



UNIVERSIDADE FEDERAL DO PARÁ
DEPARTAMENTO DE ENGENHARIA MECÂNICA
GRUPO DE VIBRAÇÕES E ACÚSTICA



Introdução ao *LabVIEW*



Roberta Tamara da Costa Nery
(Bolsista – GVA)

SUMÁRIO

1.	Introdução ao <i>LabVIEW</i>	3
2.	Introdução aos Instrumentos Virtuais.....	5
3.	Construindo o Painel Frontal.....	21
4.	Construindo o Diagrama de Bloco.....	26
5.	Gráficos.....	53
6.	Programação Estruturada.....	59

1. Introdução ao *LabVIEW*

O *LabVIEW* é um software aplicativo baseado na linguagem G (linguagem de programação gráfica) que emprega ícones ao invés de textos para criar aplicações. Diferente das linguagens de programação baseadas em textos (linhas de comando), onde as instruções determinam a execução do programa.

A programação em *LabVIEW* está baseada no fluxo de dados, onde os dados determinam a execução, trazendo algumas vantagens para aplicações científicas e de engenharia, principalmente em aplicações de aquisição e manipulação de dados.

Os aplicativos são desenvolvidos pelo usuário com o uso de um conjunto de ferramentas e objetos que possuem funções para aquisição, análise e apresentação dos dados, GPIB e controle de instrumentos seriais. Os códigos são adicionados no diagrama de blocos usando representações gráficas de funções para controlar os objetos adicionados no painel frontal. Depois de criado, o diagrama de blocos é compilado para linguagem de máquina.

```

!secao de declaracao
PROGRAM Soma_de_dois_numeros
IMPLICIT NONE
REAL :: a, b, soma
!secao de execucao
!solicitacao de dados
WRITE (*,*) 'Digite os valores'
!entrada de dados
READ (*,*) 'a, b'
!processamento
soma = a + b
!saida de informacoes
WRITE (*,*) 'A soma e igual a', soma
!secao de finalizacao
STOP
END PROGRAM

```

Figura 1 – Exemplo de programação através de linhas de comando

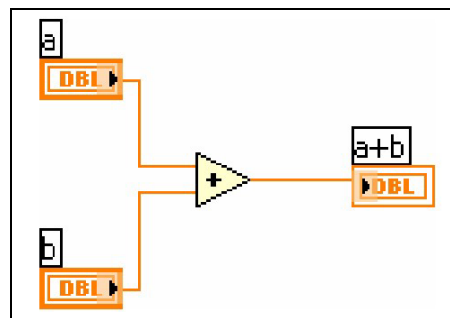


Figura 2 – Programação através de ícones

2. Introdução aos Instrumentos Virtuais

Os programas do *LabVIEW* são chamados de Instrumentos Virtuais, ou VIs, porque sua aparência e operação imitam instrumentos físicos, tais como osciloscópios e multímetros. Cada VI tem funções que manipulam a entrada pela interface do usuário ou de outras fontes e indicam essa informação ou a movem para outros arquivos ou outros computadores.

Um VI contém os três seguintes componentes:

- **Painel Frontal** – Serve como interface do usuário.
- **Diagrama de Bloco** – Contém o código fonte gráfico que define o funcionamento do VI.
- **Ícone e Conector** – Identificam o VI de modo que você possa usar o VI em um outro VI. Um VI dentro de um outro VI é chamado de subVI. Um subVI corresponde a uma sub-rotina em linguagens de programação baseadas em texto.

2.1. Painel Frontal

O painel frontal é a interface do usuário. A Figura 3 ilustra a janela do painel frontal.

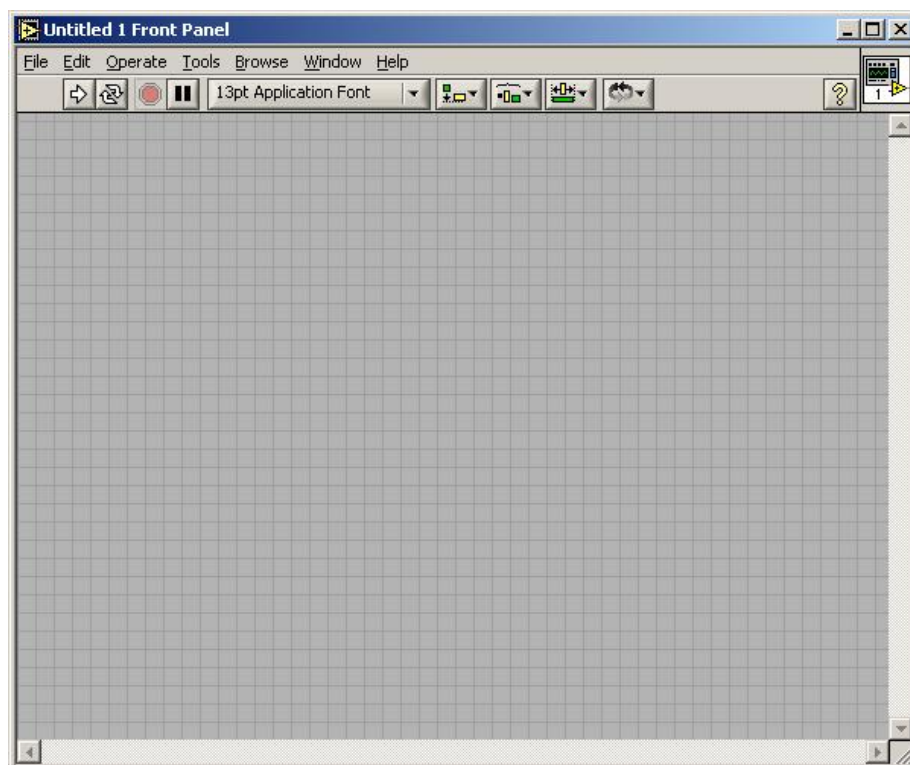


Figura 3 – Janela do Painel frontal

O painel frontal é construído com controles e indicadores, que são os terminais interativos de entrada e de saída do VI, respectivamente. Os controles são dispositivos de entrada, como botões e teclas. Os indicadores são gráficos, LEDs, ou seja, é a saída de informações.

Os controles simulam dispositivos de entrada do instrumento e de dados da fonte do diagrama de bloco do VI. Os indicadores simulam dispositivos de saída do instrumento e visualiza dados que o diagrama de bloco adquire ou gera.

2.2. Diagrama de Bloco

Depois de construir o painel frontal, o código é adicionado usando representações gráficas das funções para controlar os objetos do painel frontal. O diagrama de bloco contém este código fonte gráfico. Os objetos do painel frontal aparecem como terminais no diagrama de bloco. Não é possível apagar um terminal do diagrama de bloco. O terminal desaparece somente depois que o seu objeto correspondente no painel frontal é apagado. A Figura 4 mostra a janela do diagrama de bloco.

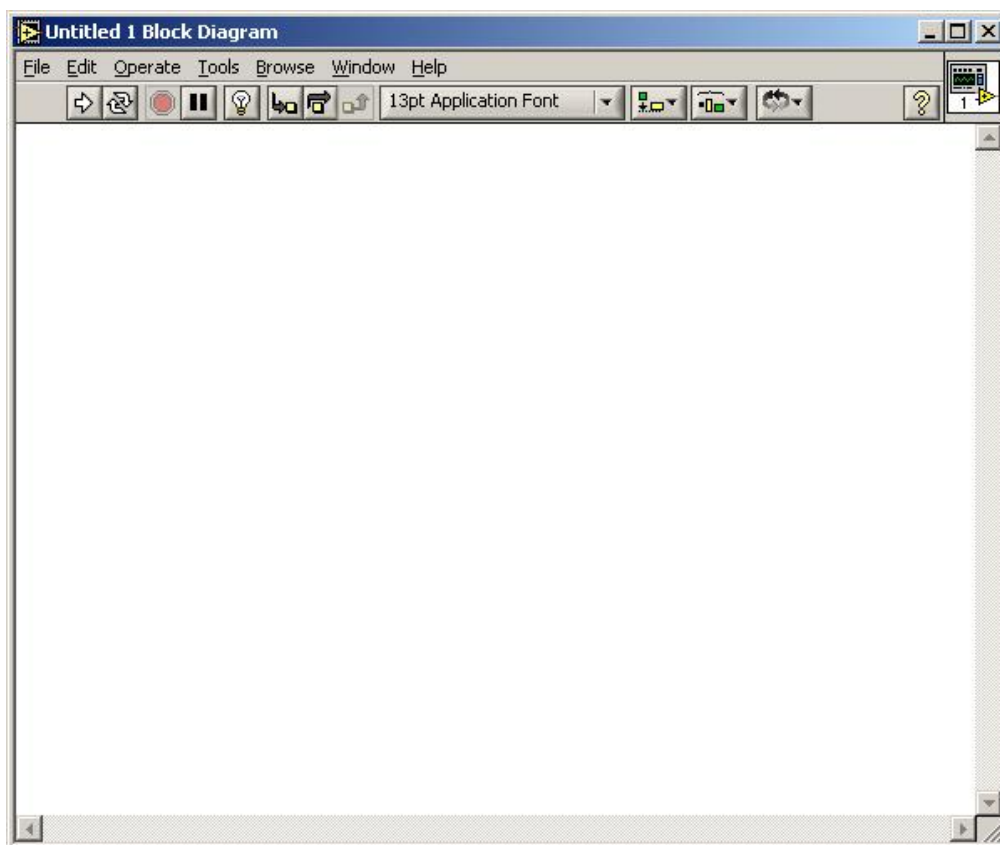


Figura 4 – Janela do Diagrama de bloco

2.2.1. Terminais

Os terminais representam o tipo de dados para o controle ou o indicador. Por exemplo, um terminal DBL representa uma dupla precisão, um controle numérico de um ponto ou um indicador.

