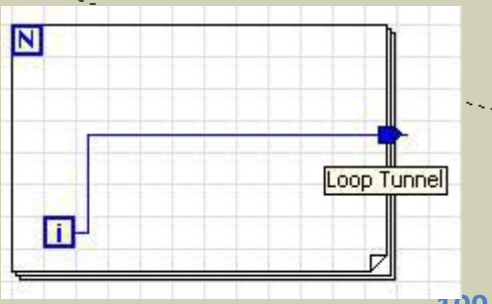


Quando a **indexação automática** se encontra **activa** o valor calculado num conector é enviado através do túnel em cada iteração do ciclo.

Quando a indexação automática se encontra **inactiva** o valor calculado num conector é enviado através do túnel apenas após a conclusão do ciclo.

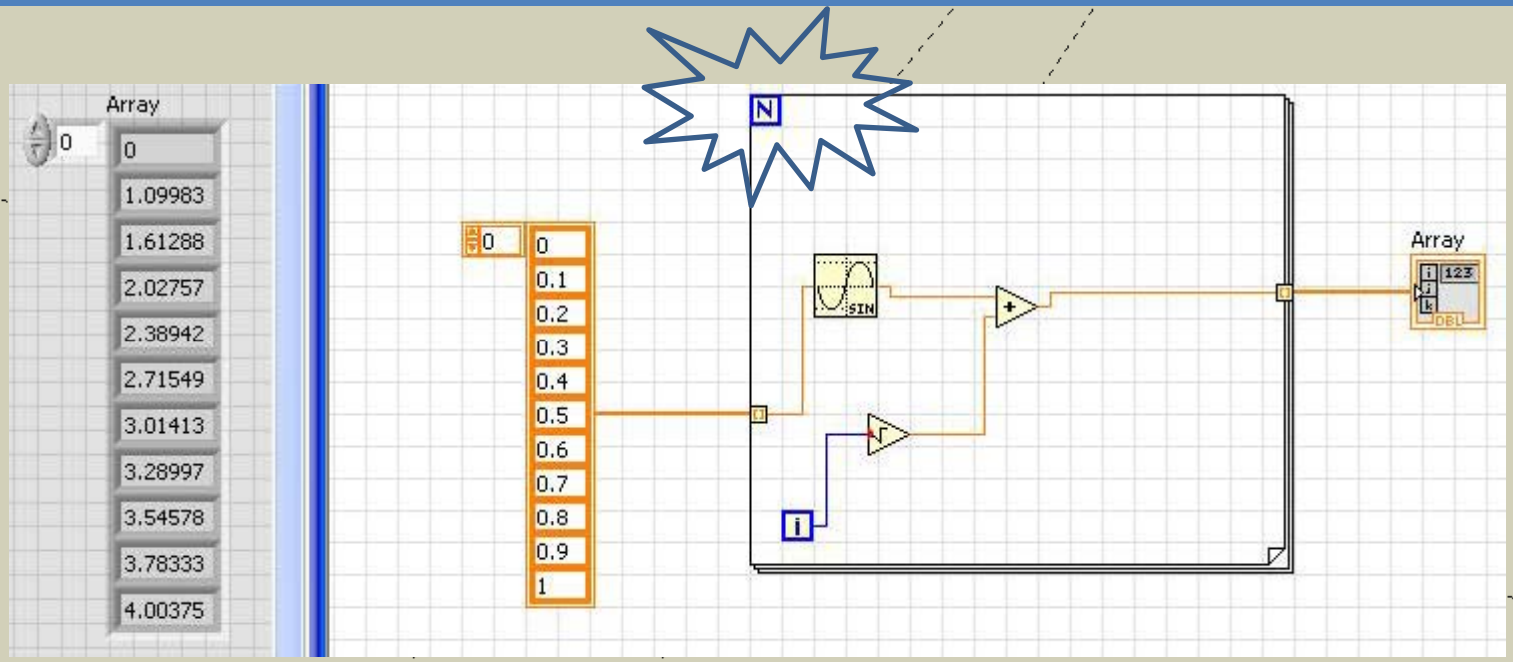


Na primeira situação a saída do túnel é um **ARRAY** e na segunda a saída é um **ESCALAR**.



6.1 Ciclos **FOR**: túneis

O valor de N (número de iterações do ciclo) pode não ser explicitado no caso de se enviar dados, por túnel, para o interior do ciclo.



Quando a indexação se encontra ACTIVA no túnel de entrada, o valor de N é admitido como sendo igual ao número de elementos do ARRAY.

6.1 Exercícios

***EX 16**: Construa um VI capaz de preencher um vector com os primeiros 100 valores da função:*

$$f(i)=\sin(pi*i/100)$$

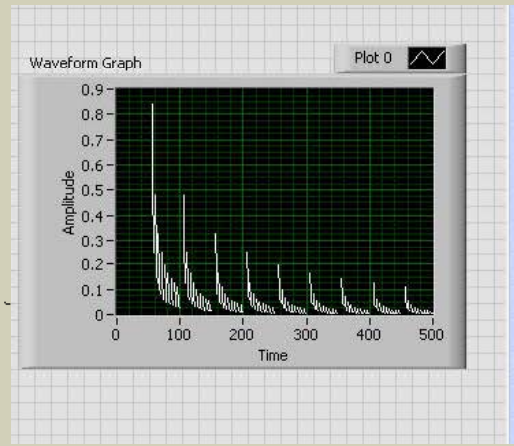
onde i é um inteiro de 0 a 99

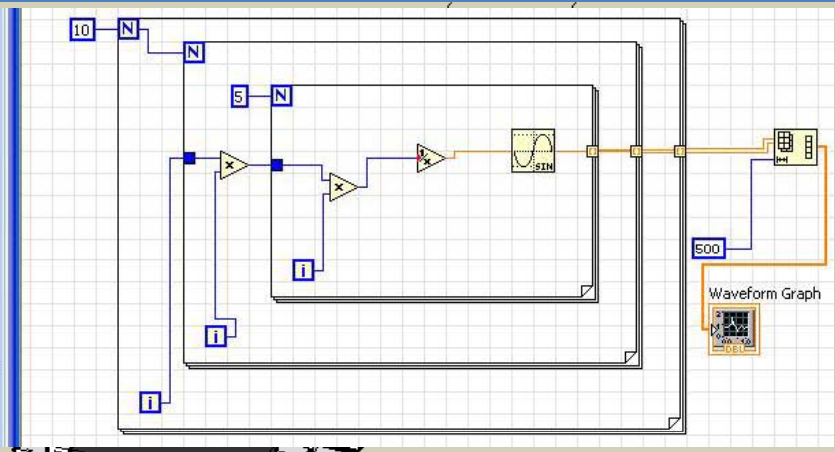
***EX 17**: Repita o mesmo exercício mas agora o valor i deve começar em -5. Para além disso um gráfico da função deve ser traçado.*

***EX 18**: Construa um VI que preencha um ARRAY numérico com 10 valores aleatórios inteiros entre 1 e 49. (sugestão: utilize o subVI que desenvolveu para o exercício 15)*

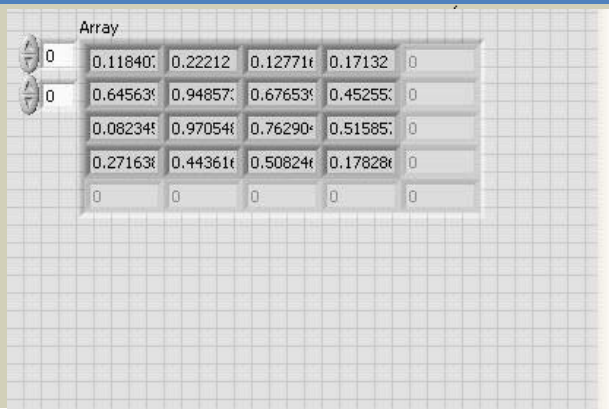
6.2 Ciclos **FOR**: aninhados

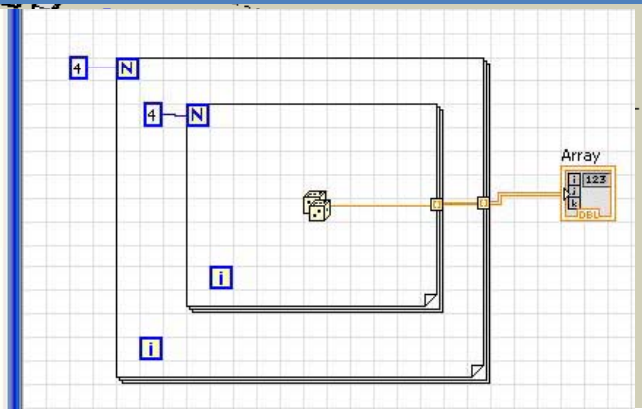
É possível colocar ciclos FOR no interior de outros ciclos FOR.





...utilizar esta estratégia para preencher ARRAYS multidimensionais...



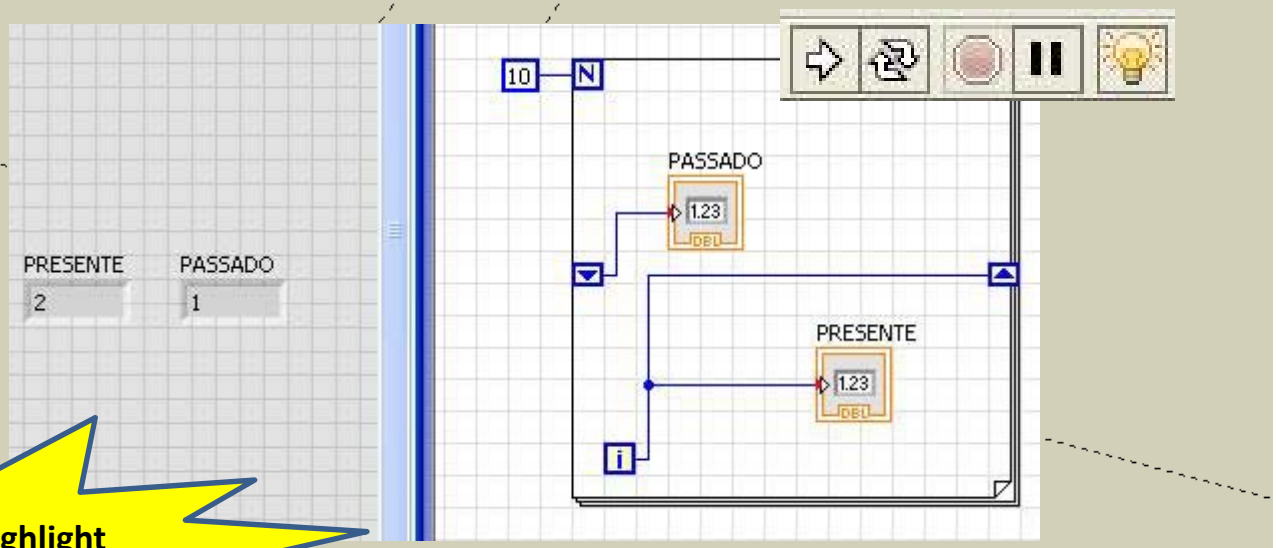


6.3 Ciclos **FOR**: Registos de Deslocamento

Frequentemente existe a necessidade de se utilizarem valores calculados numa iteração anterior do ciclo FOR.

O modo de aceder a valores passados consiste na criação de registos de deslocamento (Shift Registers).

EXEMPLO:



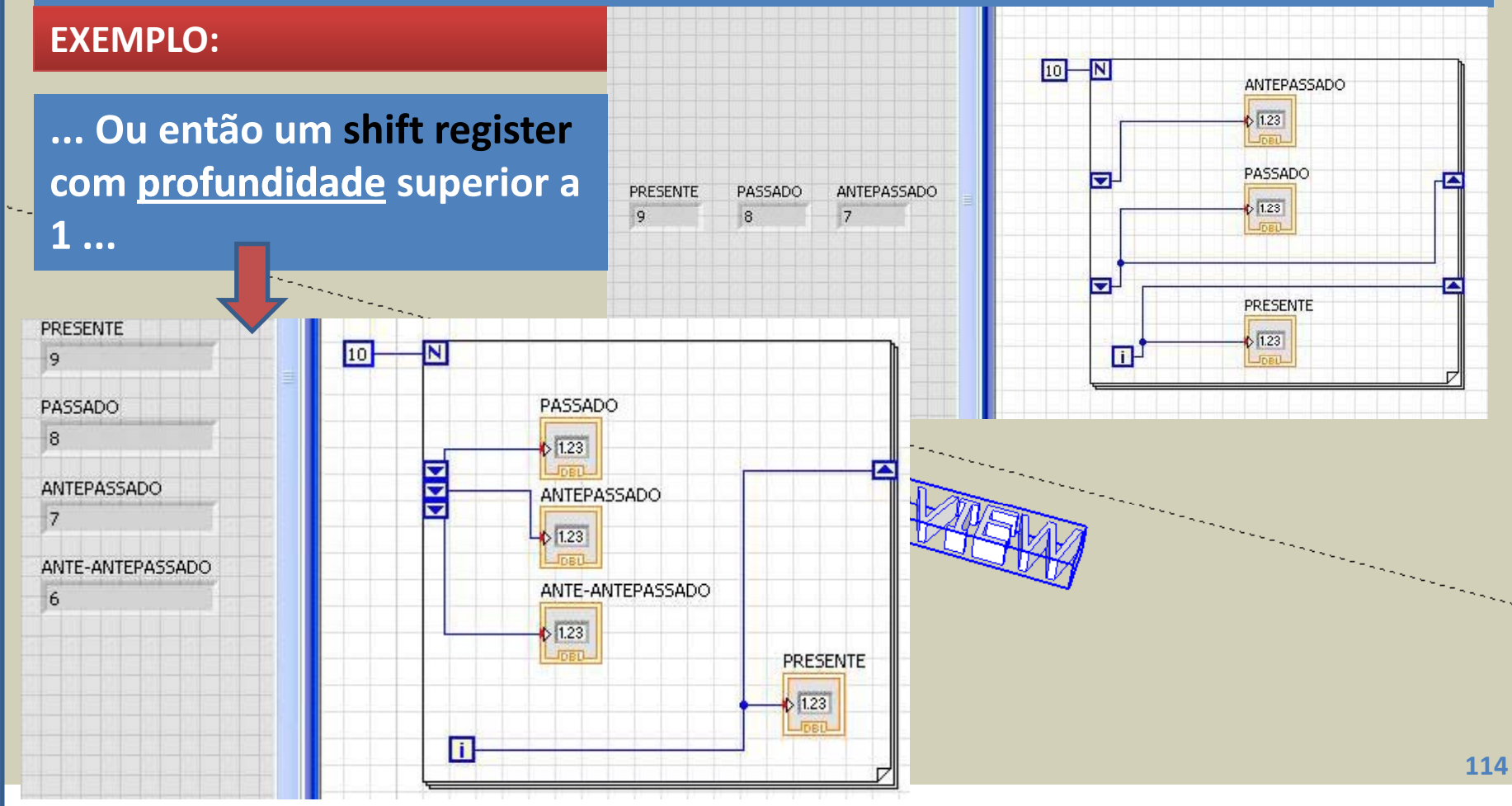
Execute com Highlight Execution ligado

6.3 Ciclos **FOR**: Registos de Deslocamento

Podem ser adicionados um número arbitrário de shift registers.