2.2.2. Nós

Os nós são os objetos no diagrama de bloco que têm entradas e/ou saídas e executam operações quando os VIs funcionam. São análogos às indicações, aos operadores, às funções, e às sub-rotinas em linguagens de programação baseadas em texto.

2.2.3. Fios

Os dados são transferidos entre objetos do diagrama de bloco através dos fios. Cada fio tem uma única origem de dados, mas é possível usar vários fios para conectar as funções para ler os dados do VI. Os fios têm diferentes cores, estilos e espessuras, dependendo dos seus tipos de dados. Um fio quebrado aparece como uma linha preta tracejada, se houver uma conexão errada.

2.2.4. Estruturas

As estruturas são representações gráficas de laços (*loops*) e de indicações de caso de linguagens de programação baseadas em texto.

2.3. Ícone e Conector

Depois que o painel frontal e o diagrama de bloco do VI estiverem construídos, é feito o ícone e o conector, dessa maneira pode-se usar o VI como um subVI. Cada VI mostra um ícone, no canto direito superior das janelas do painel frontal e do diagrama de bloco. Esse ícone é a representação gráfica de um VI. Pode conter texto, imagens, ou uma combinação de ambos. Se for usado um VI como um subVI, o ícone identifica o subVI no diagrama de bloco do VI. O ícone é o que é visto no diagrama de bloco quando um VI é usado como um subVI.

É necessário também construir um conector, para usar o VI como um subVI. O conector é um conjunto de terminais que corresponde aos controles e aos indicadores daquele VI. O conector define as entradas e saídas que podem ser conectadas ao VI, assim é possível usá-la como um subVI. Um conector recebe dados em seus terminais de entrada e passa os dados ao código do diagrama de bloco com os controles do painel frontal e recebe os resultados em seus terminais de saída dos indicadores do painel frontal.

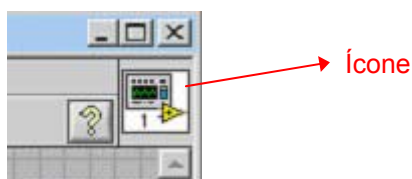


Figura 5

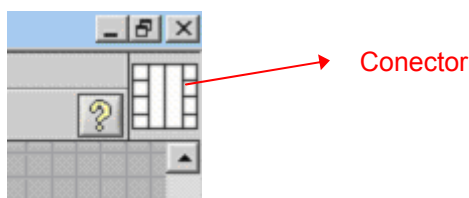


Figura 6

2.4. Paleta de Ferramentas (*Tools*)

A paleta de Ferramentas fica disponível no painel frontal e no diagrama de bloco. Uma ferramenta é um modo especial do cursor do mouse operar e é usada para modificar os objetos do painel frontal e do diagrama de bloco. Quando uma ferramenta é selecionada, o ícone do cursor muda para o ícone da ferramenta.

Selecione *Window>>Show Tools Palette* (Mostrar paleta de ferramentas) para visualizar a paleta de Ferramentas ou tecle <Shift> e clique com o botão direito do mouse para visualizar a paleta de Ferramentas. Você pode colocar a paleta de Ferramentas em qualquer lugar da tela.



Figura 7 – Paleta de ferramentas



Operating: Muda o valor de um controle ou seleciona o texto dentro de um controle.



Positioning: Posiciona, redimensiona e seleciona objetos.



Labeling: Editar textos e criar textos livres.



Wiring: Amarra objetos no diagrama de bloco.



Object Shortcut Menu: Abre um menu de atalho de um objeto.



Scrolling: Move todos os objetos simultaneamente.



Breakpoint: Conjunto de pontos nos VIs, funções, nós, fios, e estruturas para parar a execução no local indicado.



Probe: Use a ferramenta Probe para verificar valores intermediários num VI que gera questões ou resultados inesperados.



Color Copying: Copia cores para colar com a ferramenta Coloring.



Coloring: Conjunto de cores de primeiro plano e de fundo.



Obs: É interessante observar que ao pressionar a tecla <TAB> o cursor alterna entre todas as ferramentas, e através da barra de espaço é possível alternar entre as duas ferramentas mais usadas: a de Operação e a de Posicionamento (caso esteja no Painel Frontal) e entre a de Conexão e a de Posicionamento (caso esteja no Diagrama de Bloco).

2.5. Paleta de Controles (*Controls*)

A paleta de Controles fica disponível somente no painel frontal. Selecione *Window>>Show Controls Palette* (Mostrar paleta de controle) ou clique com o botão direito do mouse na área de trabalho do painel frontal para

visualizar a paleta de Controle. Você pode colocar a paleta de Controle em qualquer lugar da tela.



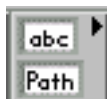
Figura 8 – Paleta de controles



Subpaleta **Numeric**. Consiste de controles e indicadores para dados numéricos.



Subpaleta **Boolean**. Consiste de controles e indicadores para valores Boolean.



Subpaleta **String**. Consiste de controles e indicadores para strings e tables.



Subpaleta **Array & Cluster**. Consiste de controles e indicadores para grupos do mesmo tipo de dados.



Subpaleta **List & Ring**. Consiste de controles e indicadores para menu rings e listboxes.



Subpaleta **Graph**. Consiste de indicadores e controles para plotar dados em gráficos.



Subpaleta **Ring & Enum Controls**. Cria uma lista de strings.

Subpaleta **Containers**. Usados para criar guias, como fichários.



Subpaleta **I/O Controls and Indicators**. Usada para se comunicar com um instrumento ou um dispositivo da aquisição de dados



Subpaleta **Dialog Controls**. Os controles e os indicadores do diálogo são projetados especificamente para o uso em caixas de diálogo.



Subpaleta **Front Panel Controls and Indicators**.



Subpaleta **Refnum Controls**. Trabalha com arquivos, diretórios, dispositivos e conexões de rede.

Subpaleta **Decorations**. Usado para agrupar ou separar objetos de um painel frontal com caixas, linhas, ou setas. Estes objetos são para a decoração somente e não indicam dados.



Subpaleta **Select a Control**. Usado para selecionar um controle que você projetou.



Subpaleta **User Controls**. Usado para mostrar automaticamente controles customizados no diretório user.lib.

2.6. Paleta de Funções (*Functions*)

A paleta de Funções fica disponível somente no diagrama de bloco. Selecione *Window>>Show Functions Palette* (Mostrar paleta de funções) ou clique com o botão direito do mouse na área de trabalho do diagrama de bloco para visualizar a paleta de Funções. Você pode colocar a paleta de Funções em qualquer lugar da tela.



Figura 9 – Paleta de funções



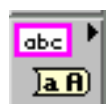
Subpaleta **Structures**. Consiste em estruturas de controle do programa como Laços.



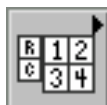
Subpaleta **Numeric**. Consiste de funções aritméticas, trigonométricas, logarítmicas e numéricas.



Subpaleta **Boolean**. Use as funções booleanas para executar funções lógicas, como TRUE/FALSE.



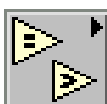
Subpaleta **String**. Use as funções string para criar e manipular strings, tais como **Concatenate Strings**, **Format into String**, e **Array to Spreadsheet String**.



Subpaleta **Array**. Consiste de funções para processar arrays.



Subpaleta **Cluster**. Consiste de funções para processamento de clusters.



Subpaleta **Comparison**. Consiste de funções para comparar números, Booleans, e strings.



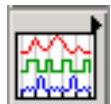
Subpaleta **Time & Dialog**. Consiste de funções para criar caixas de diálogo, tempo, e manipulação de erro.



Subpaleta **File I/O**. Consiste de funções e VIs para arquivo I/O.



Subpaleta **NI Measurements**. Use Measurements VIs e funções para criar interfaces com Traditional NI-DAQ, NI-DAQmx, e outros dispositivos de aquisição.



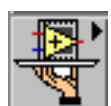
Subpaleta **Waveform**. Use VIs waveform e as funções para executar operações do waveform.



Subpaleta **Analyze**. Consiste de VIs de análise de dados.



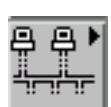
Subpaleta **Instrument I/O**. Consiste de VIs para GPIB, serial, e VISA instrument control.



Subpaleta **Application Control**. Consiste de funções e VIs programados para imprimir, mudar menus do *LabVIEW*, mostrar a janela Help, e parar ou sair do *LabVIEW*.



Subpaleta **Graphics & Sound**. Cria displays, importa e exporta dados de arquivos gráficos e de sons.



Subpaleta **Communication**. Troca dados entre as aplicações.



Subpaleta **Report Generation**. Cria e manipula relatórios das aplicações do *LabVIEW*.



Subpaleta **Advanced**. Chama códigos da biblioteca, manipular dados para outras aplicações do *LabVIEW*, e chamar o conjunto de códigos como em linguagens de programação baseadas em texto.



Subpaleta **Select a VI...**. É uma caixa de diálogo que serve para inserir subVIs no VI atual.



Subpaleta **Decorations**. Usado para agrupar ou separar objetos de um diagrama de bloco com caixas, linhas, colunas ou textos. Estes objetos são para a decoração somente e não indicam dados.



Subpaleta **User Libraries**. Ver as aplicações do VI que você construiu e salvou.

2.7. Barra de Menus e Barra de Ferramentas

Os itens dos menus e da barra de ferramentas são usados para operar e modificar objetos do painel frontal e diagrama de bloco. Existem botões na barra de ferramentas que executam os VIs.

2.7.1. Barra de Menus

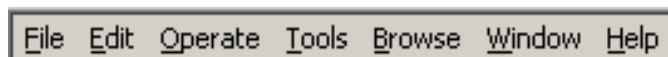


Figura 10 – Barra de menus

A barra de menus no alto da tela do *LabVIEW* contém diversos menus. Estes menus contêm as opções comuns à maioria das aplicações, tais como **Abrir, Salvar, Copiar e Colar**, bem como, outras restritas ao *LabVIEW*.

2.7.1.1. Menu Arquivo (*File*)

As opções no menu **Arquivo** são usadas principalmente para abrir, fechar, salvar e imprimir VIs.

- **New VI** → Cria um novo VI.
- **New** → Abre uma caixa de diálogo para que seja selecionado um tipo de arquivo a ser criado.
- **Open** → Abre um VI já feito.
- **Close** → Fecha o arquivo atual. Uma caixa de diálogo de confirmação aparece para que sejam salvas todas as alterações feitas no VI.
- **Close All** → Fecha todos os VIs abertos.
- **Save** → Salva o VI atual. Se um VI novo estiver sendo salvo pela primeira vez, uma caixa de diálogo aparece dando a opção de nomear o arquivo e determinar sua localização.
- **Save As** → Permite salvar o arquivo atual com um nome, um tipo, ou uma localização diferentes.
- **Save All** → Salva todos os arquivos abertos.
- **Save A Copy As** → Salva uma cópia do VI com um novo nome.
- **Save with Options** → Permite salvar o VI atual usando as seguintes opções:
 - **Changed VIs** → Salva todas as mudanças do VI atual e seus subVIs.
 - **Development Distribution** → Salva todos os VIs em bibliotecas, em controles, e em sub-rotinas externas a uma única localização, um diretório ou biblioteca.

- **Application Distribution** → Salva todos os VIs, controles, e sub-rotinas externas, incluindo aqueles VIs em bibliotecas, a uma única localização e remove os diagramas de bloco de todos os VIs.
- **Custom Save** → Permite selecionar as opções específicas que se deseja da área Selecionada de Opções da caixa de diálogo.
- **Template** → Salva o VI como um formato do VI com a extensão de arquivo .vit. Quando um formato do VI é carregado, cria uma cópia que deva ser salva sob um novo nome.
- **Revert** → Rejeita todas as mudanças feitas desde a última vez o arquivo foi salvo.
- **Page Setup** → Permite configurar a impressora e modificar como a documentação de VI deve ser impressa.
- **Print** → Acessa uma caixa de diálogo, que mostra diversas opções para imprimir a documentação.
- **Print Window** → Imprime a janela atual.
- **VI Properties** → Acessa a caixa de diálogo das propriedades do VI, que permite que uma senha seja criada para o arquivo e modificar uma descrição do arquivo. A caixa de diálogo das propriedades do VI também indica o número de revisão do arquivo e fornece uma vista geral do uso da memória do arquivo.
- **Recently Opened Files** → Permite que abra rapidamente os arquivos mais recentemente abertos.
- **Exit** → Fecha o programa.

2.7.1.2. Menu Editar (*Edit*)

As opções no menu **Editar** são usadas para modificar o painel frontal e objetos do diagrama de bloco de um VI.

- **Undo** → Desfaz a última ação executada.
- **Redo** → Refaz a última ação executada.
- **Cut** → Recorta o objeto selecionado e coloca ele na área de transferência.
- **Copy** → Copia o objeto selecionado e coloca ele na área de transferência.
- **Paste** → Cola uma cópia do conteúdo da área de transferência para a janela ativa.

- **Clear** → Apaga o objeto selecionado.
- **Find** → Acessa uma caixa de diálogo, que permite procurar pelos VIs, pelos outros objetos, ou pelo texto.
- **Show Search Results** → Acessa a caixa de diálogo dos resultados da busca, mostra os resultados da busca e permitem que acessar cada artigo na lista.
- **Customize Control** → Permite que modificar o objeto atual selecionado do painel frontal e salvar o arquivo resultante com o sufixo do ctl.
- **Scale Object With Panel** → Faz o objeto selecionado do painel frontal redimensionar para mesma escala da janela do painel frontal, sempre que a janela do painel frontal redimensionar.
- **Set Tabbing Order** → Ajusta a ordem de objetos do painel frontal.
- **Import Picture from File** → Importa uma figura no VI.
- **Remove Broken Wires** → Apaga todos os fios quebrados do VI atual.
- **Create SubVI** → Cria um subVI novo dos objetos selecionados.
- **Run-Time Menu** → Cria um novo menu para o VI atual, menus *run-time*.

2.7.1.3. Menu Operar (*Operate*)

Os comandos no menu **Operar** são usados para executar o VI.

- **Run** → Executa o VI.
- **Stop** → Para o VI antes que termine a execução.
- **Suspend when Called** → Faz com que o VI suspenda a execução quando é chamado como um subVI.
- **Print at Completion** → Imprime o painel frontal resultante quando o VI termina a execução.
- **Log at Completion** → Executa registro de dados quando o VI termina a execução.
- **Data Logging** → Acessa o registro, recupera e remove dados.
- **Make Current Values Default** → Armazena os valores atuais dos controles e das constantes como valores de defeito.
- **Reintialize All to Default** → Retorna todos os controles e constantes atuais a seus valores de defeito.
- **Change to Run Mode** → Muda o VI para o modo de funcionamento.

2.7.1.4. Menu Ferramentas (*Tools*)

As opções no menu **Ferramentas** são usadas para configurar os projetos e VIs, no *LabVIEW*.

- **Measurement & Automation Explorer** → Acessa o *Measurement & Automation Explorer* para configuração de placa de aquisição.
- **Instrumentation** → Acessa a página na *Internet* da instrumentação instalada, atualiza *drivers*.
- **Data Acquisition** → Acessa uma janela de aquisição de dados.
- **VI Revision History** → Acessa a caixa de diálogo do *VI Revision History*, que permite documentar mudanças ao VI atual.
- **User Name** → Permite ajustar ou mudar o nome do usuário do *LabVIEW*.
- **Edit VI Library** → Permite editar os índices da biblioteca VI atual (LLB) ou criar um novo.
- **Web Publishing Tool** → Acessa a ferramenta *Web Publishing*.

2.7.1.5. Menu Browse

As opções do menu **Browse** são usadas para verificar aspectos do VI atual e a sua hierarquia.

- **Show VI Hierarchy** → Acessa a janela de hierarquia do VI.
- **This VI's Callers** → Permite acessar uma lista de todos os VIs que chamam o VI atual como um subVI.
- **This VI's SubVIs** → Permite acessar uma lista de todos os subVIs ao VI atual.
- **Unopened SubVIs** → Permite acessar uma lista de todos os subVIs fechados ao VI atual.
- **Unopened Type Defs** → Permite acessar uma lista de todas as definições fechadas do tipo ao VI atual

2.7.1.6. Menu Janela (*Window*)

O menu **Janela** é usado para encontrar rapidamente janelas abertas e abrir janelas dos sub VIs e VIs chamados.

- **Show Diagram / Show Panel** → Interruptores entre o painel frontal e a janela do diagrama de bloco do VI atual.
- **Show Controls Palette** → Mostra a paleta de controles. Na janela do diagrama de bloco, este item é mudado para paleta de funções, e mostra a paleta de funções.
- **Show Tools Palette** → Mostra a paleta de ferramentas.
- **Show Clipboard** → Mostra a área de transferência dos objetos copiados.
- **Show Error List** → Acessa uma caixa de diálogo que mostra os erros no VI atual.
- **Tile Left and Right** → Arranja as janelas abertas da esquerda para a direita.
- **Tile Up and Down** → Arranja as janelas abertas do alto ao fundo.
- **Full Size** → Faz a janela atual expandir para preencher toda tela.

2.7.2. Barra de Ferramentas do Painel Frontal

As janelas do Painel e do Diagrama contêm uma barra de ferramentas com teclas de comando e com indicadores de status que são usados controlando o VI. Uma das duas barras de ferramentas está disponível, dependendo se está trabalhando na janela do Painel ou do Diagrama. A seguinte barra de ferramentas aparece no alto da janela do Painel.



Figura 11 – Barra de ferramentas do painel frontal



Run. Executa o VI. Quando os VIs estão executando; a tecla muda para:




List Errors. Esta tecla substitui a tecla *Run* e indica que o VI não pode compilar devido aos erros. Para visualizar os erros, clique nesta tecla. Acima da janela são listados todos os erros.





Abort Execution. Esta tecla é habilitada quando os VIs estão executando. Clique nesta tecla para parar imediatamente o VI.




Run Continuously. Clique nela para executar o VI continuamente.

 Quando o VI é executado no modo de funcionamento contínuo, o símbolo muda como mostrado na esquerda. Clique nesta tecla para desabilitar o funcionamento contínuo.

 *Pause/Continue*. Esta tecla pausa a execução VI. Para sair do modo de pausa, pressione a tecla outra vez, e o VI continua a execução.

 *Text Settings*. Esta caixa ajusta opções de fonte, incluindo o tipo de fonte, o tamanho, o estilo e a cor.

 *Align Objects*. Deve-se usar a ferramenta *position/size/select* para selecionar os objetos a serem alinhados. Os objetos podem ser alinhados pela borda superior, inferior, esquerda, direita ou pelo centro horizontal ou vertical, para dois ou mais objetos. A Figura 12 mostra os possíveis modos de alinhamento dos objetos.