EXEMPLO:

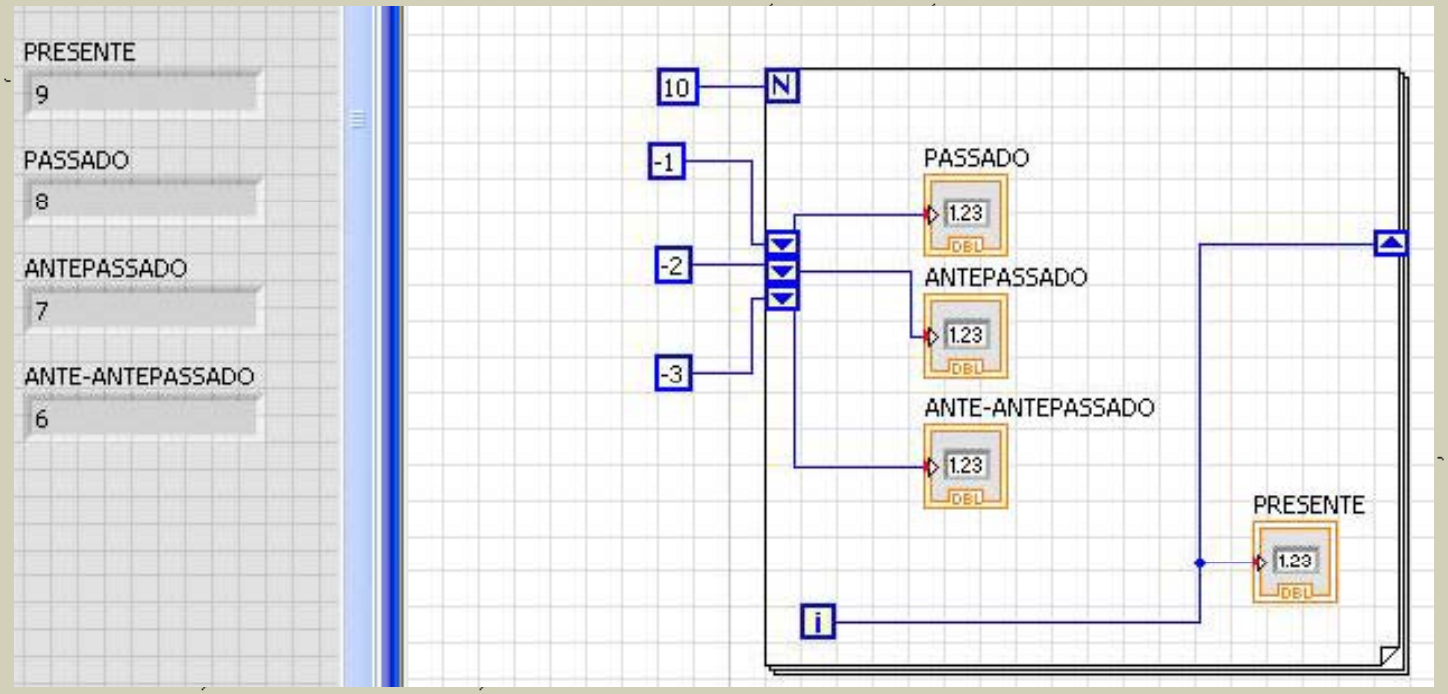
... Ou então um shift register com profundidade superior a 1 ...



6.3 Ciclos **FOR**: Registos de Deslocamento

Os shift registers podem ser inicializados com valores diferentes de 0.

EXEMPLO:



6.3 Exercícios

***EX 19**: Construa um ARRAY bidimensional, utilizando ciclos **FOR**, com o seguinte aspecto:*

***EX 20**: Faça um VI capaz de apresentar num **ARRAY** os k primeiros números da sequência de **Fibonacci**:*

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

*O valor de k deve ser introduzido pelo utilizador a partir de um **controlo numérico** e deve ser superior a 2.*

6.4 Ciclos **WHILE**

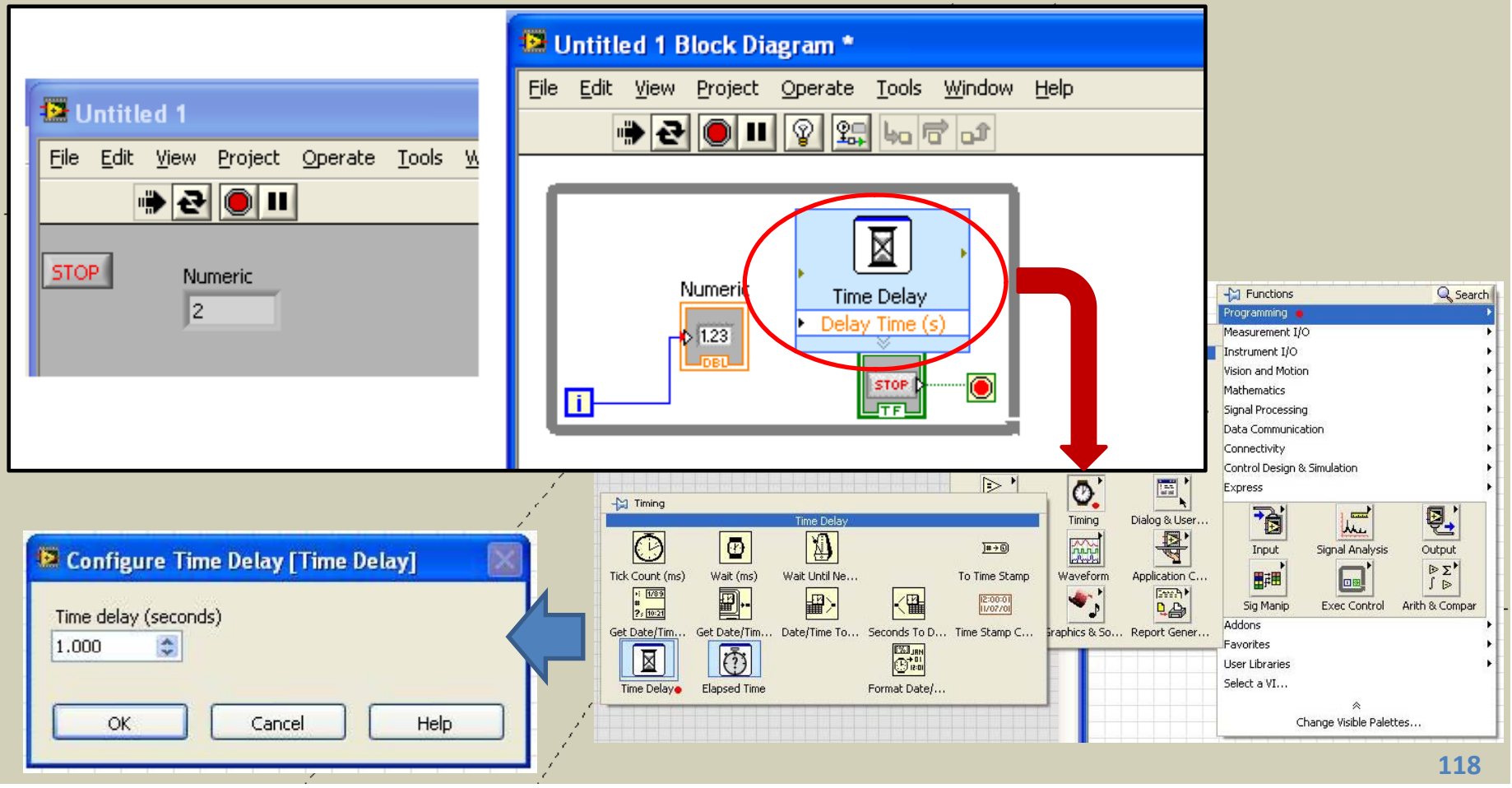
Executa um conjunto de operações ATÉ que uma condição lógica seja satisfeita.

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.

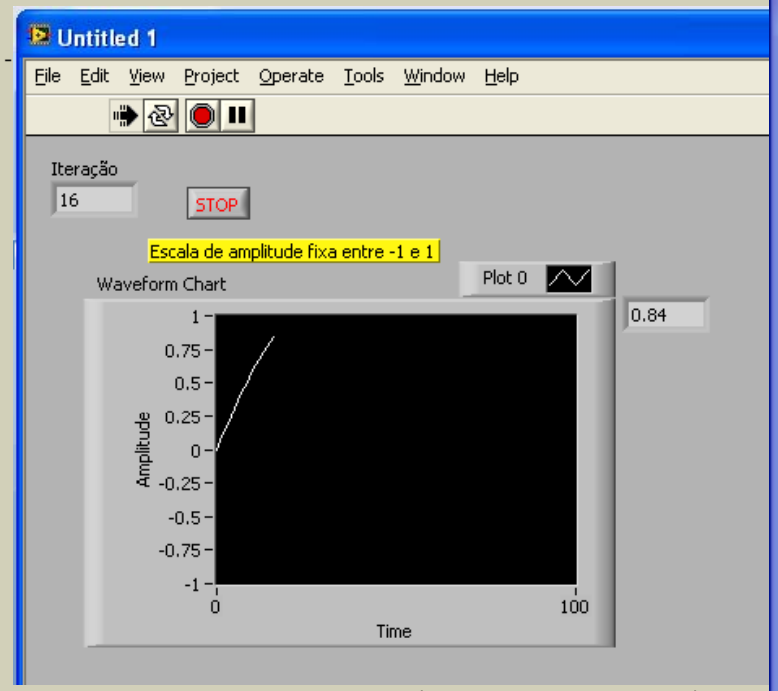
6.4 Ciclos **WHILE**

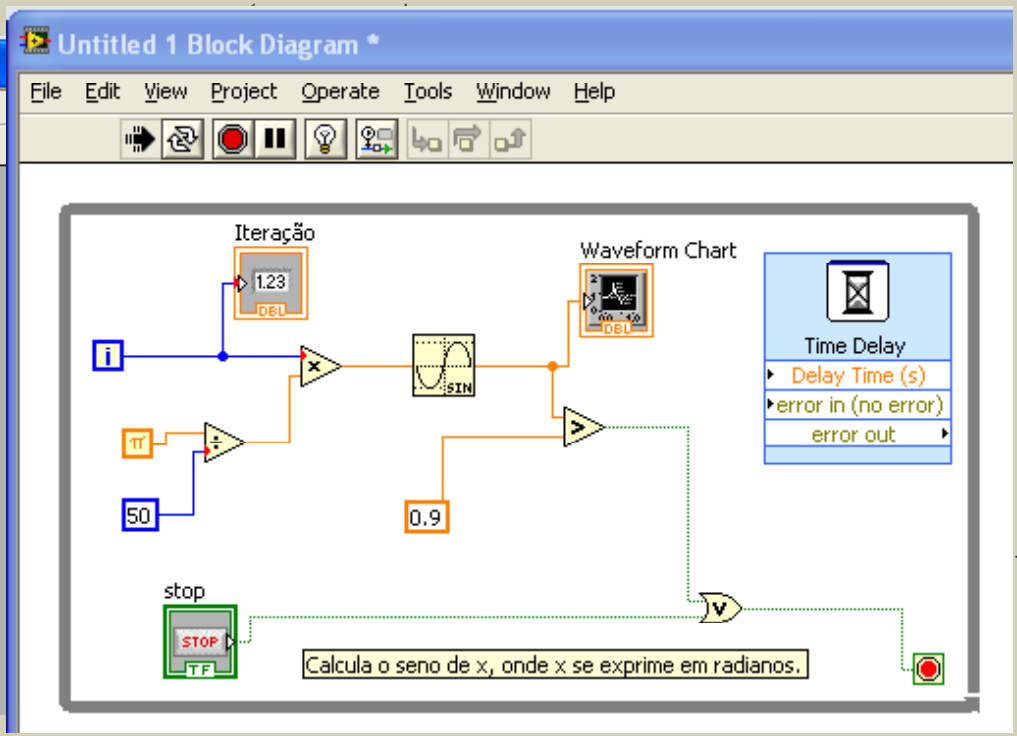
Construa e execute o seguinte VI:



6.4 Ciclos **WHILE**

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

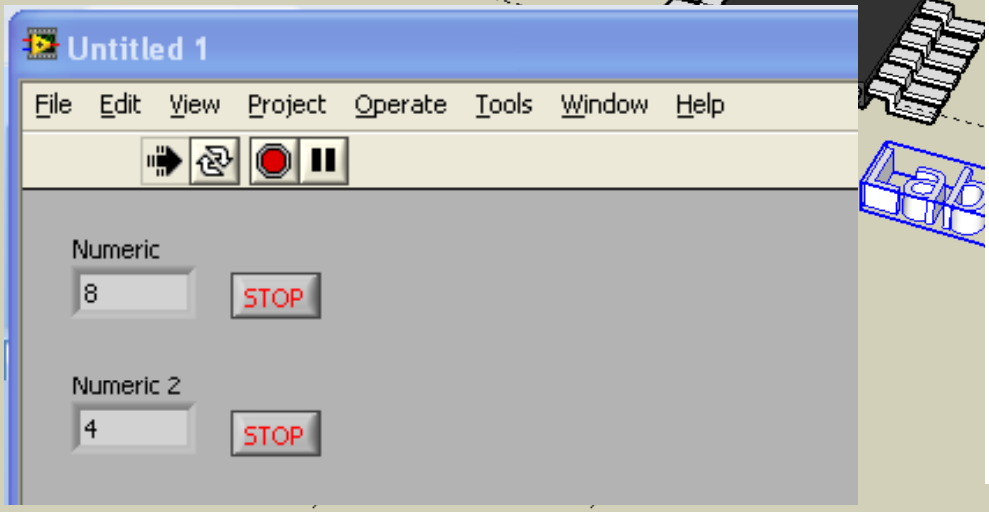


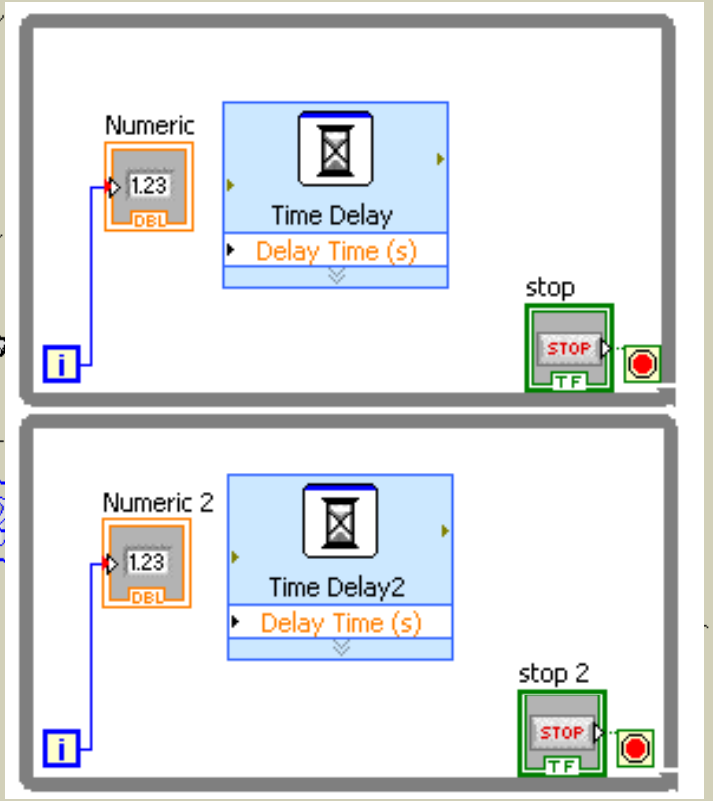


6.4 Ciclos **WHILE**

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).

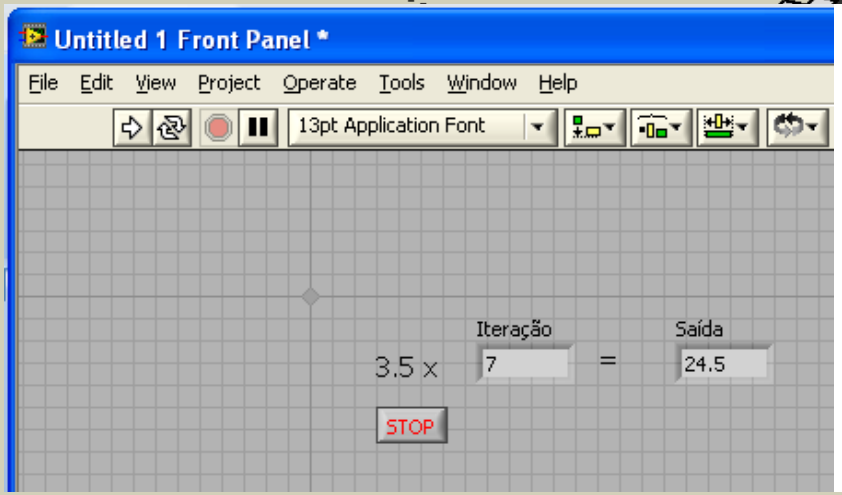


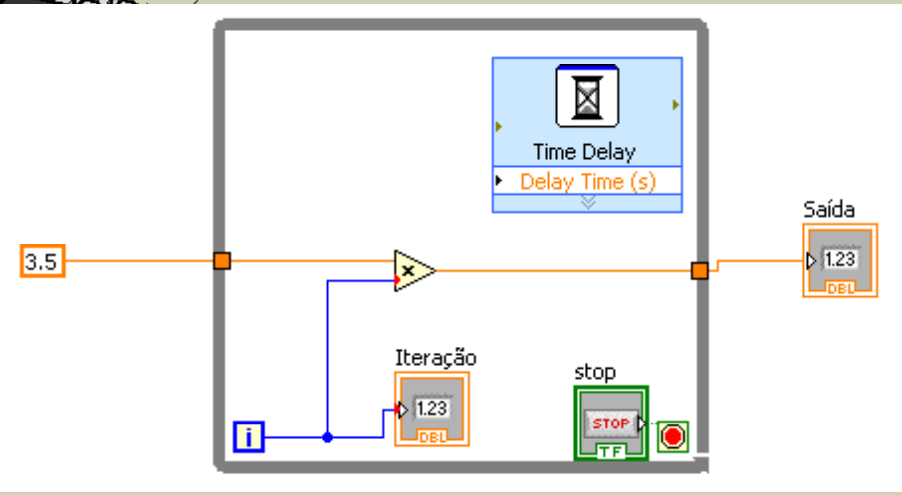


6.4 Ciclos **WHILE**

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

EXEMPLO: Construa e simule o seguinte VI...