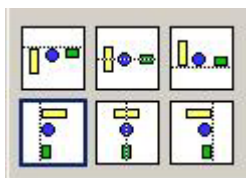



Figura 12 – Alinhamento dos objetos

 *Distribute Objects*. Deve-se usar a ferramenta *position/size/select* para selecionar os objetos a serem distribuídos. A Figura 13 mostra os possíveis modos de distribuição dos objetos.

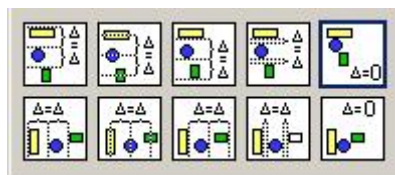



Figura 13 – Distribuição dos objetos

 *Resize Objects*. Deve-se usar a ferramenta *position/size/select* para selecionar os objetos a serem redimensionados. Um objeto pode ser reduzido ou ampliado para o mesmo tamanho de outro objeto previamente selecionado. A Figura 14 mostra as possíveis maneiras de redimensionar os objetos.

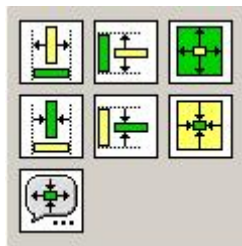


Figura 14 – Redimensionamento dos objetos



Reorder. Esta opção deve ser usada quando se tem objetos que se sobrepõem e se deseja definir qual objeto fica na frente ou atrás do outro. Selecione um dos objetos com a ferramenta *position/size/select* e depois selecione para mover para frente ou para trás. É possível também agrupar objetos.

2.7.3. Barra de ferramentas do diagrama de bloco

A barra de ferramentas do Diagrama de Bloco contém a maioria das teclas que a barra de ferramentas do Painel Frontal. Além dessas há quatro teclas características.

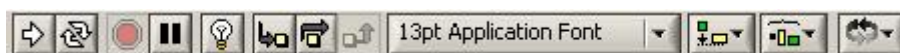


Figura 15 – Barra de ferramentas do diagrama de bloco



Highlight Execution. Ao clicar nesta tecla, ela muda para a seguinte tecla.



Do not Highlight Execution. Neste modo é possível visualizar o fluxo dos dados através do Diagrama de Bloco. Basta clicar novamente para desabilitar o destaque da execução.



Start Single Stepping. Pisca cada nó para denotar que o nó está pronto para executar. Clique nesta tecla outra vez para piscar um laço, sub VI, e assim por diante.



Start Single Stepping. Pisca cada nó para denotar que o nó está pronto para executar. Clique nesta tecla outra vez para piscar um laço, sub VI, e assim por diante.



Step Out. Clique nesta etapa na tecla outra vez para piscar um laço, sub VI, e assim por diante.

3. Construindo o Painel Frontal

O painel frontal é a interface do usuário de um VI. Geralmente, cria-se primeiro o painel frontal, a seguir é criado o diagrama de bloco para executar tarefas nas entradas e nas saídas criadas no painel frontal.

O painel frontal é construído com controles e indicadores, que são os terminais interativos de entrada e de saída do VI, respectivamente. Os controles são botões, teclas, seletores, e outros dispositivos de entrada. Os indicadores são gráficos, LEDs, e outras visualizações. Os controles simulam dispositivos de entrada do instrumento. Indicadores simulam dispositivos de saída do instrumento e visualização de dados que o diagrama de bloco adquire ou gera.

Selecione *Window>>Show Controls Palette* para visualizar a paleta de Controles. Selecione controles e indicadores da paleta de Controles e coloque-os no painel frontal.

3.1. Configurando objetos no painel frontal

Pode-se melhorar o painel frontal usando os menus de controles e de indicadores, ajustando a ordem de objetos do painel frontal, e usando gráficos. É possível também mover manualmente objetos do painel frontal e ajustá-los para mover automaticamente quando o tamanho da janela muda.

3.2. Substituindo os objetos do painel frontal

Um objeto do painel frontal pode ser substituído por um controle ou um indicador diferente. Quando você clica com o botão direito do mouse num objeto e seleciona *Replace* (Substitui) no menu de atalho, uma paleta de Controles provisória aparece, mesmo se a paleta de Controles já estiver aberta. Selecione um controle ou um indicador da paleta de Controles provisória para substituir o objeto atual no painel frontal.

Selecionando *Replace* (Substitui) no menu de atalho as informações possíveis sobre o objeto original são preservadas, tais como seu nome, descrição, dados de defeito, sentido de fluxo de dados (controle ou indicador), cor, tamanho, e assim por diante. Entretanto, o objeto novo retém seu próprio

tipo de dados. Os fios do terminal do objeto ou das variáveis locais permanecem no diagrama de bloco, mas podem ser quebrados.

É possível também colar objetos da área de transferência para substituir controles e indicadores existentes do painel frontal. Este método não preserva nenhuma característica do objeto velho, mas os fios permanecem conectados ao objeto.

3.3. Ajustando atalhos do teclado para controles

É possível atribuir teclas de atalho aos controles para os usuários poderem navegar pelo painel frontal sem usar o mouse. Basta dar um clique com o botão direito do mouse no controle e selecionar *Advanced>>Key Navigation* do menu de atalho para abrir a caixa de diálogo *Key Navigation*.

Quando um usuário insere uma tecla de atalho para funcionar o VI, o controle associado recebe o foco. Se o controle for um controle de texto, o *LabVIEW* destaca o texto, podendo ser editado. Se o controle for *Boolean*, a aparência da tecla muda.

O item *Advanced>>Key Navigation* do menu de atalho fica desabilitado para indicadores, porque não é possível introduzir dados a um indicador.

3.4. Ajustando a ordem de navegação dos objetos no painel frontal

É possível ajustar a ordem de navegação de objetos do painel frontal selecionando *Edit>>Set Tabbing Order*. Pode então usar a tecla *<Tab>* para passar de um controle para outro de acordo com a ordem especificada, quando o VI estiver funcionando.

Os controles e os indicadores em um painel frontal têm uma ordem, chamada “a ordem do painel”, que é a sua posição no painel frontal. O primeiro controle ou indicador criado no painel frontal é o elemento 0, o segundo é 1, e assim por diante. Se um controle ou um indicador for retirado, a ordem do painel ajusta automaticamente.

A ordem do painel determina a ordem de navegação quando um VI funcionar. A ordem do painel determina também a ordem em que os controles

e os indicadores aparecem nos registros de arquivos que você cria quando você registra os dados do painel frontal.

Para impedir que os usuários acessem um controle usando a tecla <Tab> ao funcionar o VI, marque a opção *Skip this control when tabbing* na caixa de diálogo *Key Navigation*, como pode ser visto na Figura 16.

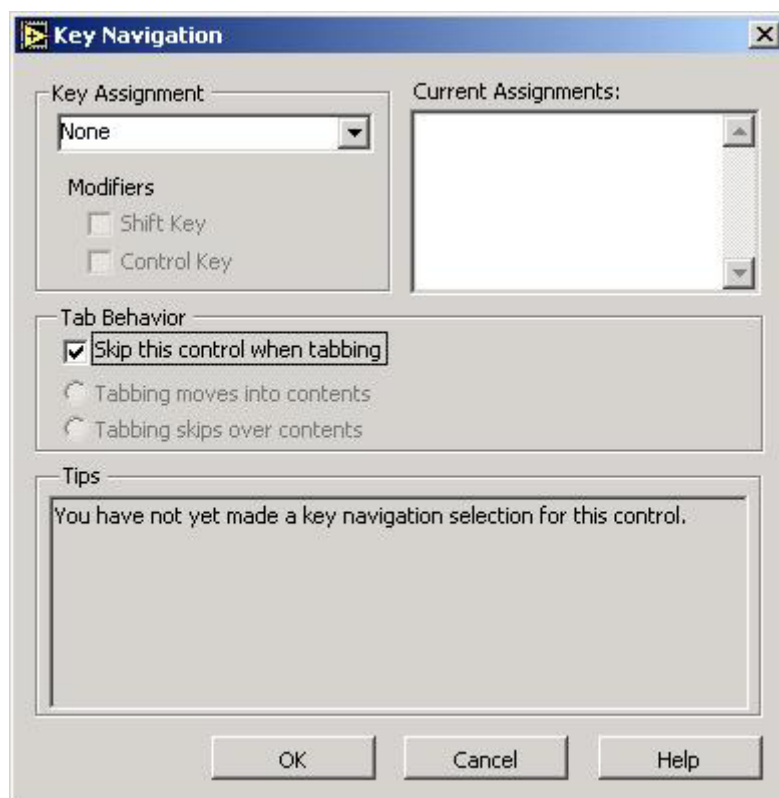


Figura 16

3.5. Colorindo objetos

É possível mudar a cor de muitos objetos, mas não de todos. Por exemplo, os terminais do diagrama de bloco de objetos do painel frontal e os fios usam cores específicas para o tipo e a representação dos dados que carregam, portanto eles não podem ser alterados.



Figura 17 – Caixa de cores

3.6. Redimensionando objetos

O tamanho da maioria de objetos do painel frontal pode ser alterado. Quando se move a ferramenta de Posicionamento sobre um objeto, marcações aparecem nos cantos de um objeto retangular, e círculos aparecem em um objeto circular. Quando um objeto é redimensionado, o tamanho do terminal permanece o mesmo.

Alguns objetos mudam o tamanho somente horizontalmente ou verticalmente quando são redimensionados, tais como controles numéricos digitais e indicadores. Outros mantêm as mesmas proporções ao serem redimensionados, tais como botões. Manualmente pode-se restringir o sentido do crescimento quando um objeto é redimensionado. Para redimensionar um objeto em torno de seu ponto central, pressione <Ctrl> e clique nas marcações de redimensionamento.