6.4 Exercícios

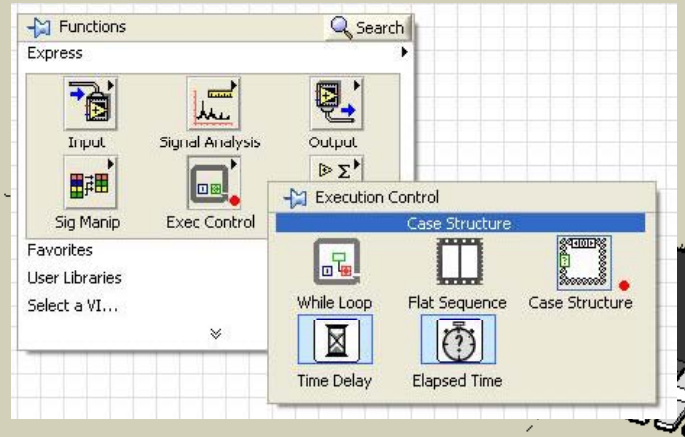
***EX 21**: Construa um ciclo que se mantenha em execução até que o valor originado, por um gerador de números aleatórios inteiros entre 0 e 100, seja igual a 50. Coloque um atraso de 100 ms dentro do ciclo e apresente, no **front panel**, o número de iterações produzidas.*

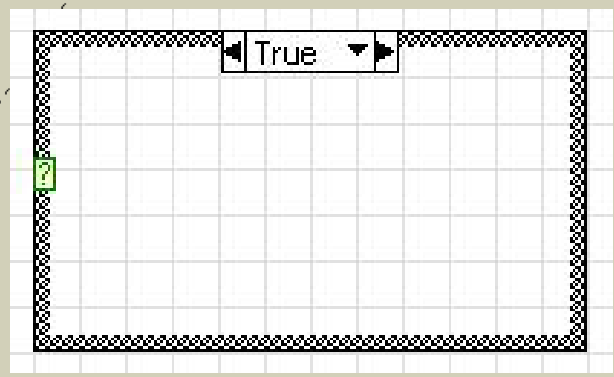
***EX 22**: Apresente um VI capaz de fazer um LED piscar com uma frequência igual a 1Hz:*
a) Com recurso a shift-registers
b) Sem recurso a shift-registers (quando [i] for "par" o LED acende e quando [i] for ímpar o LED apaga)

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

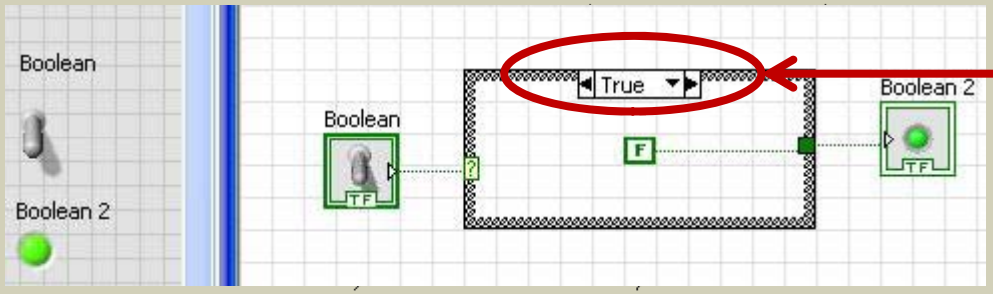
6.5 Estrutura **CASE**

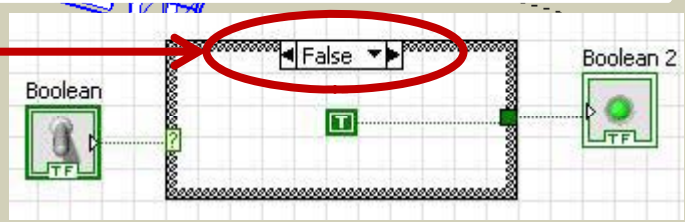
Estruturas do tipo CASE permitem que determinadas acções tomem lugar em função de possíveis valores de uma variável de decisão.





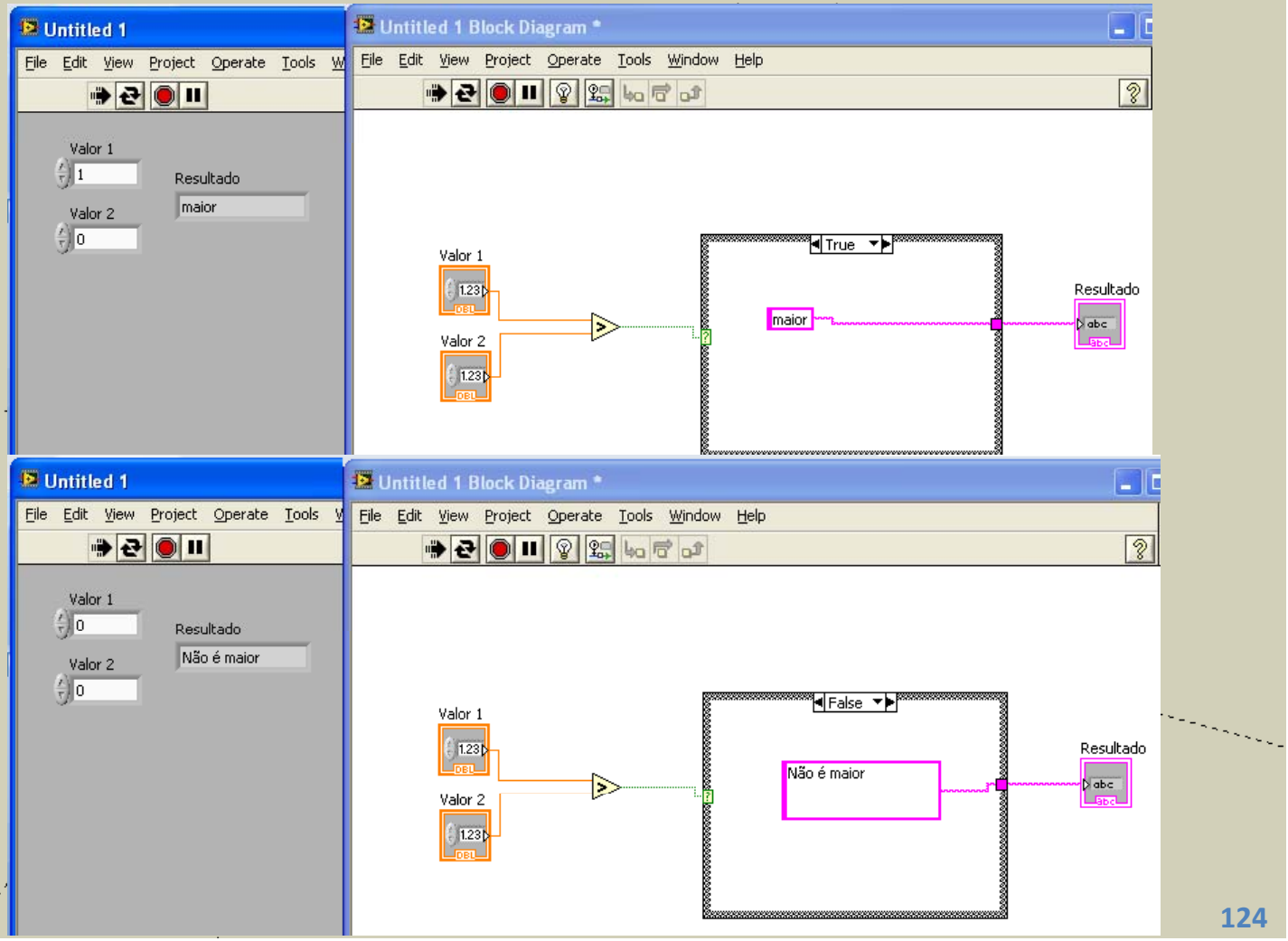
EXEMPLO: Construa e simule o seguinte VI...