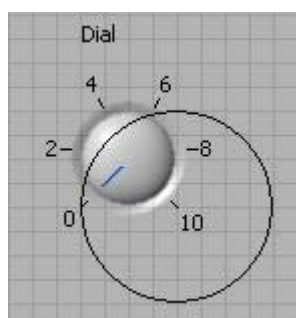


Figura 18 – Redimensionando objetos sem pressionar <Ctrl>

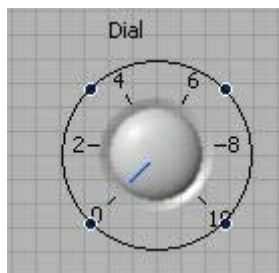


Figura 19 - Redimensionando objetos pressionando <Ctrl>

3.7. Escalando objetos do painel frontal

Os objetos do painel frontal podem ser ajustados à escala, ou redimensionados automaticamente com relação ao tamanho da janela, quando a janela do painel frontal é restaurada ou maximizada. Entretanto, vários objetos não podem ser ajustados à escala do painel frontal, a menos que os objetos sejam agrupados primeiramente. Para escalar um objeto, selecione o objeto e selecione *Edit>>Scale Object with Panel* no painel.

Se um único objeto do painel frontal for ajustado à escala, o objeto redimensiona automaticamente em relação a qualquer mudança no tamanho da janela do painel frontal. Os outros objetos no painel frontal se reposicionam para permanecerem consistentes com sua colocação precedente no painel frontal, mas não se escalam de acordo com o tamanho da janela nova do painel frontal.

Quando o *LabVIEW* escala objetos automaticamente, segue as mesmas convenções de quando um objeto é redimensionado manualmente. Por exemplo, alguns objetos podem redimensionar somente horizontalmente ou verticalmente.

4. Construindo o Diagrama de Bloco

Depois que o painel frontal estiver pronto, códigos devem ser adicionados usando representações gráficas de funções para controlar os objetos do painel frontal.

4.1. Relacionando objetos do painel frontal e terminais do diagrama de bloco

Os objetos do painel frontal aparecem como terminais no diagrama de bloco. Um duplo clique num terminal do diagrama de bloco mostra o controle ou indicador correspondente no painel frontal. O terminal desaparece somente quando o seu objeto correspondente no painel frontal é deletado.

4.2. Objetos no Diagrama de Bloco







Objetos no diagrama de bloco incluem terminais, nós, e funções. O diagrama de bloco é construído conectando os objetos com a ferramenta *Wire*.




4.3. Terminais do Diagrama de Bloco

Terminais do diagrama de bloco representam o tipo de dado do controle ou indicador do painel frontal. Por exemplo, um terminal DBL representa uma dupla precisão, um controle numérico ou um indicador. Usa-se fios para conectar terminais e passar dados a outros terminais.

4.4. Tipo de Dados do Controle e Indicador

A tabela mostra os símbolos para os tipos diferentes de terminais do controle e do indicador. A cor e o símbolo de cada terminal indicam o tipo de dados do controle ou do indicador. Os terminais do controle têm uma borda mais espessa do que terminais do indicador.

Controle	Indicador	Tipo de Dado
		Número com ponto-flutuante de simples precisão
		Número com ponto-flutuante de dupla precisão
		Número com ponto-flutuante de precisão estendida

		Número com ponto-flutuante de simples precisão complexa
		Número com ponto-flutuante de dupla precisão complexa
		Número com ponto-flutuante de precisão estendida complexa
		Número inteiro codificado de 8-bit
		Número inteiro codificado de 16-bit
		Número inteiro codificado de 32-bit
		Número inteiro não codificado de 8-bit
		Número inteiro não codificado de 16-bit
		Número inteiro não codificado de 32-bit
		Tipo Enumerado (Conjunto ordenado de identificadores)
		<i>Booleano</i> (Valores Lógicos – Verdadeiro ou Falso)
		<i>String</i> – Cadeia de Caracteres
		<i>Array</i> – É semelhante a uma matriz. Todos os elementos de vem ter o mesmo tipo de dados.
		<i>Cluster</i> – Inclui diversos tipos de dados. O tipo de dados é marrom se os elementos do conjunto forem do mesmo tipo ou cor-de-rosa se os elementos do conjunto forem de tipos diferentes.
		<i>Path</i>
		<i>Waveform</i> – Representa forma de ondas
		<i>Refnum</i> – Número de Referência
		<i>Variant</i> – Inclui o controle ou o nome do indicador, a informação do tipo de dados, e os dados próprios.
		<i>Polymorphic</i> – Indica que um VI ou uma função aceita mais do que uma variável do tipo de dados.
		<i>I/O name</i> – <i>DAQ channel names</i> , <i>VISA resource names</i> , e <i>IVI logical names</i> permite configurar os VIs para comunicação com um instrumento ou um dispositivo de aquisição de dados.
		<i>Picture</i> – Visualiza figuras que podem conter linhas, círculos, textos, e outros tipos de formas gráficas.



Obs: Cada tipo de variável corresponde a um fio de cor diferente no diagrama de blocos.

4.5. Constantes

As constantes são os terminais no diagrama de bloco que fornecem valores fixos dos dados ao diagrama de bloco. As constantes universais são constantes com valores fixos, tais como o pi (π) e o infinito (∞). As constantes definidas pelo usuário são constantes definidas e editadas antes que o VI seja executado.

4.5.1. Constantes Universais

Constantes universais são usadas para computações matemáticas e *strings* ou *paths*. O LabVIEW inclui os seguintes tipos de constantes universais:

- **Universal numeric constants** – Um conjunto de valores matemáticos e físicos de precisão elevada e, geralmente, usados como base natural do logaritmo. Esta constante está localizada na paleta *Functions>>Numeric>>Additional Numeric Constants*.

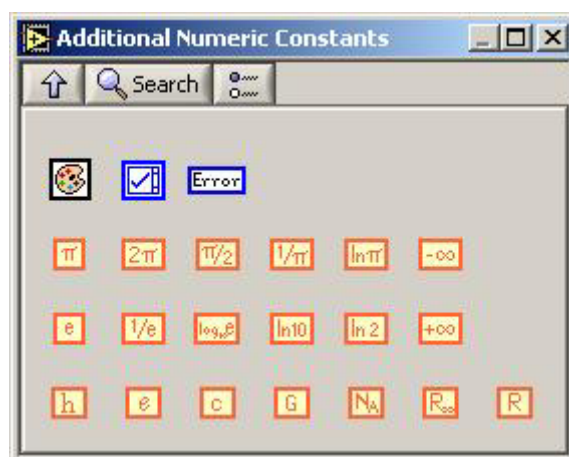


Figura 20

- **Universal string constants** – Um conjunto de caracteres *string*. Esta constante localizada na paleta *Functions>>String*.



Obs: A constante tem o mesmo valor contido no controle ou indicador do painel frontal.

4.6. Nós do Diagrama de Bloco

Nós são objetos do diagrama de bloco que têm entrada e/ou saídas e executam operações quando o VI está funcionando. Eles são análogos às indicações, operações, funções e subrotinas de linguagens de programação baseadas em linhas de comando. O *LabVIEW* inclui os seguintes tipos de nós:

- **Functions** – Construídas nos elementos de execução, semelhantes a um operador, a uma função, ou a uma indicação.
- **Sub VIs** – VIs usados no diagrama de bloco de outros VIs, semelhantes às subrotinas.
- **Structures** – Elementos de controle do processo, tais como *Sequence Structures*, *Case Structures*, *For Loops*, ou *While Loops*.
- **Formula Nodes** – Estruturas para equações serem inseridas diretamente num diagrama de bloco.
- **Property Nodes** – Conjunto ou de propriedades de uma classe.
- **Invoke Nodes** – Executa métodos de uma classe.
- **Code Interface Nodes (CINs)** – Chama códigos para linguagens de programação baseadas em texto.

4.7. Conhecendo algumas funções no diagrama de bloco

4.7.1. Subpaleta Numeric

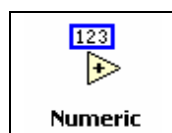


Figura 23

Use as funções da subpaleta *Numeric* para criar e executar operações aritméticas, trigonométricas, logarítmicas e matemáticas complexas e para converter números de um tipo de dado em outro. Veja alguns exemplos desta subpaleta:

4.7.1.1. Add

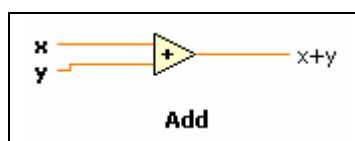


Figura 24

A função *Add* faz a soma de dois valores de entrada. Há também as funções de subtração, multiplicação e divisão.



Figura 25 – Este VI executa uma soma entre dois valores de entrada

4.7.1.2. Increment

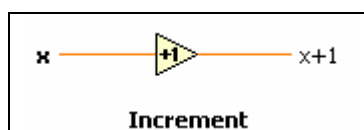


Figura 26

A função *Increment* acrescenta 1 ao valor de entrada.

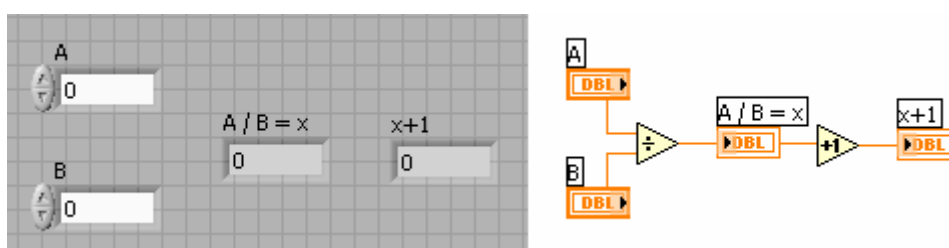


Figura 27 – Este VI acrescenta 1 ao valor de entrada

4.7.1.3. Compound Arithmetic

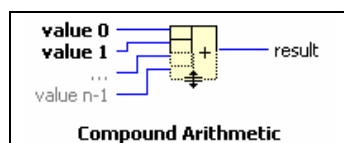


Figura 28

A função *Compound Arithmetic* faz operações entre valores numéricos, *arrays*, *cluster* e valores *booleanos*. É possível mudar o tipo de operação a ser feita e pode-se redimensionar para o número desejado de *Inputs*.

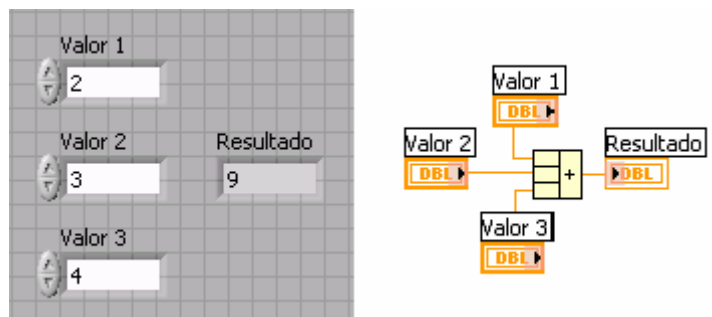


Figura 29 – Este VI faz a soma de três valores

4.7.1.4. Trigonometric

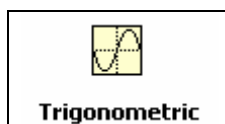


Figura 30

Use as funções trigonométricas para criar e executar equações trigonométricas.

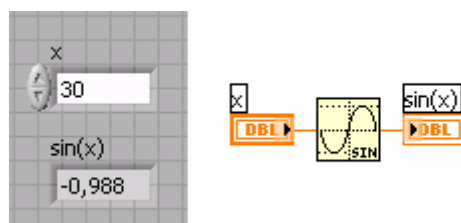


Figura 31 – Este VI calcula o seno de x, onde x está em radianos.

4.7.1.5. Logarithmic

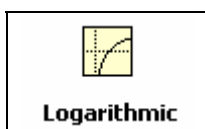


Figura 32

Use as funções logarítmicas para criar e executar equações logarítmicas.

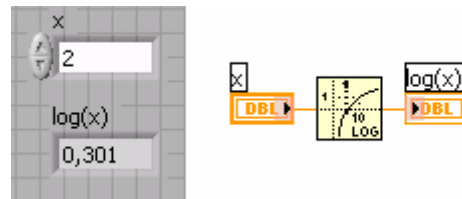


Figura 33 – Este VI calcula o logaritmo de x na base 10

4.7.1.6. Random Number (0 – 1)

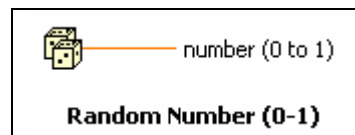


Figura 34

A função *Random Number* produz um número com ponto-flutuante de dupla-precisão, entre 0 e 1, exclusivamente. A distribuição é uniforme.

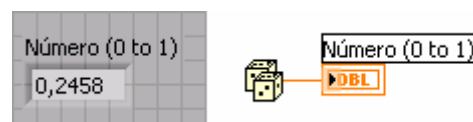


Figura 35 – Este VI produz um número aleatório de 0 a 1

4.7.1.7. Square Root

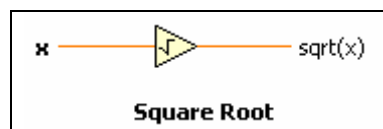


Figura 36

Calcula a raiz quadrada de um número.

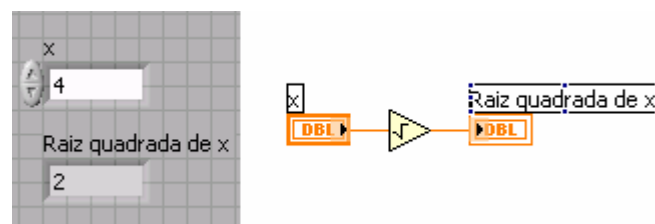


Figura 37 – Este VI calcula a raiz quadrada de um número

4.7.2. Subpaleta Boolean



Figura 38

As funções da subpaleta *Boolean* executam operações com valores lógicos, do tipo verdadeiro ou falso.

4.7.2.1. And

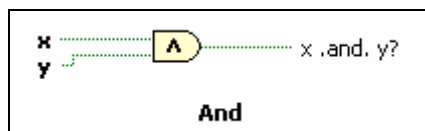


Figura 39

Calcula o valor lógico *AND* entre os valores de entrada. Ambas as entradas são *Booleanas* ou numéricas. Se os dois valores são verdadeiros (*TRUE*), o resultado será verdadeiro (*TRUE*), caso contrário o resultado será falso (*FALSE*).

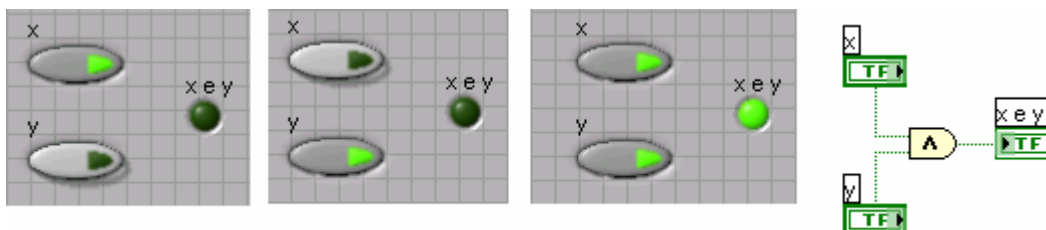


Figura 40 – Neste VI, se só um botão estiver ligado o indicador não acende. Ele só acende se os dois botões estiverem acionados

4.7.2.2. Or

Calcula o valor lógico *OR* entre os valores de entrada. Se ambas as entradas são falsas (*FALSE*), o resultado será falso (*FALSE*). Caso contrário, o resultado será verdadeiro (*TRUE*).

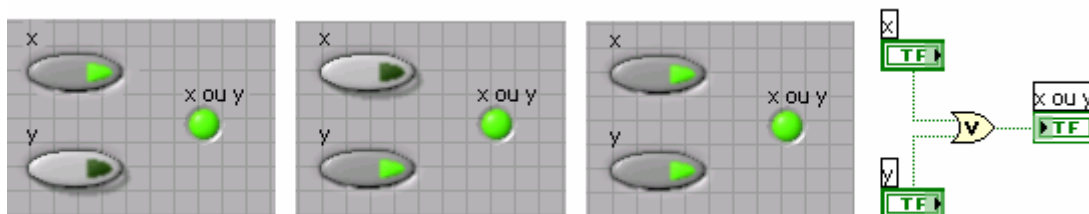


Figura 41 – Neste VI, tanto faz um ou dois botões estarem acionados que o indicador acende

Através da ferramenta *PROBE* colocada nos fios que conectam os controles e os indicadores é possível entender melhor como a função *OR* é executada, e também como a ferramenta *PROBE* funciona.

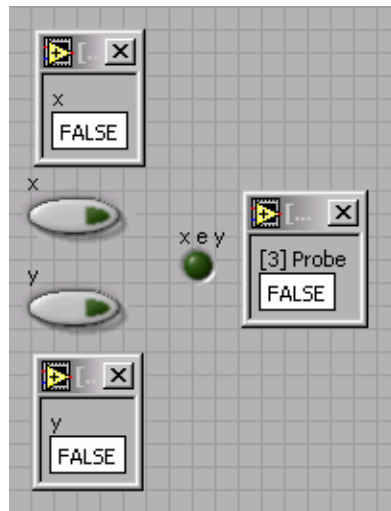


Figura 42 – Os dois valores de entrada (x e y) são falsos (*FALSE*), como é mostrado através da ferramenta *Probe*, e o seu resultado também é falso

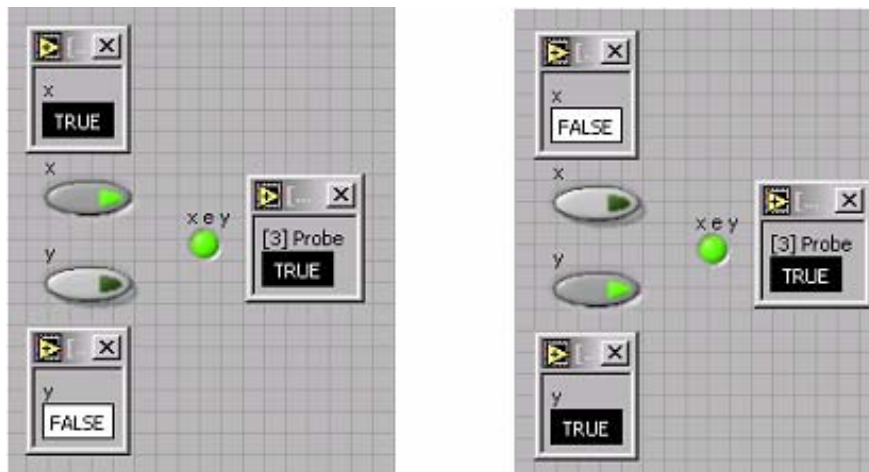


Figura 43 – Basta um valor x ou y ser verdadeiro (*TRUE*) para que o resultado seja verdadeiro (*TRUE*)