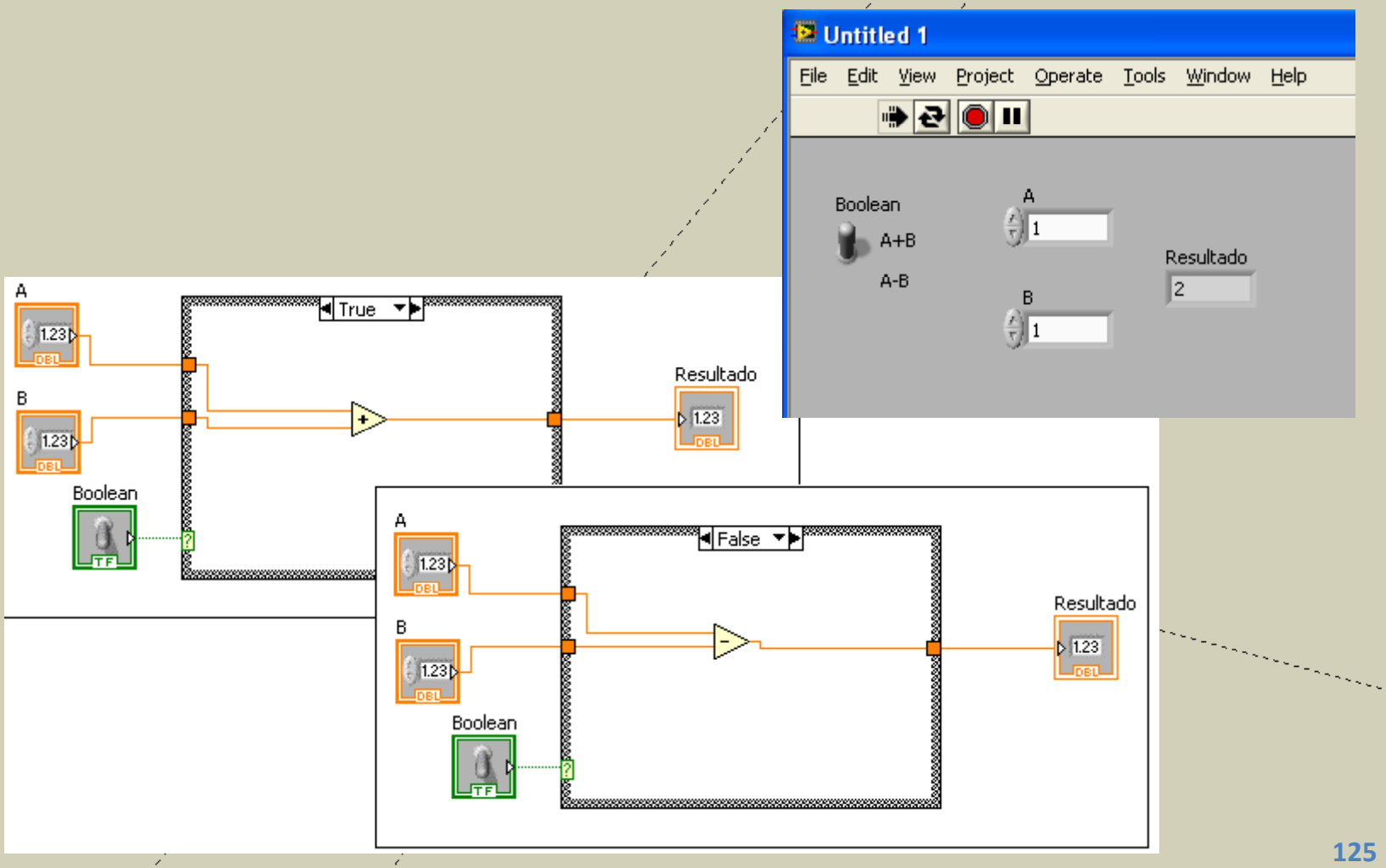
6.5 Estrutura **CASE**

- **CONDIÇÕES LÓGICAS**



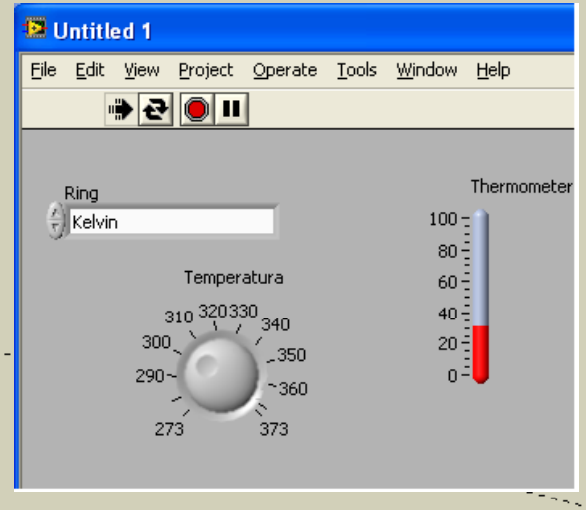
6.5 Estrutura **CASE**

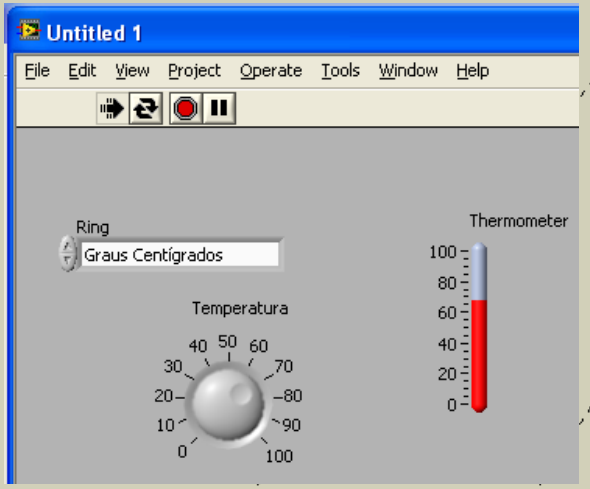
- CONDIÇÕES LÓGICAS

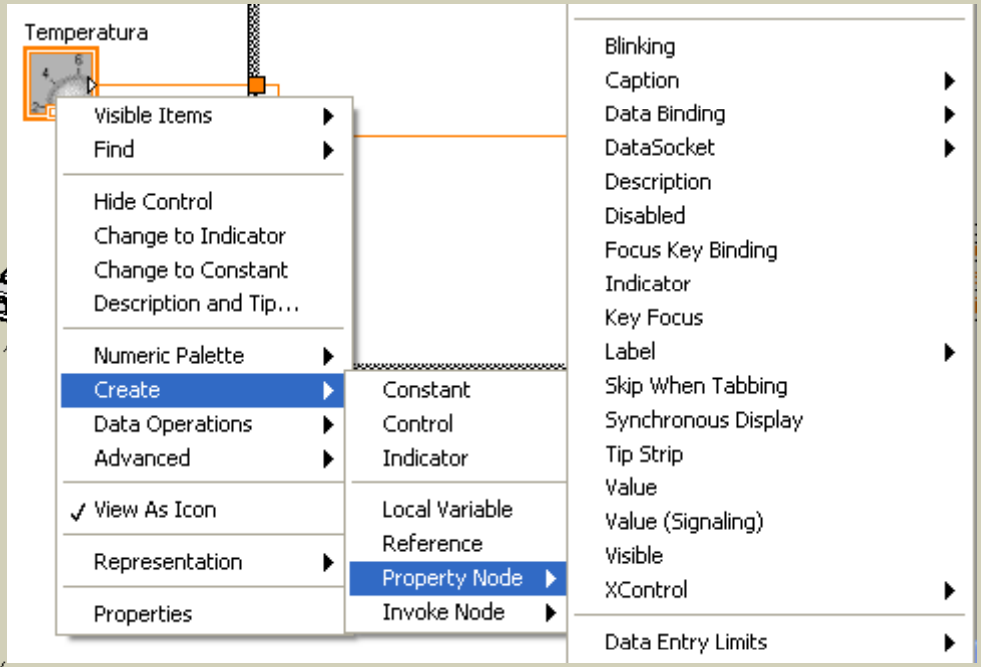


6.5 Estrutura **CASE**

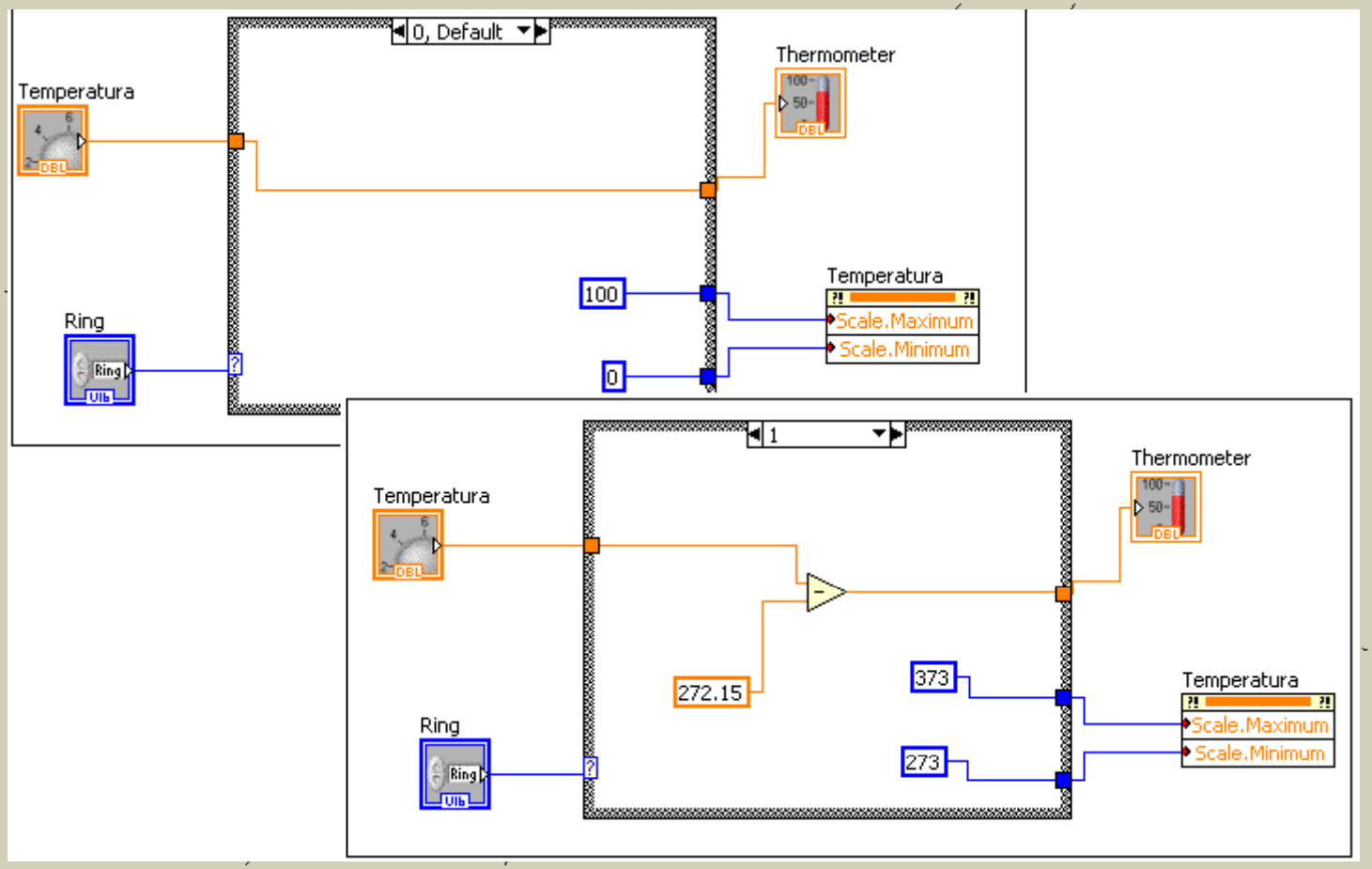
- CONDIÇÕES: INTEIROS







6.5 Estrutura **CASE**

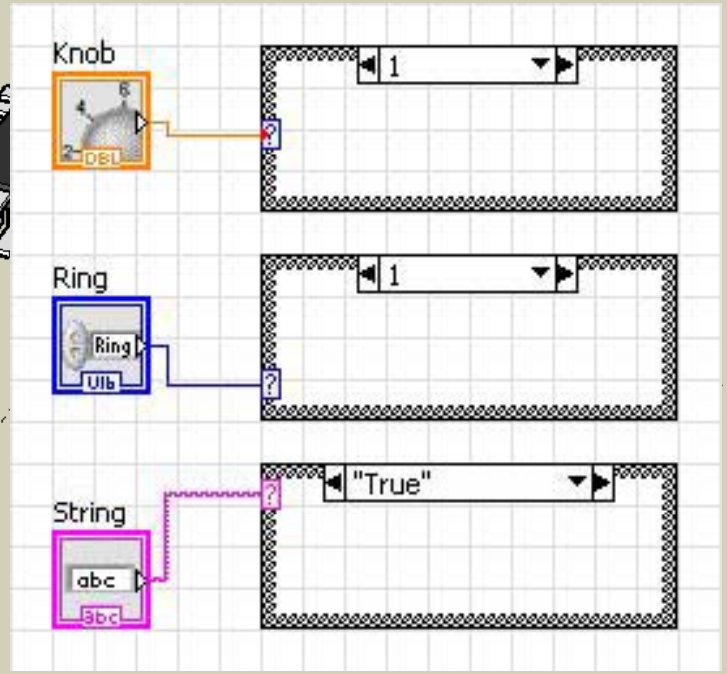


6.5 Estrutura **CASE**

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

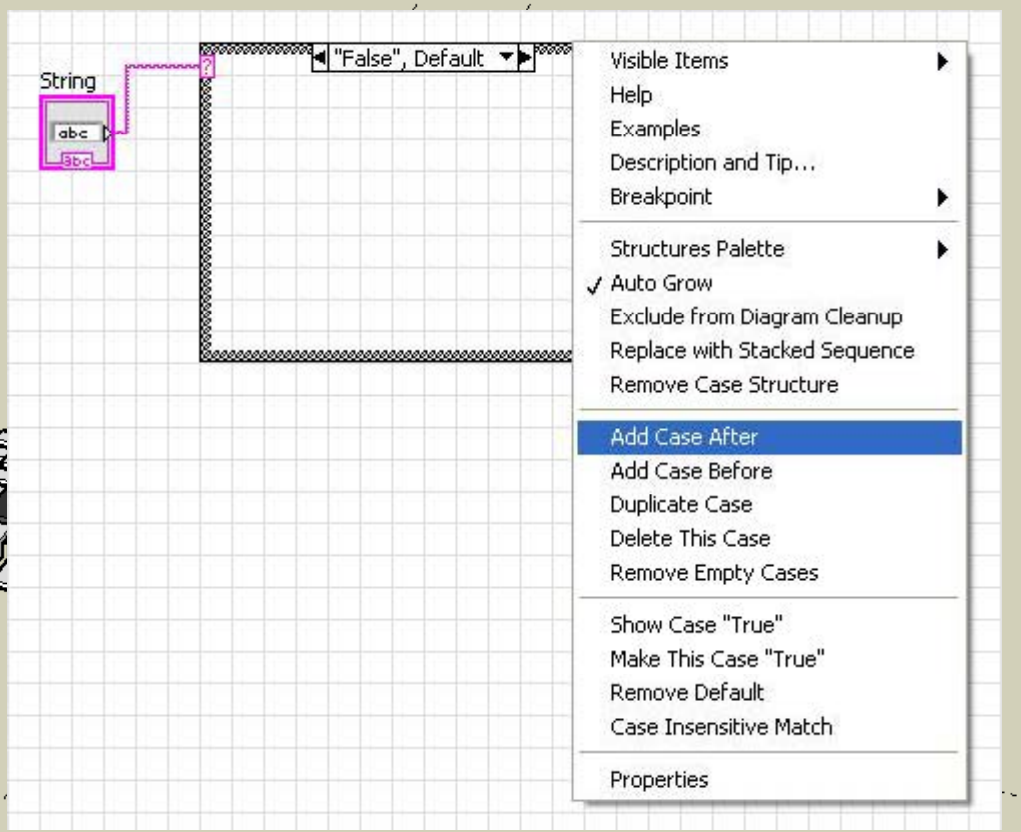
A paleta de opções pode ser facilmente estendida para situações não-booleanas.

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



6.5 Estrutura **CASE**

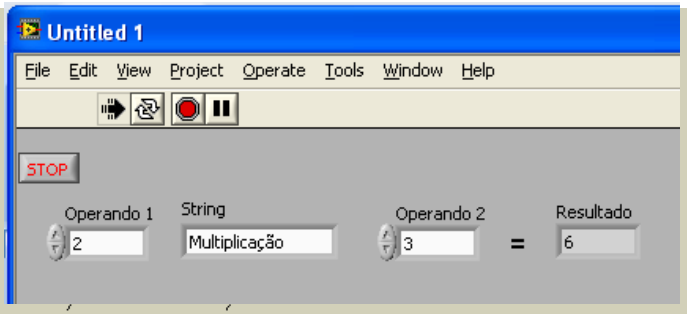
O número de situações a serem contempladas pela estrutura CASE podem ser aumentadas ou diminuídas recorrendo ao pop-up menu associado a esse objecto...

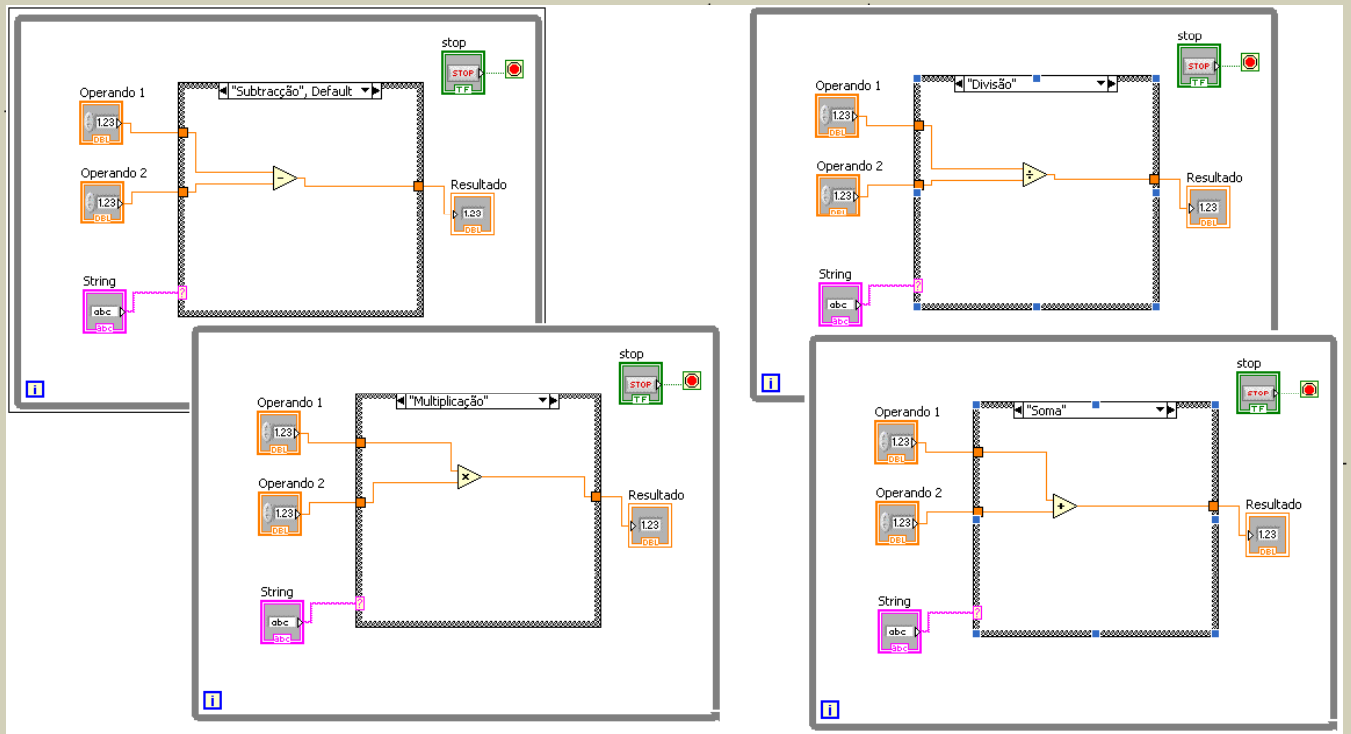


8.4 Estrutura **CASE**

- **CONDIÇÕES: STRINGS**

CASE dentro de WHILE





6.5 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.*

6.5 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.***

6.3 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.*

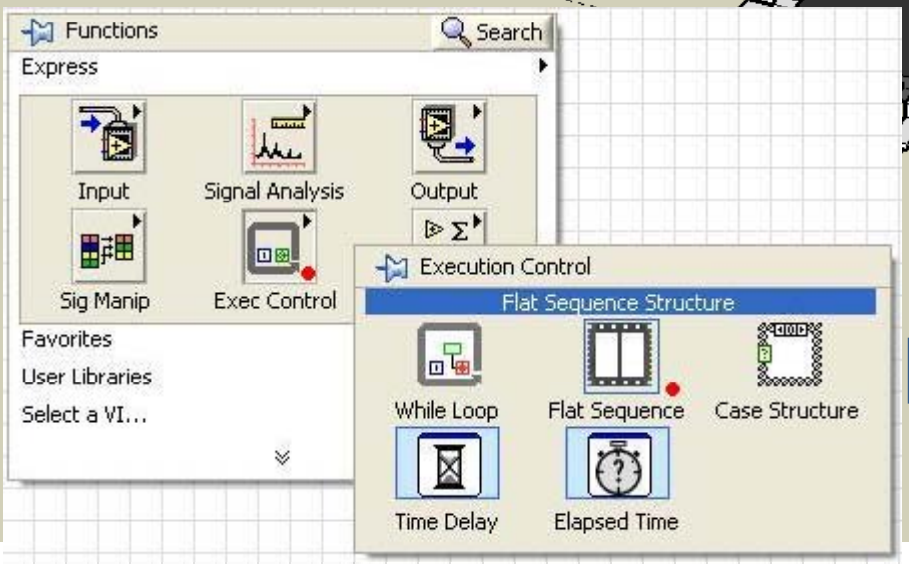
Caso o jogador ganhe, o valor da aposta é acrescentado ao saldo. Se o jogador perder o valor da aposta será retirado do saldo

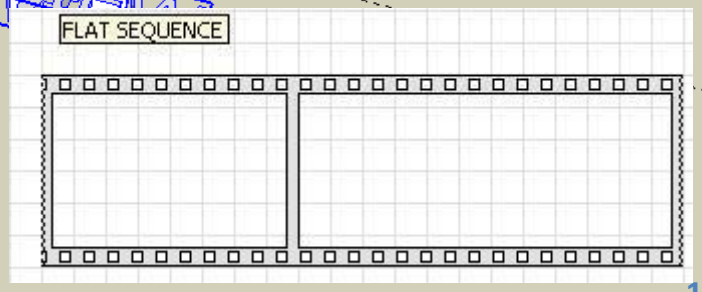
EX 26d: *O saldo deve ser iniciado com 50€. Para além disso o jogo deve terminar quando o saldo do jogador chegar a zero ou então desistir. Não devem ser permitidas apostas superiores ao saldo.*

7.0 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

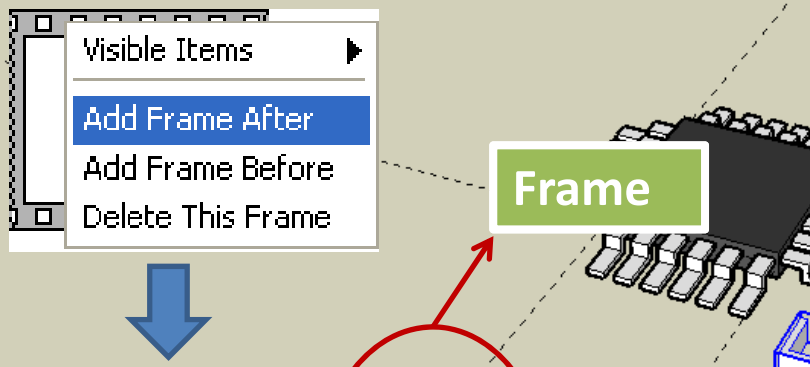


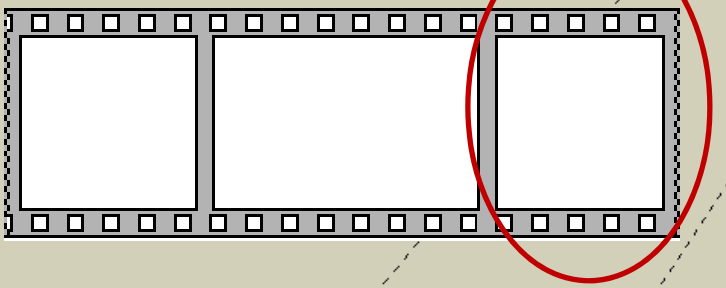


7.1 Sequências Planas

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.