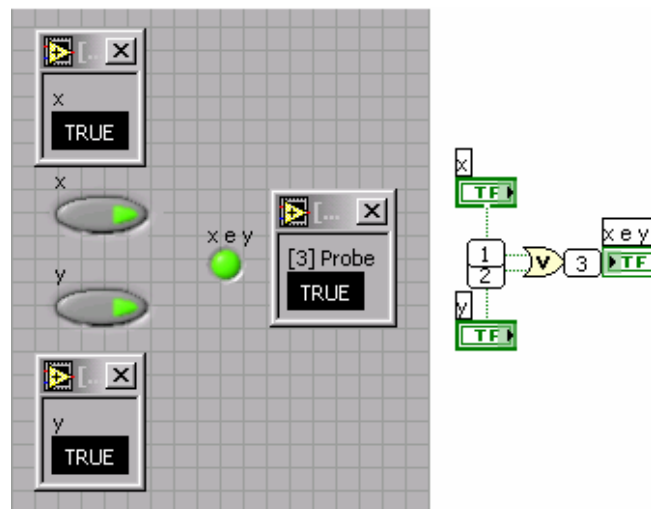


Figura 44 – Os valores x e y são verdadeiros (*TRUE*), portanto o resultado também é verdadeiro (*TRUE*). O diagrama de bloco é mostrado ao lado

4.7.2.3. Exclusive Or



Figura 45

Calcula o valor lógico exclusivo *Or* (*XOR*) entre os valores de entrada. Se os dois valores *x* e *y* são verdadeiros (*TRUE*) ou falsos (*FALSE*), o resultado será falso (*FALSE*). Caso contrário será verdadeiro (*TRUE*).

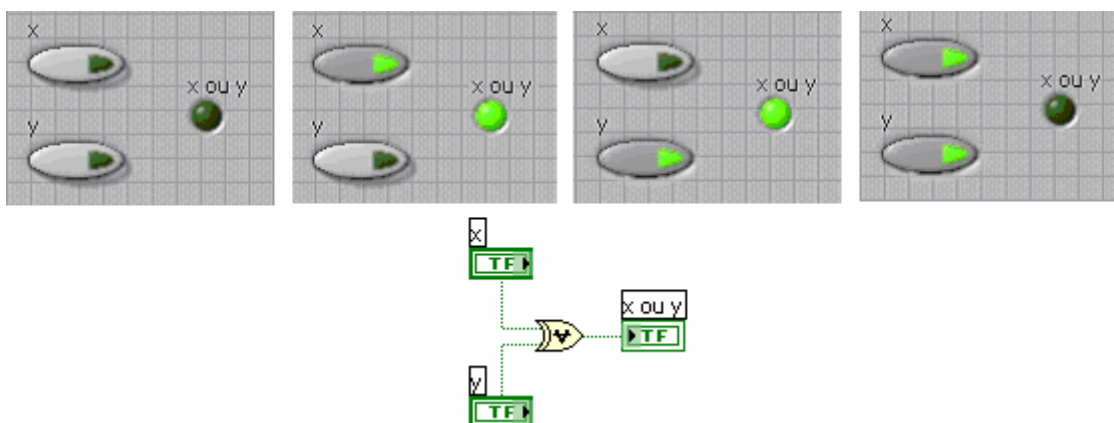


Figura 46 – Essa função é exclusiva, como é visto neste VI. Ou é *x* ou é *y*, nunca os dois, por isso, é exclusivo

4.7.3. Subpaleta String

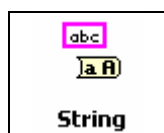


Figura 47

Use as funções *String* para concatenar dois ou mais *strings*, converter dados em *strings*, e formatar um *string* para processamento de textos.

4.7.3.1. String Length

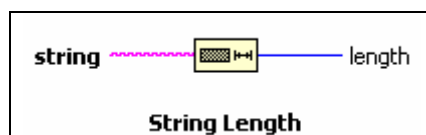


Figura 48

O *String Length* indica o número de caracteres (*bytes*) contidos num *string*.

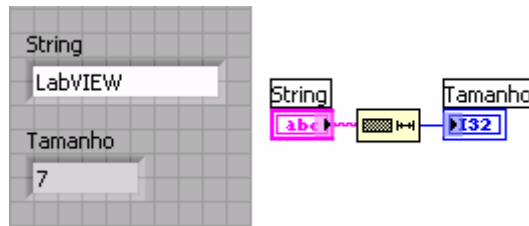


Figura 49 – Este VI conta o número de caracteres contidos na palavra *LabVIEW*

4.7.3.2. Concatenate Strings

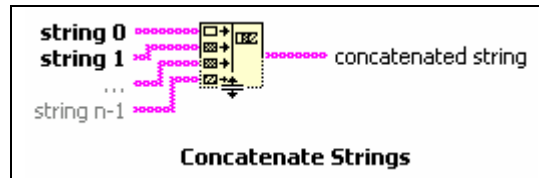


Figura 50

Esta função concatena *strings*, ou seja, une diferentes *strings* em um só.

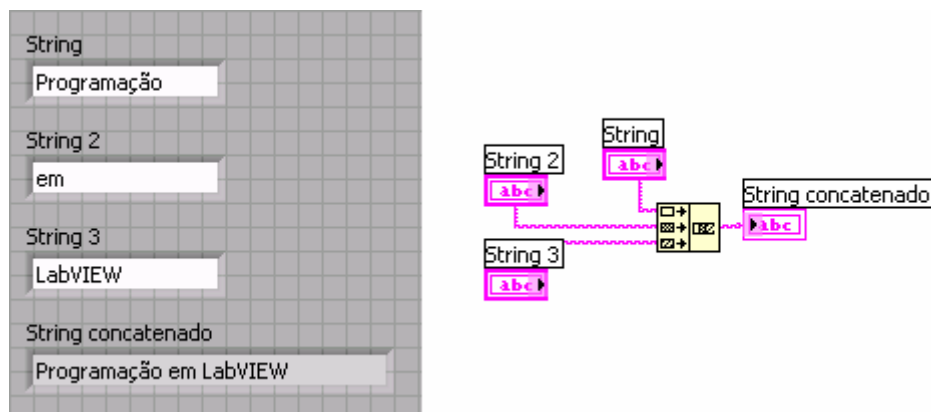


Figura 51 – Este VI une vários *strings* diferentes em um só

4.7.3.3. To Upper Case

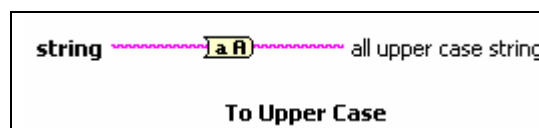


Figura 52

Esta função converte de minúsculo para maiúsculo todos os caracteres contidos no *string*. É possível realizar a conversão de maiúsculo para minúsculo utilizando a função *To Lower Case*.

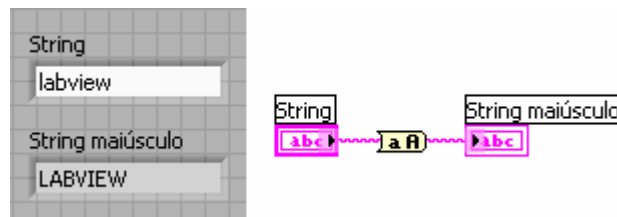


Figura 53 – Este VI transforma as letras em maiúsculo

4.7.3.4. String Subset

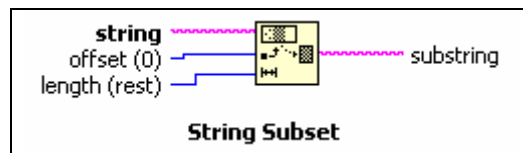


Figura 54

Esta função forma um *substring* de acordo com os valores inseridos nos controles. Deve ser indicado o número de letras que formarão o substring e com que letra deve ser iniciado, o zero deve ser incluído.

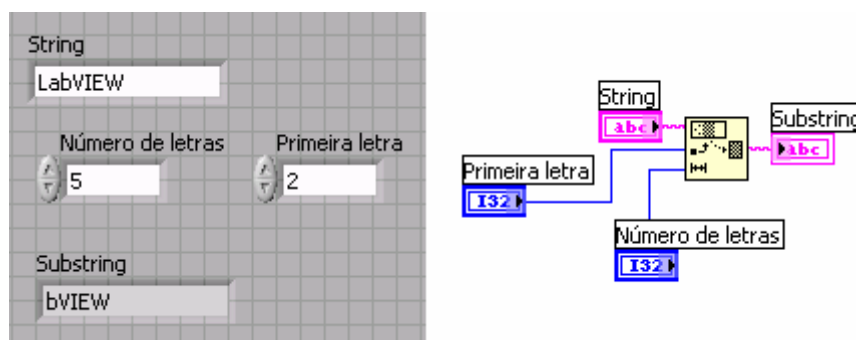


Figura 55 – Neste VI, foi inserido um *string* com sete letras (*LabVIEW*), e foi formado um *substring* com cinco letras e que começa pela letra “b”, porque o zero equivale à letra “L”, o um equivale à letra “a” e o dois equivale à letra “b”

4.7.4. Subpaleta Array

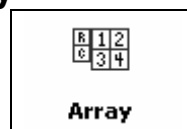


Figura 56

Um *array* é um conjunto de elementos combinados na forma de matrizes ou tabelas. Todos os seus elementos devem ter o mesmo tipo de dados.

4.7.4.1. Build Array

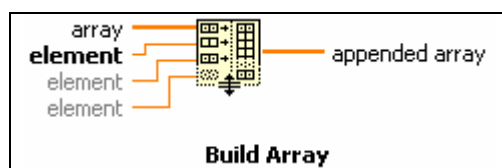


Figura 57

Esta função é usada para construir *arrays*.