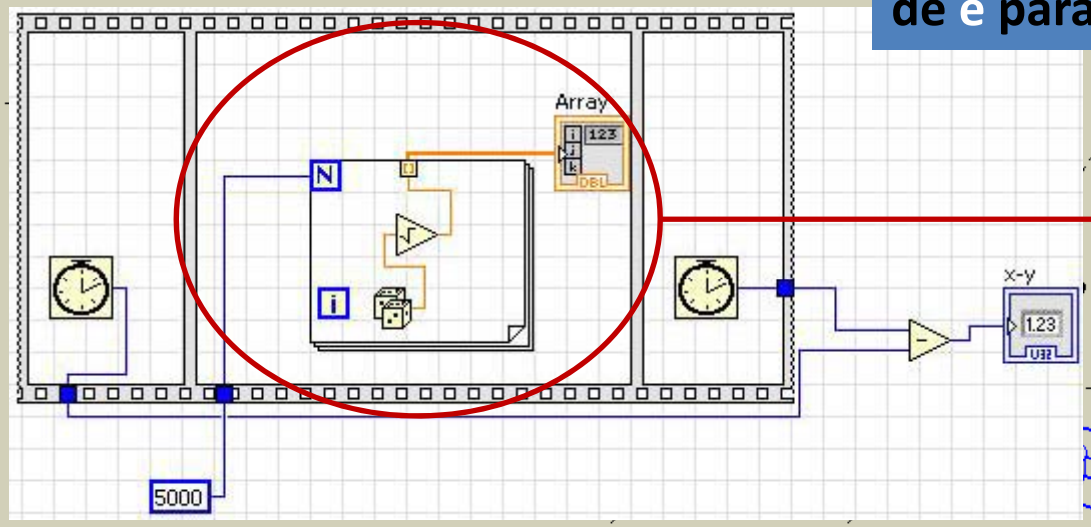
Frame

MAU! DA ESQUERDA PARA A DIREITA É QUE NÃO PODE SER...

7.1 Sequências Planas

EXEMPLO: Construa e simule o seguinte VI...

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

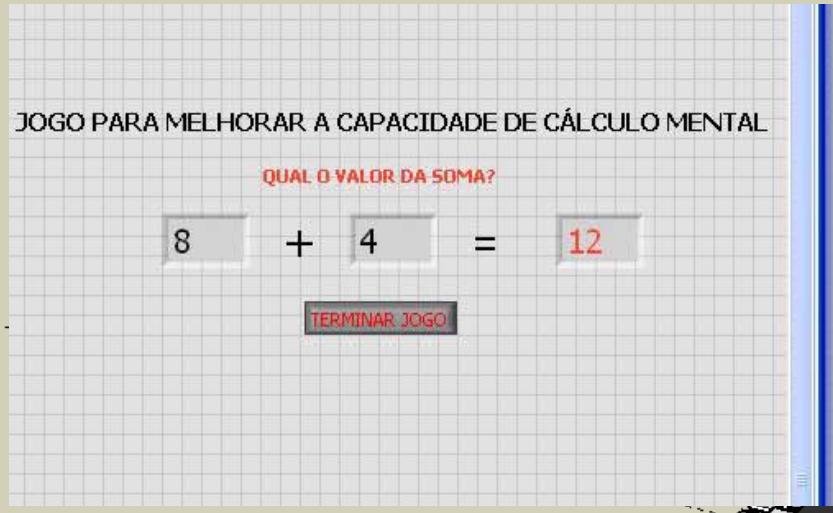


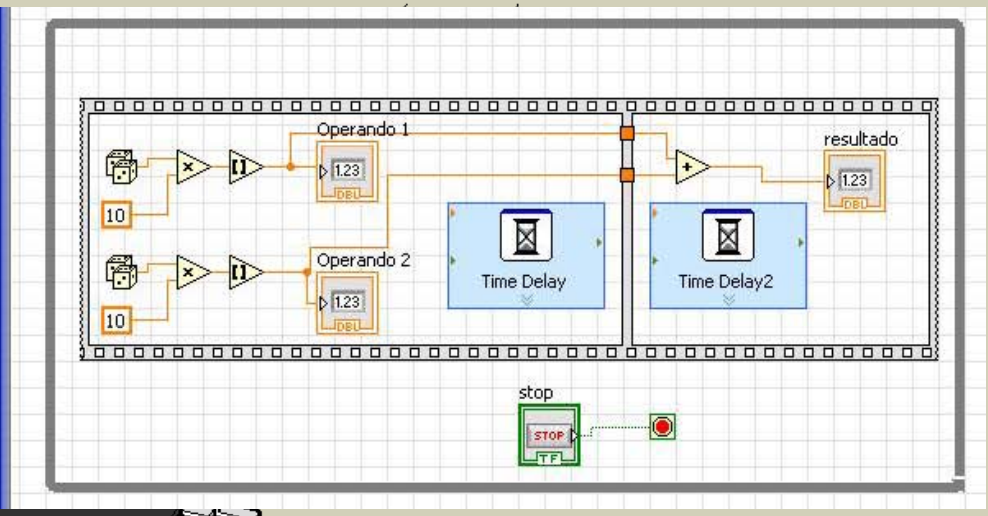
Subdiagrama

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.

7.1 Sequências Planas



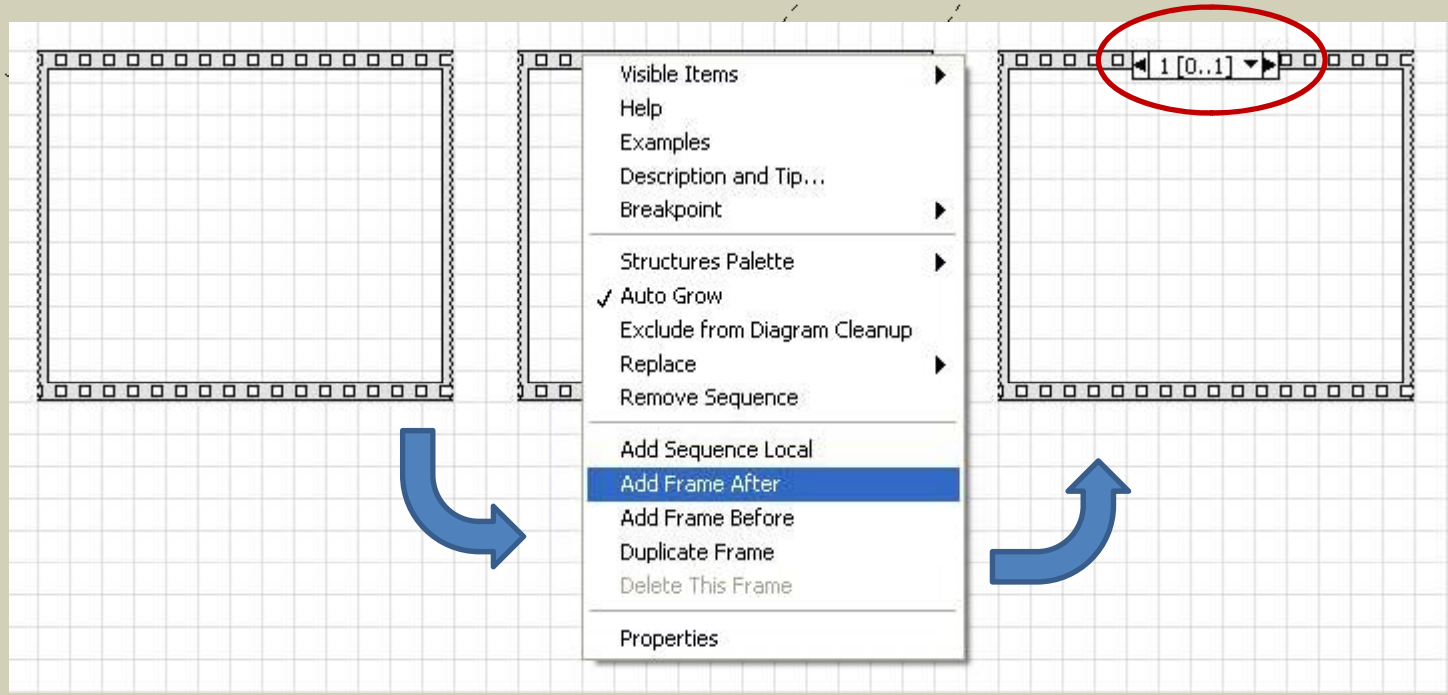


ESTE PROGRAMA É CAPAZ DE ME DAR JEITO... ANDO-ME A ENGANAR COM AS CONTAS DA REFORMITA!

7.2 Sequências Sobrepostas

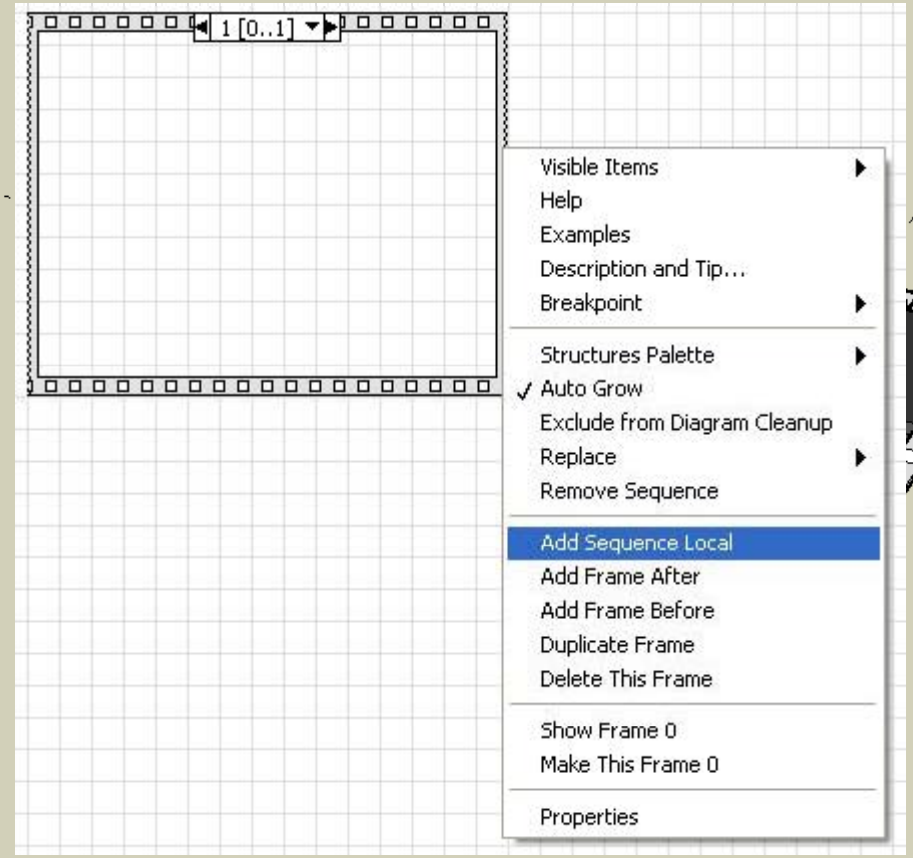
EX 27: ***Altere o "Jogo dos Números" de modo a que o utilizador tenha a possibilidade de escolher uma das quatro operações elementares.***

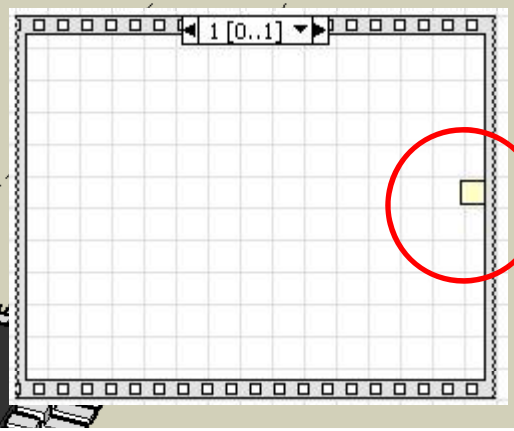
Referem-se a sequências cujas frames se encontram sobrepostas



7.2 Sequências Sobrepostas

Passagem de parâmetros entre frames: variáveis locais

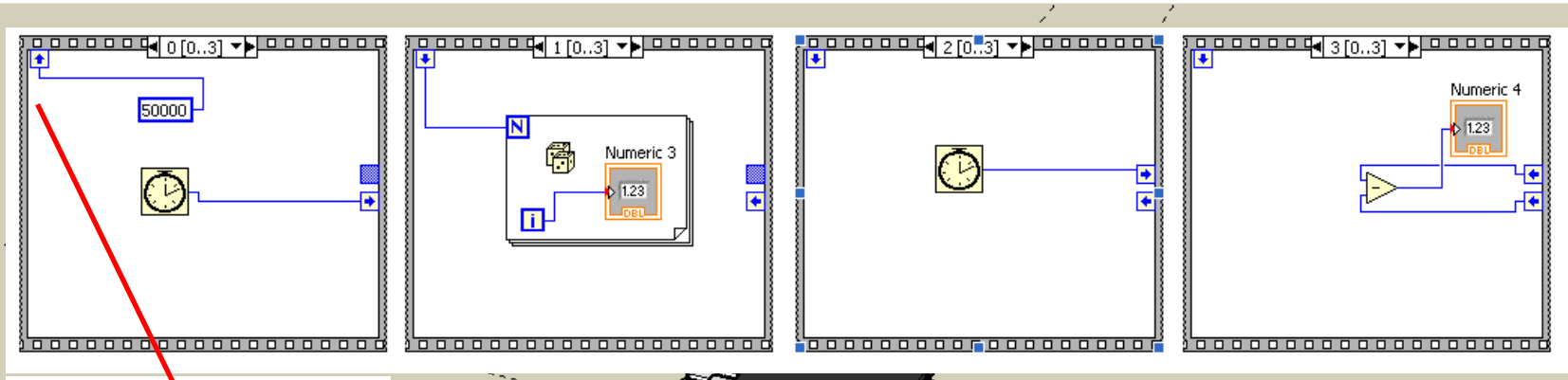


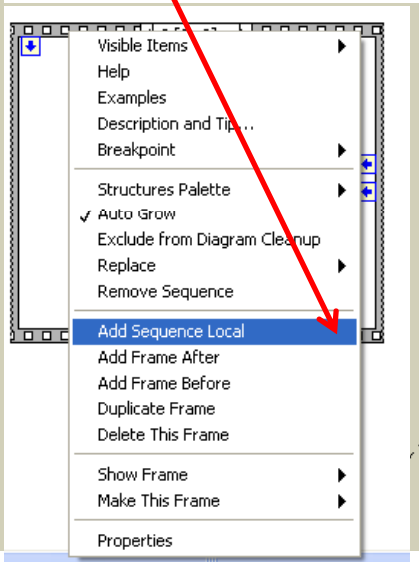


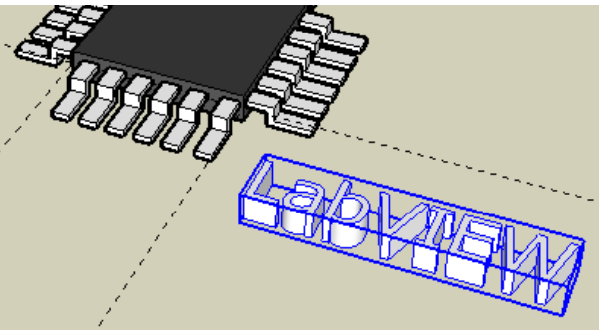
O valor de uma variável local criada num frame ficará disponível para todas as frames criadas apenas subsequentemente.

7.2 Sequências Sobrepostas

EXEMPLO: Construa e simule o seguinte VI...

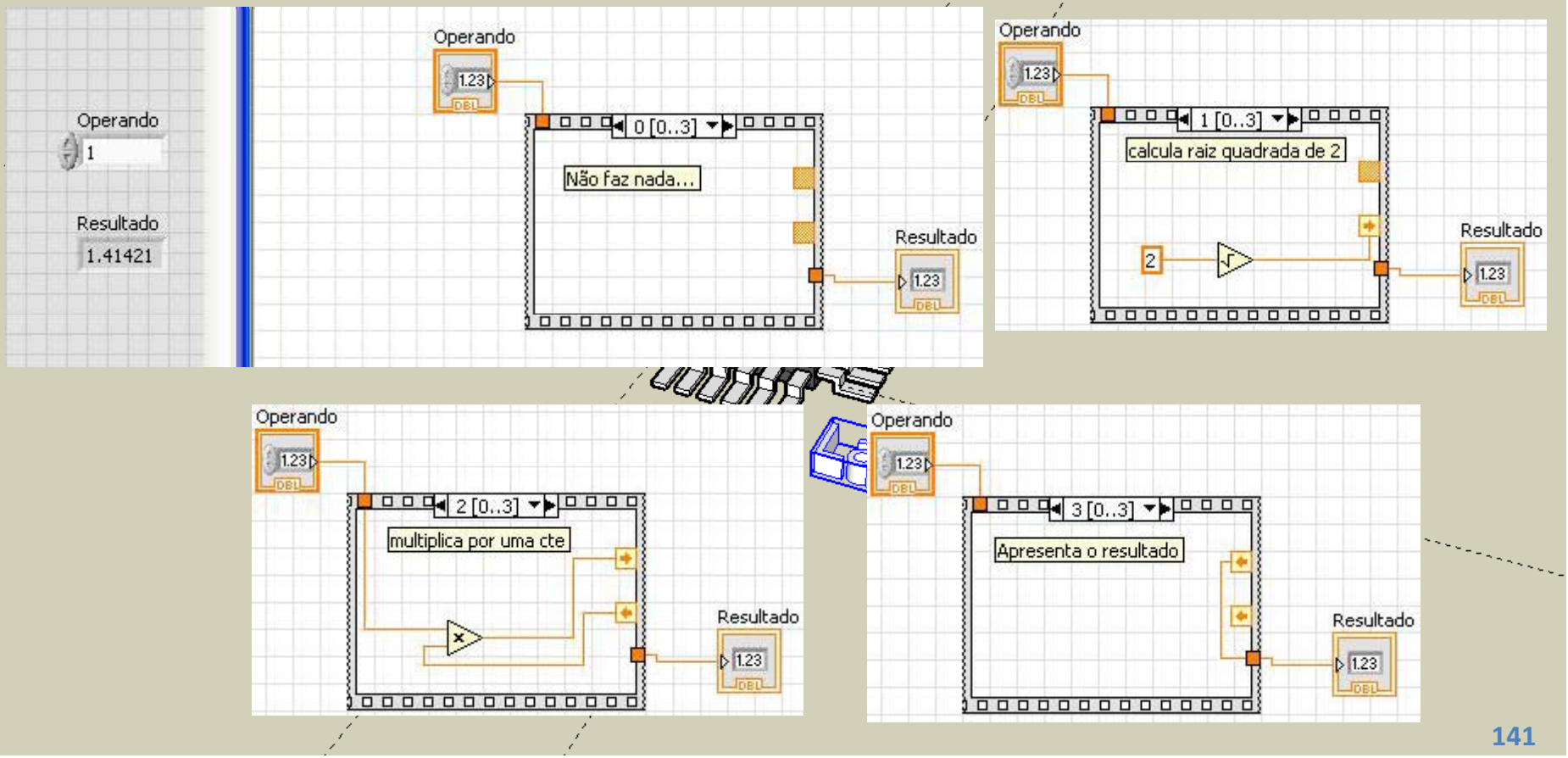






7.2 Sequências Sobrepostas

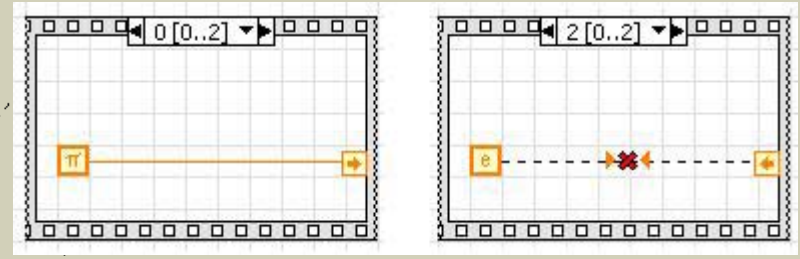
EXEMPLO: Construa e simule o seguinte VI...



7.3 Sequências: algumas considerações

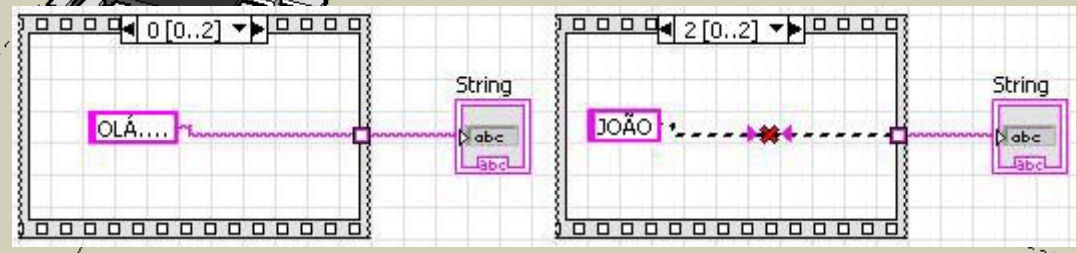
Uma **variável local** só pode ser instanciada UMA vez

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.

7.4 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.*

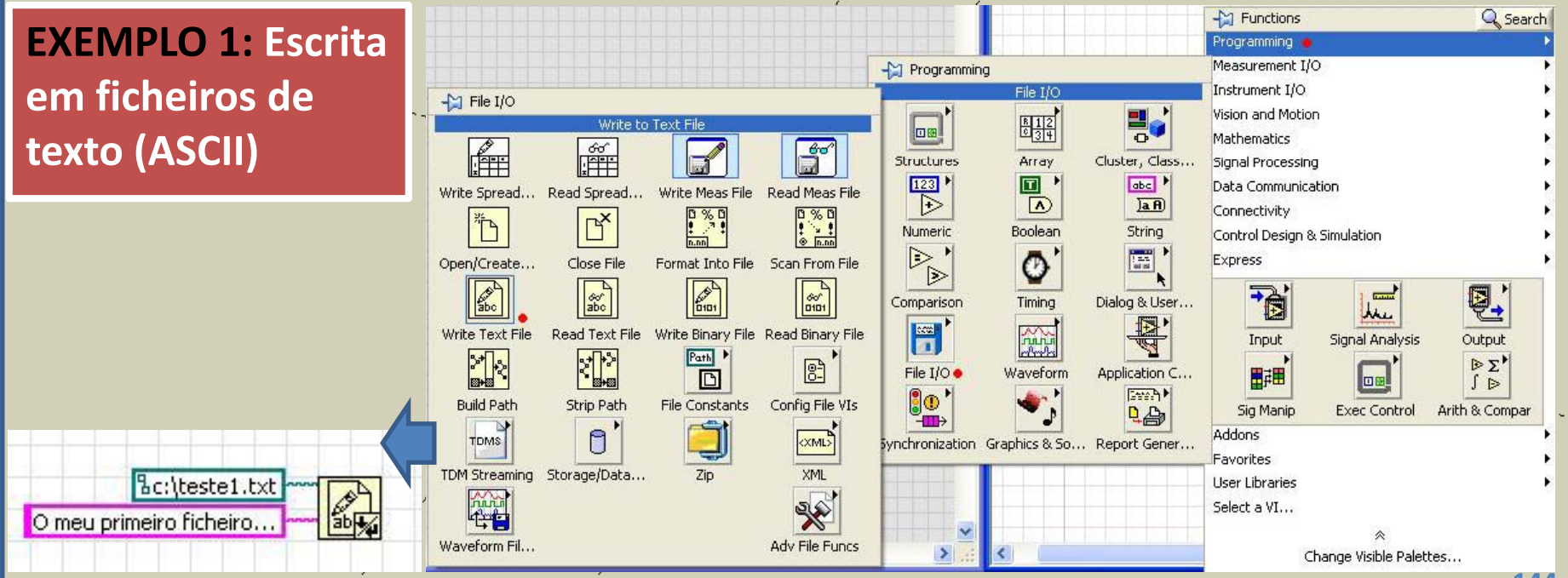
EX 30: *Um ciclo FOR é executado 10 vezes. Em cada iteração é gerado um número aleatório inteiro entre 0 e 100. Quando este ciclo terminar, um segundo ciclo FOR é executado onde N será o maior dos valores aleatórios gerados no ciclo anterior.*

8.0 I/O em ficheiros

Com frequência existe a necessidade de transferir informação do LabVIEW para um ou mais arquivos em disco.

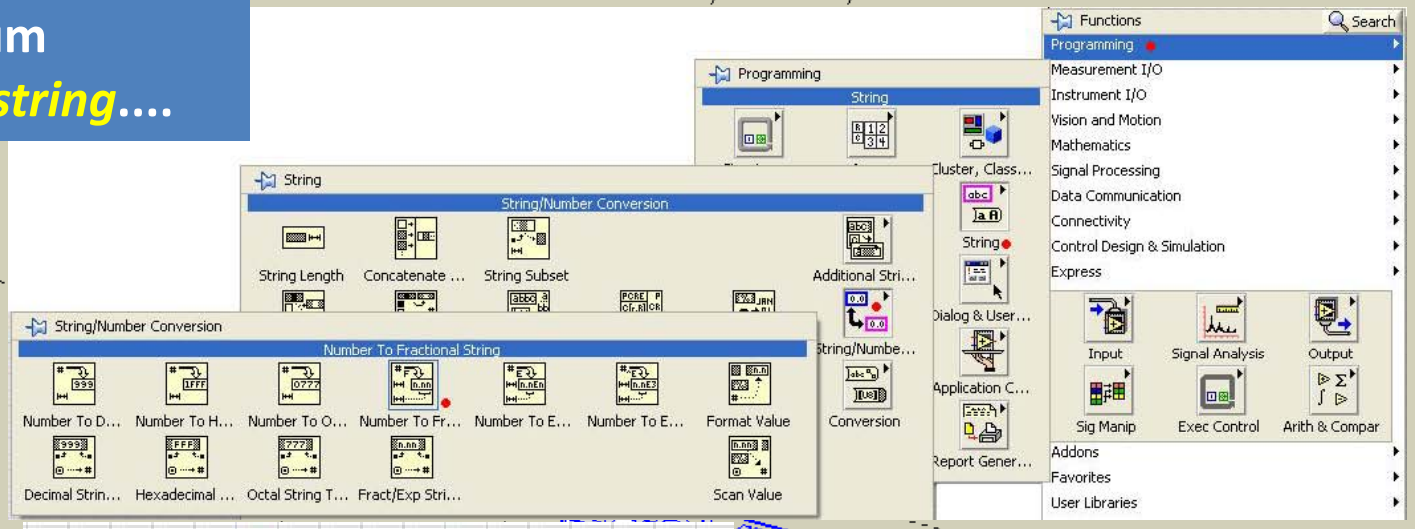
Existem diversos formatos de ficheiros: ASCII e Binários.

EXEMPLO 1: Escrita em ficheiros de texto (ASCII)

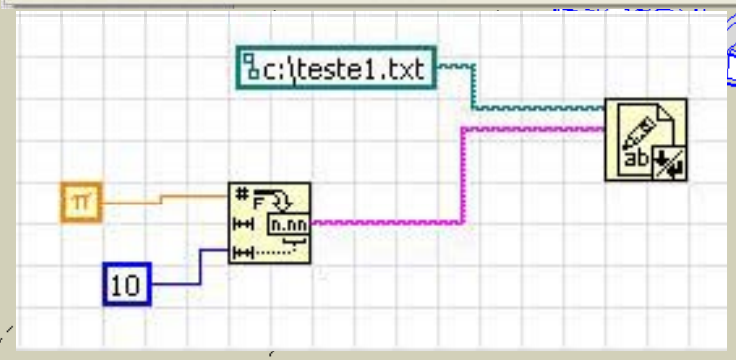


EXEMPLO 2: E se for um valor numérico?

Conversão de um número numa *string*....

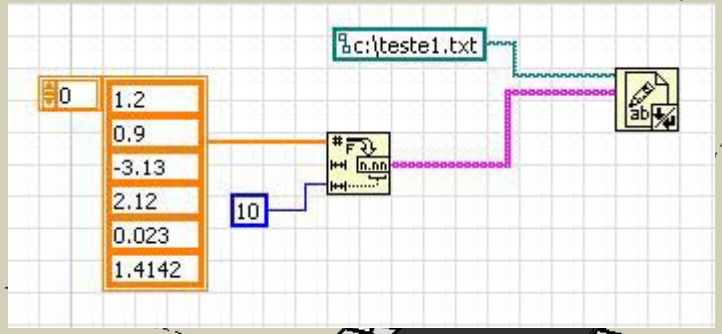


Testar....



EXEMPLO 3: E se for um array?

Testar....



EX 31: *Criar um ficheiro com 100 amostras da função:*

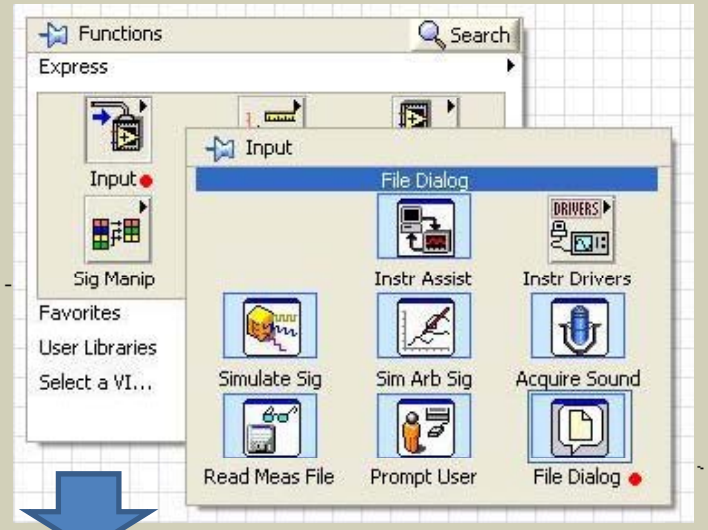
$$y = \sin(\pi * k/100)$$

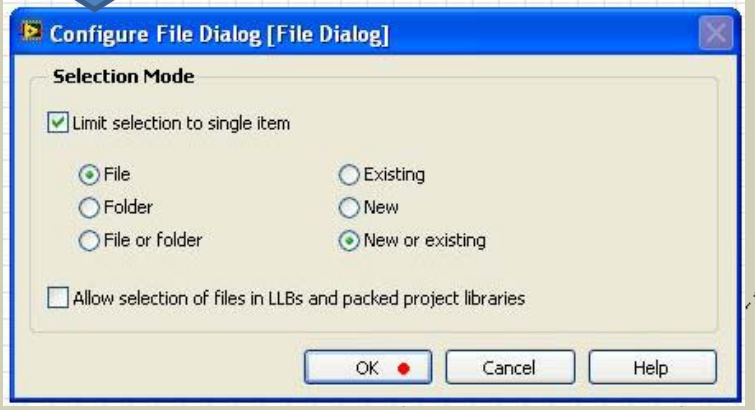
para k=0,...,99

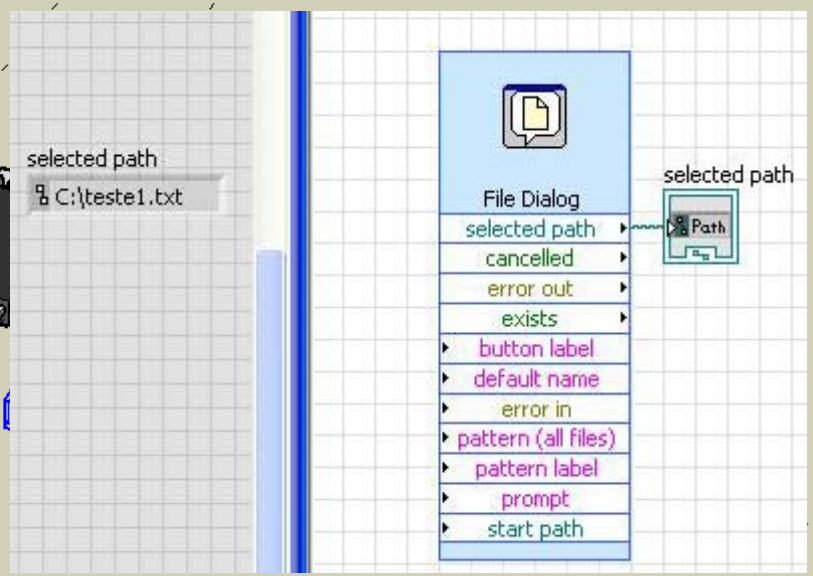
Posteriormente utilize o Microsoft Excel para visualizar, num gráfico, os valores existentes no ficheiro.

9.0 Janelas de Diálogo

EXEMPLO: Construa e simule o seguinte VI

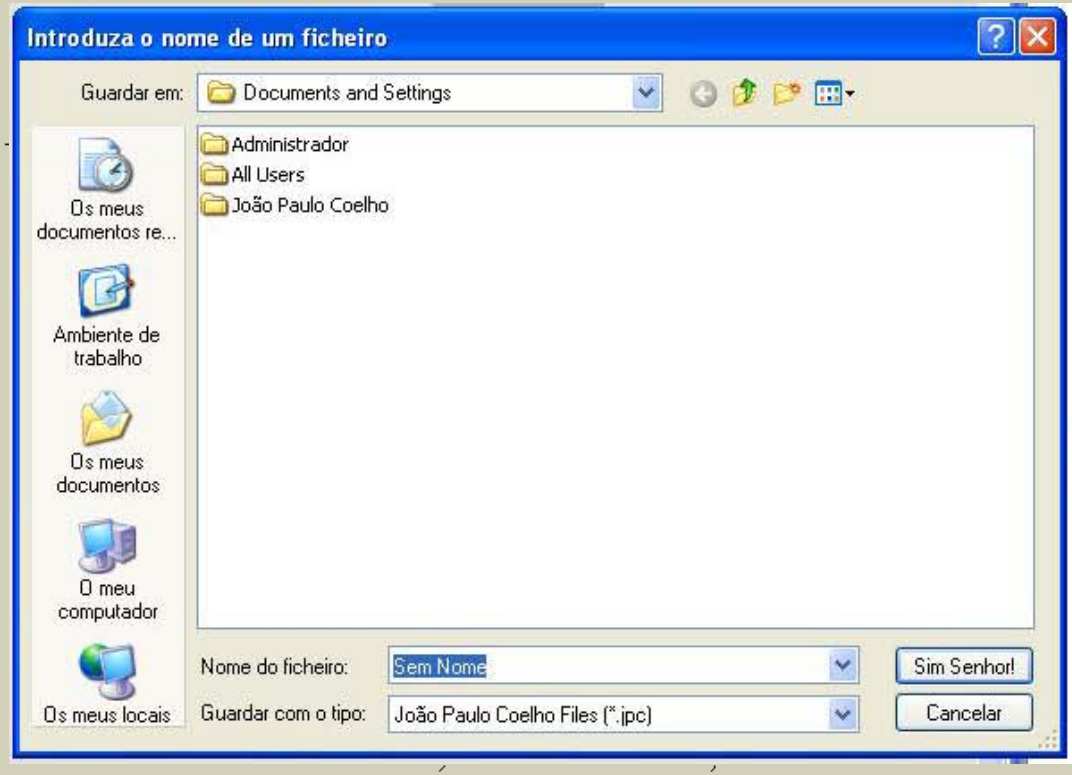


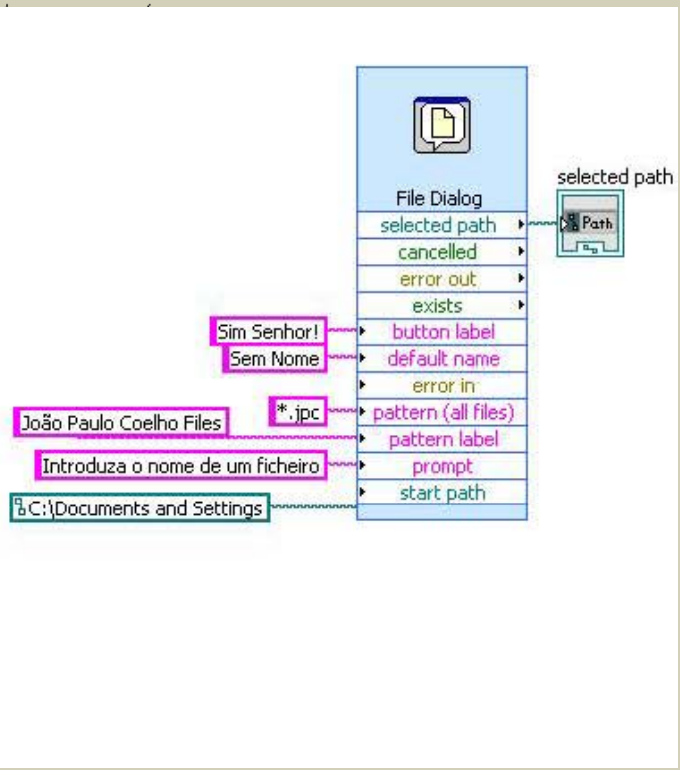




9.0 Janelas de Diálogo

EXEMPLO: Personalização da caixa de diálogo...



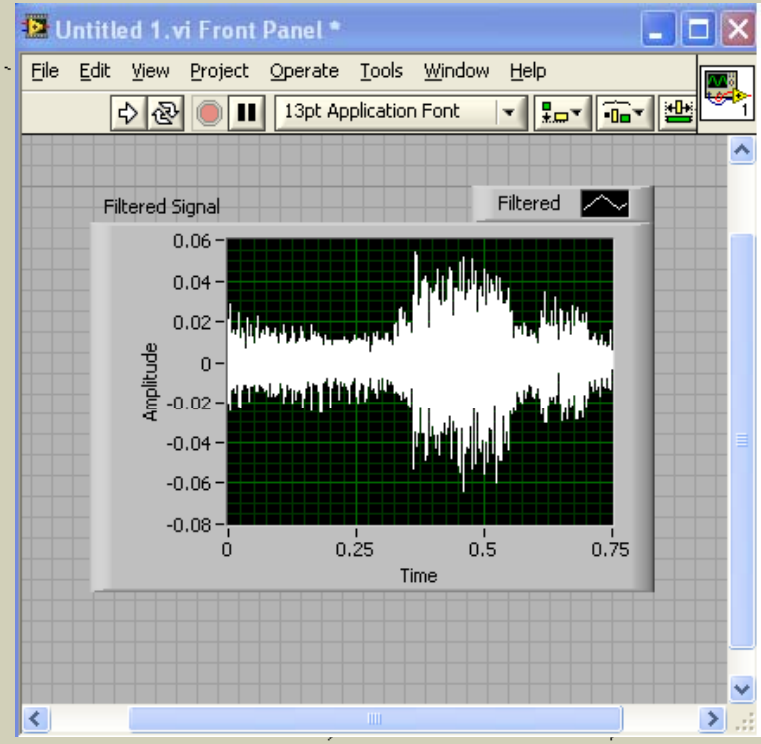


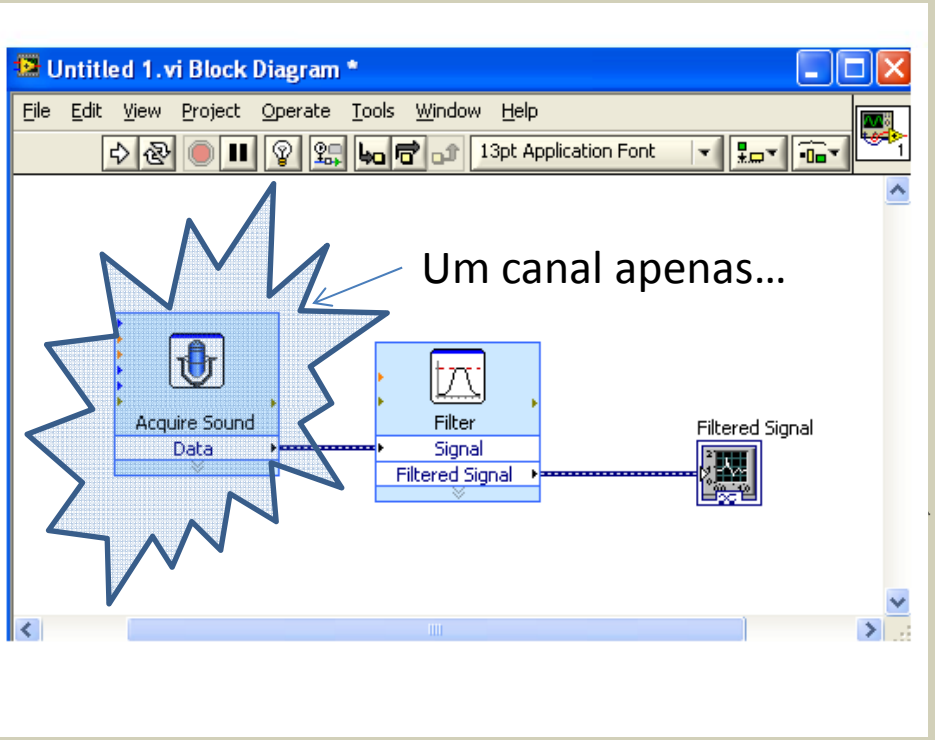
10.0 Aquisição de Dados

- Aquisição de Dados:
  - Placa de Som
  - DAQ
- "Data Logging":
  - Escrita de dados em ficheiro
  - Leitura de dados de ficheiros

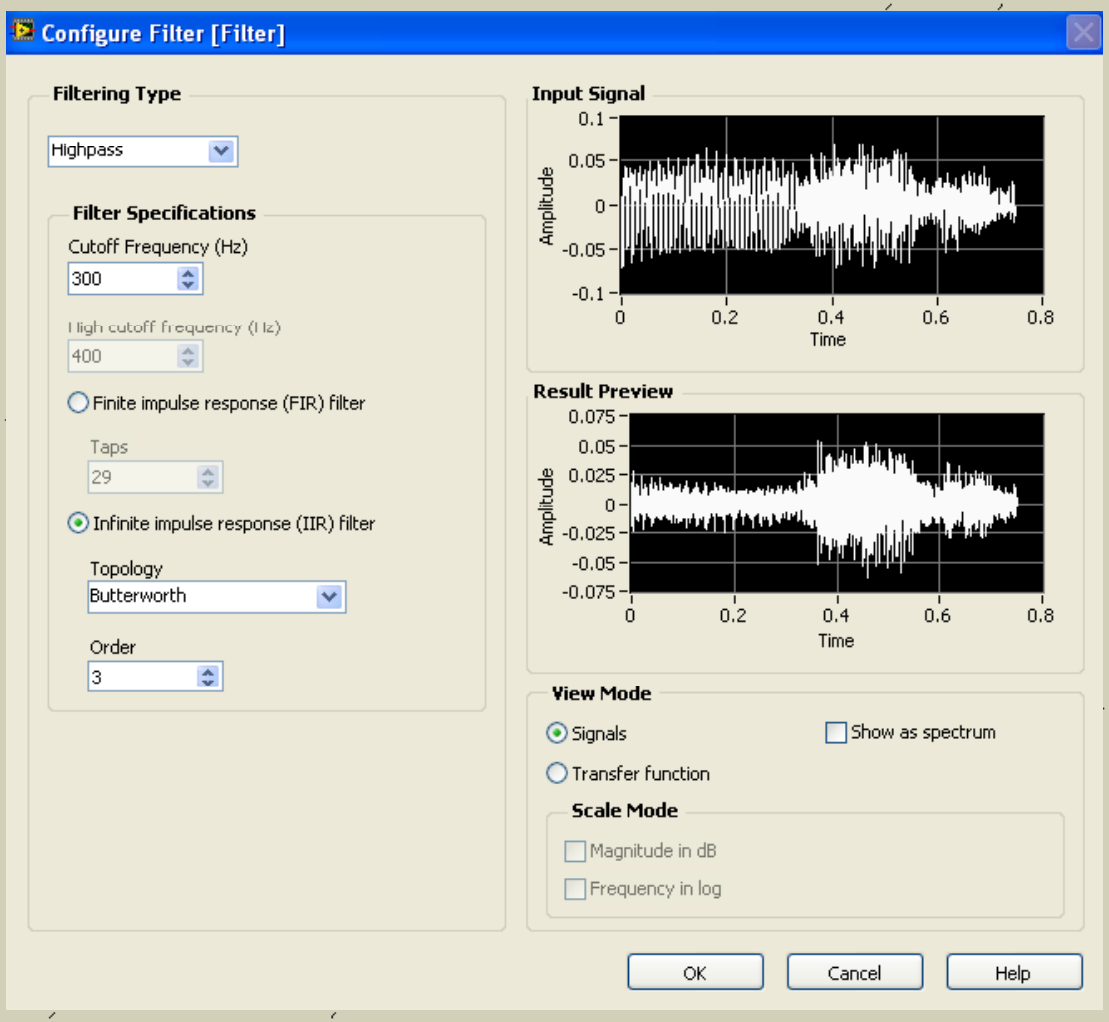
10.1 Placa de Som

- Aquisição e apresentação de 1 segundo de som obtido a partir do microfone.



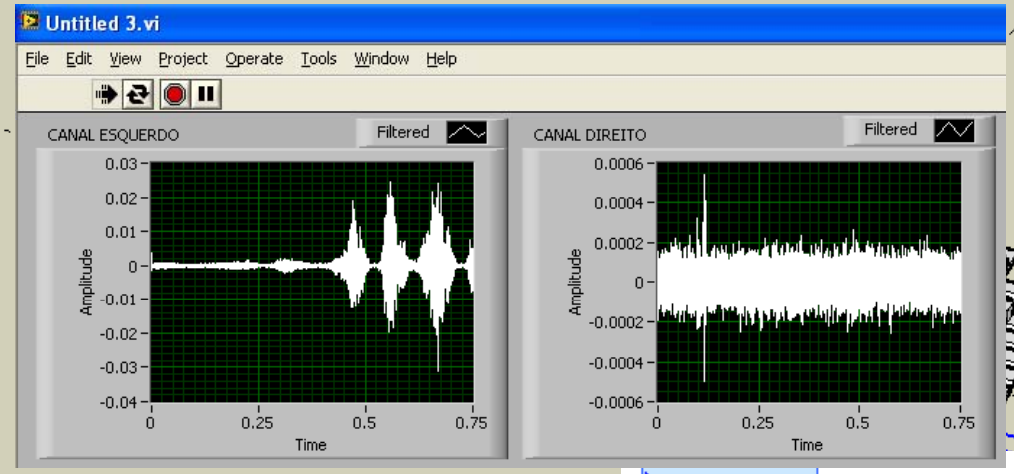


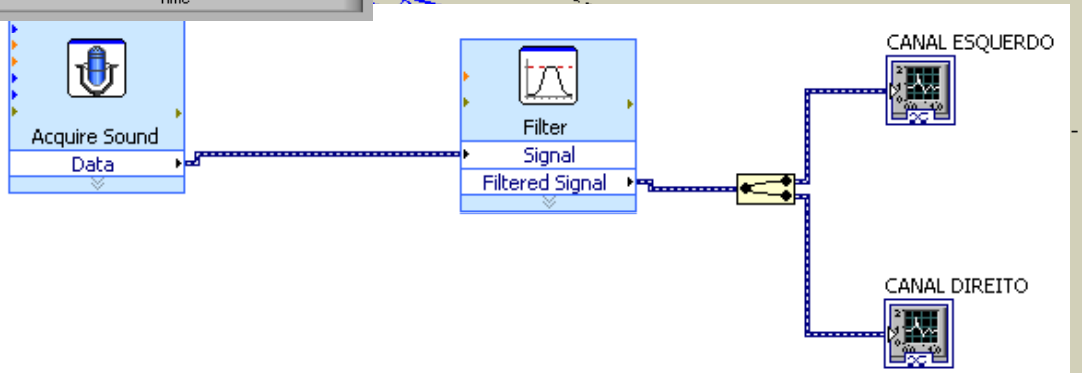
10.1 Placa de Som



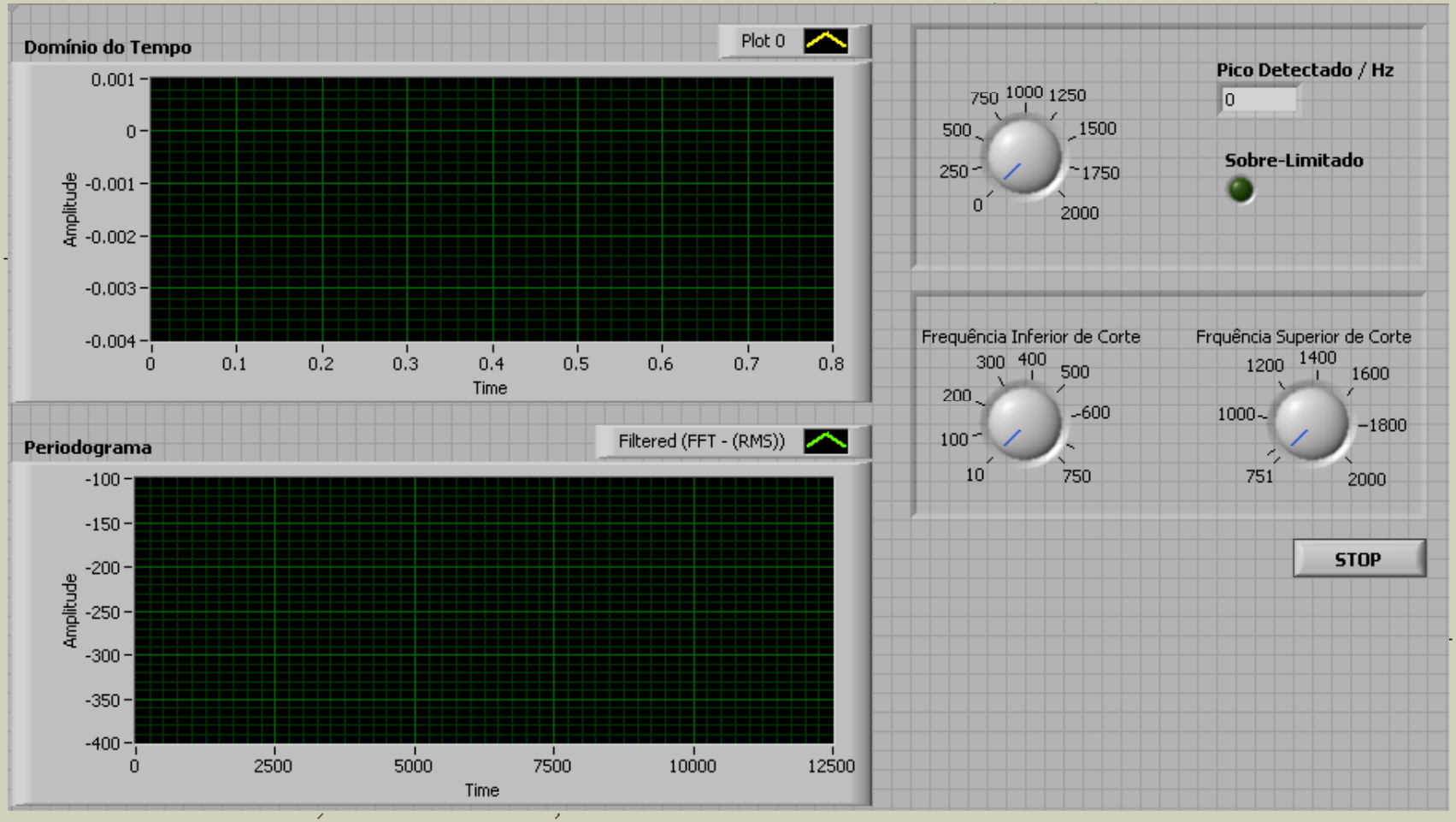
10.1 Placa de Som

- Repetir agora para STEREO.
  - Cada canal gráfico diferente!

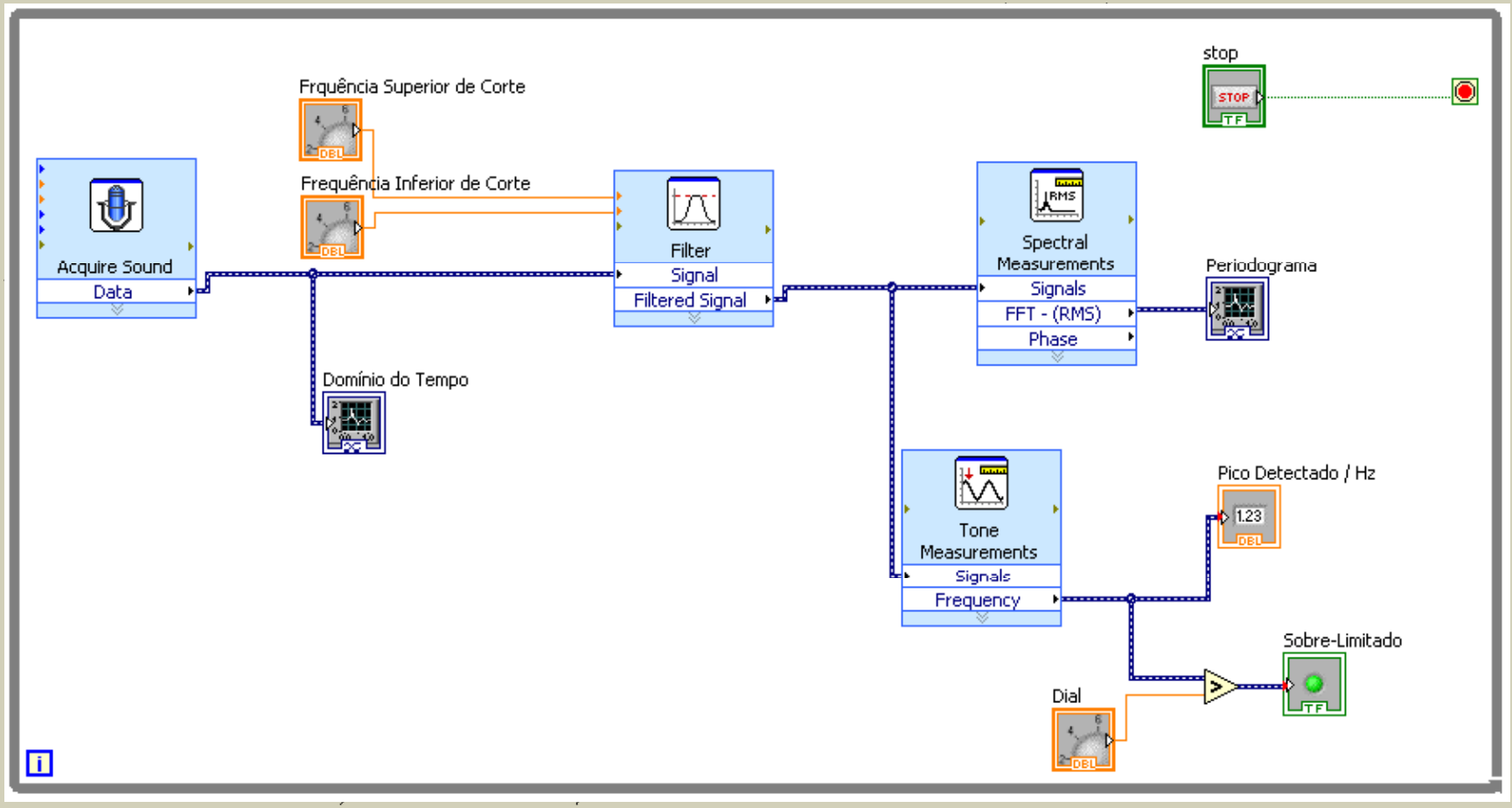




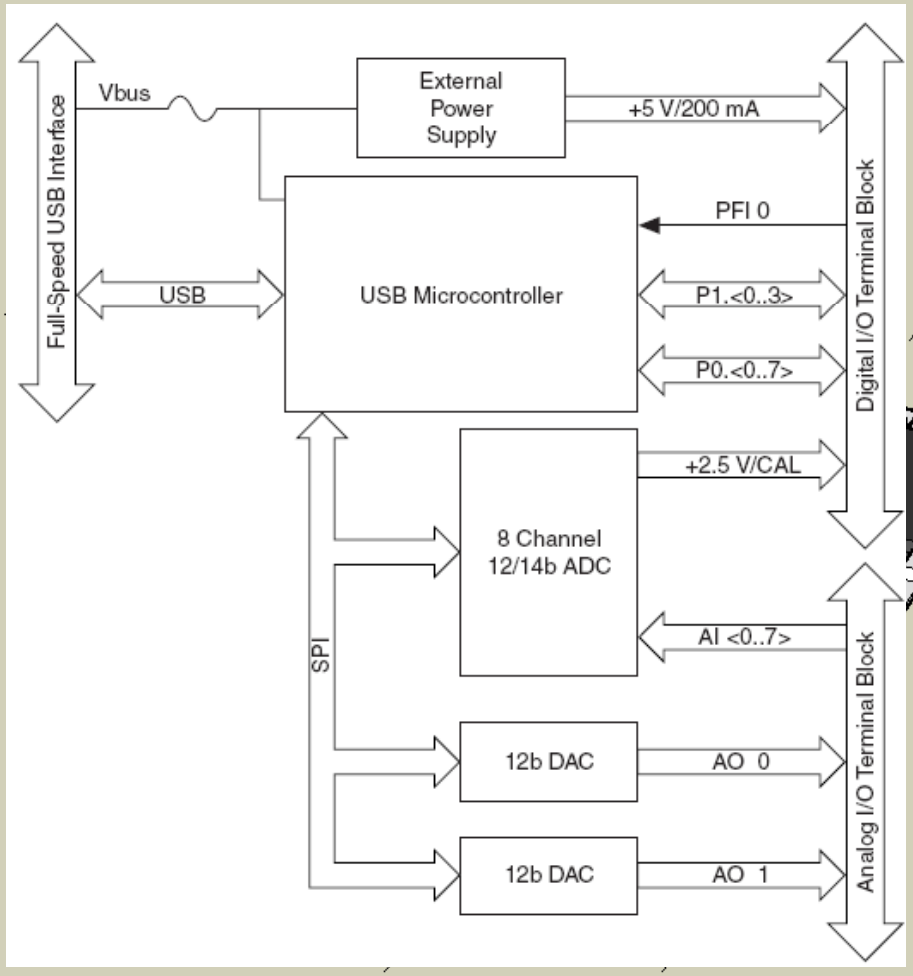
10.1 Placa de Som



10.1 Placa de Som



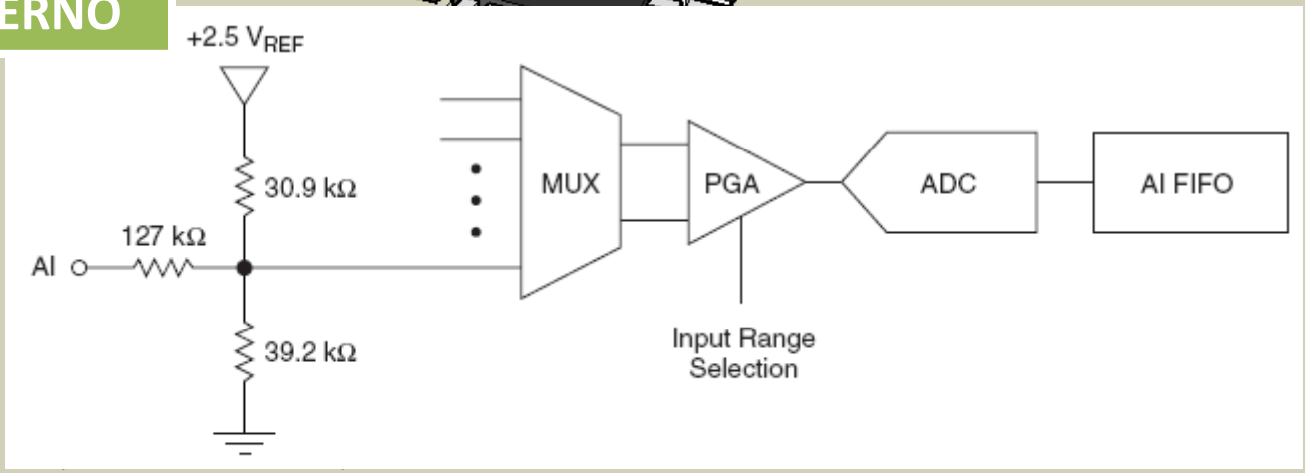
10.2 DAQ – NI USB 6008



10.3 Entradas Analogicas

Tipo de Conversor ..... APROXIMAÇÕES SUCESSICVAS
Entradas Analógicas .....8 SINGLE-ENDED, 4 DIFFERENTIAL,SOFTWARE SELECTABLE
Resolução de Entrada.....12 BITS DIFFERENTIAL, 11 BITS SINGLE-ENDED
Máxima razão de amostragem .....10 KS/S
Tensão de referência (SINGLE-ENDED).....±10 V
Tensão de referência (DIFFERENTIAL) .....±20 V, ±10 V, ±5 V, ±4 V,±2.5 V, ±2 V, ±1.25 V, ±1 V
Tensão de trabalho .....±10 V
Impedância de entrada.....144 KΩ
Protecção de sobretensão .....±35V

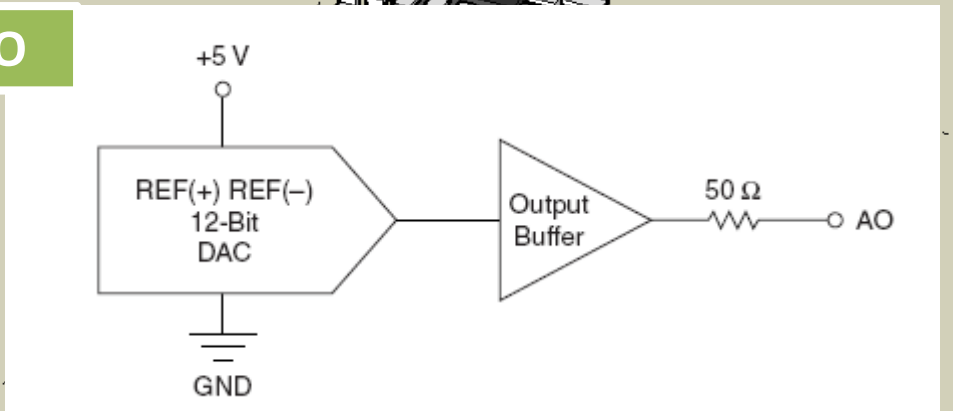
MODELO INTERNO



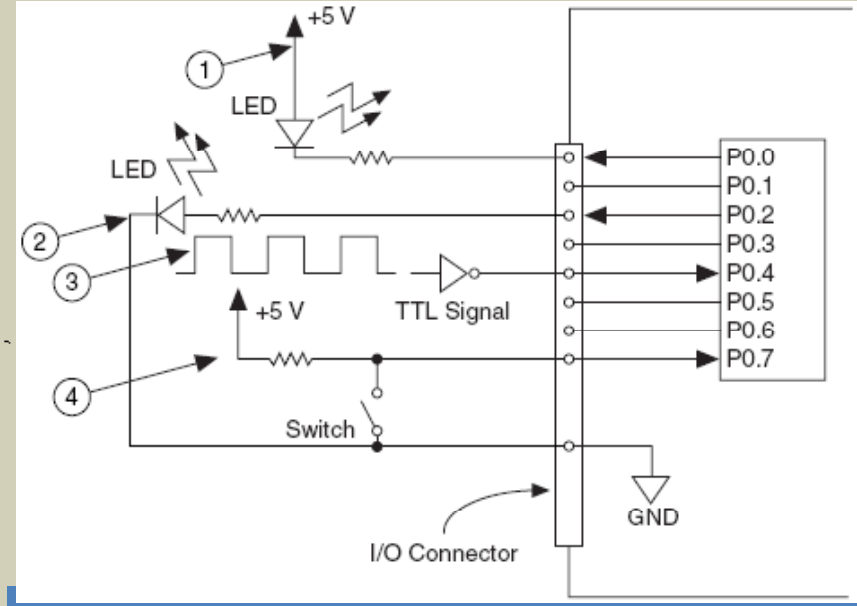
10.3 Saídas Analógicas

Tipo de conversão..... APROXIMAÇÕES SUCESSIVAS
Nº Saídas 2
Resolução..... 12 BITS
Taxa máxima de actualização ..... 150 HZ, SOFTWARE-TIMED
Gama de saída..... 0 TO +5 V
Impedância de saída..... 50 Ω
Corrente de saída .....5 mA
Tensão de inicialização .....0 V
SLEW RATE .....1 V/mS
Corrente de curto-circuito.....50 mA

MODELO INTERNO



10.3 Saídas/Entradas Digitais



1 P0.0 configured as an open collector digital output driving a LED
2 P0.2 configured as a active drive digital output driving a LED
3 P0.4 configured as a digital input receiving a TTL signal from a gated invertor
4 P0.7 configured as a digital input receiving a 0 V or 5 V signal from a switch

P0.<0..7> .....8 LINES

PI.<0..3> .....4 LINES

Direcção de controlo.....CADA CANAL PODE SER PROGRAMADO INDIVIDUALMENTE

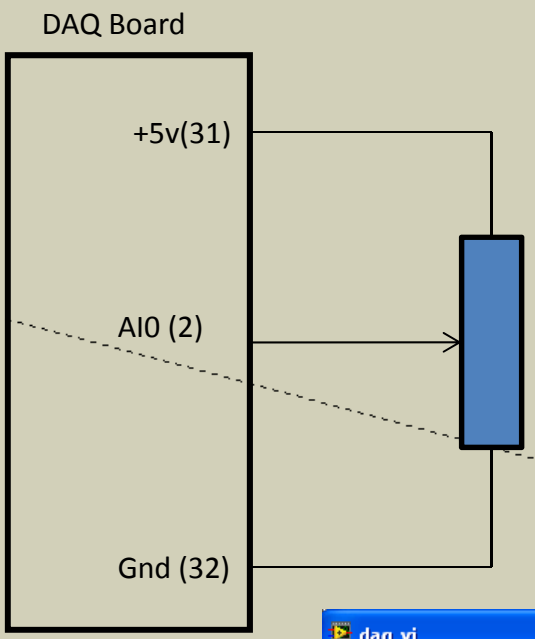
Tipo de saída.....COLECTOR ABERTO (OPEN-DRAIN)

Compatibilidade .....TTL, LVTTL, CMOS

Gama de tensão máxima absoluta .....–0.5 A 5.8 V (RELATIVAMENTE A GND)

Resistências de Pull-Up .....4.7 KΩ

10.4 Ensaio 1: A/D



+2.5 External References

The USB-6008/6009 creates a high-purity reference voltage supply for the ADC using a multi-state regulator, amplifier, and filter circuit. The resulting +2.5 V reference voltage can be used as a signal for self test.

+5 V Power Source

The USB-6008/6009 supplies a 5 V, 200 mA output. This source can be used to power external components.