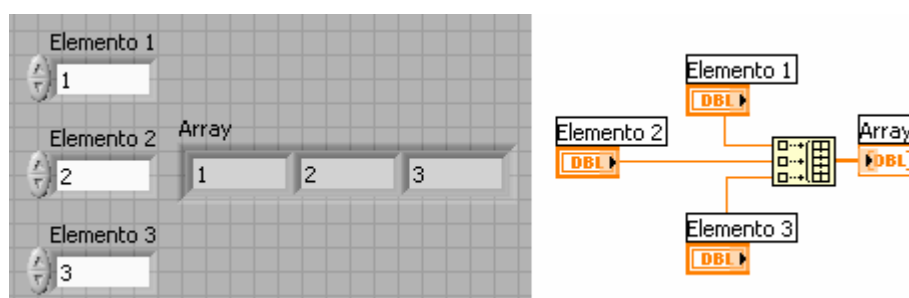


Figura 58 – Array de valores numéricos

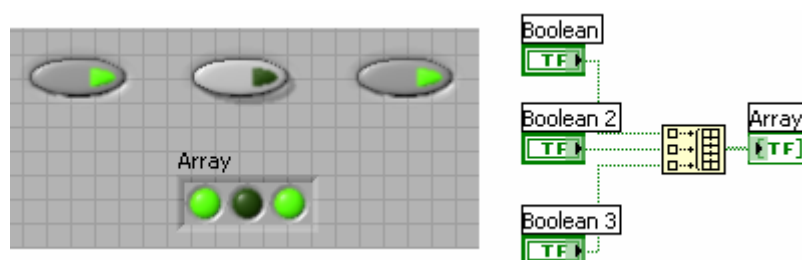


Figura 59 – Array de valores lógicos



Obs: Se forem conectados ao build array apenas elementos, de qualquer tipo (numérico, booleano, string e etc), ele só poderá formar uma linha. Portanto, para criar uma matriz com mais de uma dimensão, com linhas e colunas, é necessário criar um array de um array. Como é mostrado no exemplo da Figura 60.

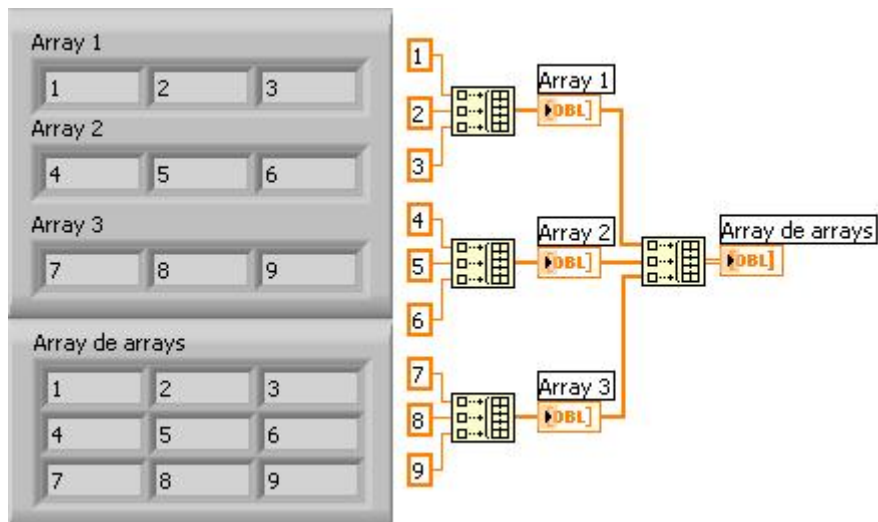


Figura 60 – Array de arrays



Obs: Pode-se fazer a soma ou o produto dos elementos de um array. As funções correspondentes, *Add Array Elements* e *Multiply Array Elements*, são encontradas na subpaleta *Numeric*.

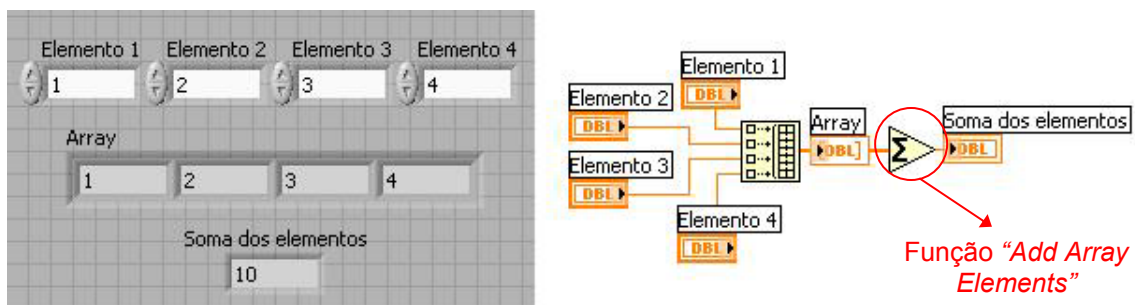


Figura 61 – Soma dos elementos de um array

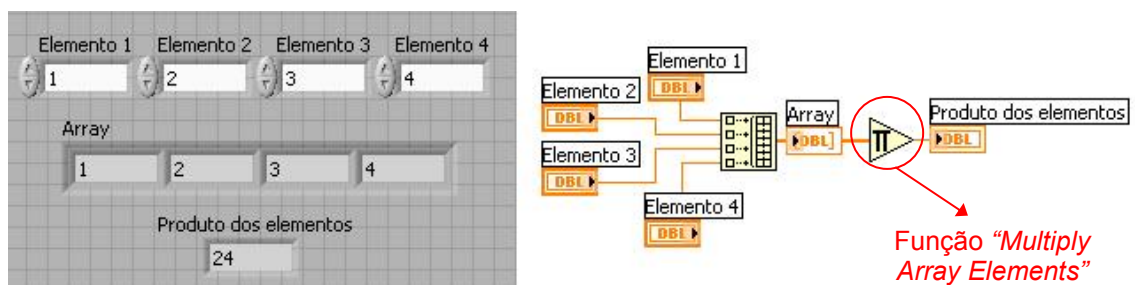


Figura 62 – Produto dos elementos de um array

4.7.4.2. Array Size

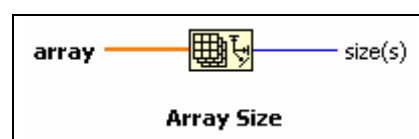


Figura 63

O *Array size* indica o número de elementos que estão formando um *array*.

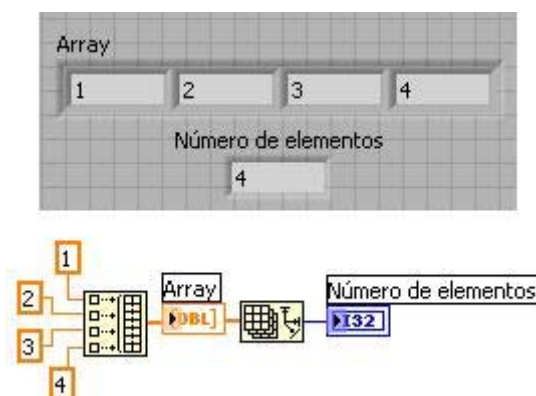


Figura 64 – Programa indicando o número de elementos

4.7.4.3. Index Array

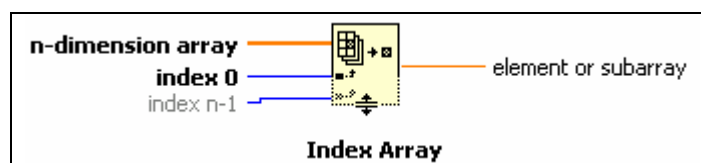


Figura 65

O *Index array* indica o elemento da matriz de acordo com os índices indicados. É importante saber que o índice começa no zero, ou seja, a primeira linha ou primeira coluna corresponde ao índice zero.

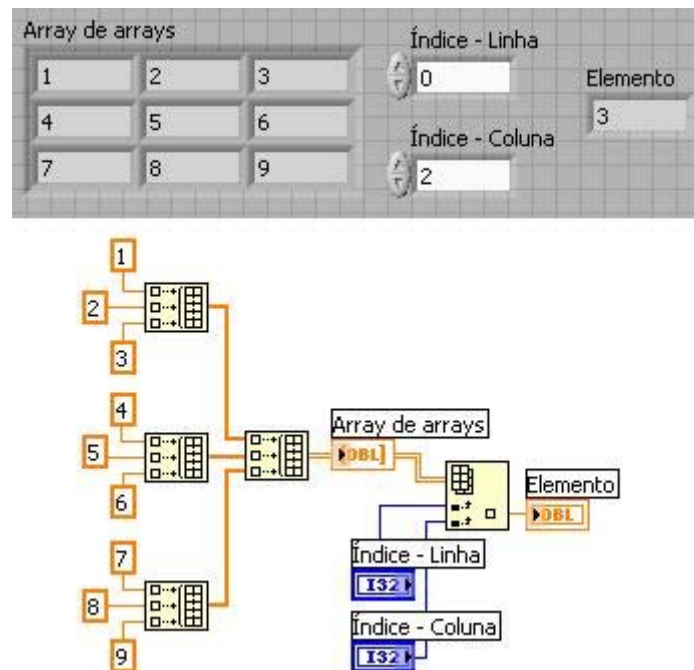


Figura 66 – O elemento da 1ª linha (índice 0) e 3ª coluna (índice 2) é o número 3

4.7.4.4. Transpose 2D Array

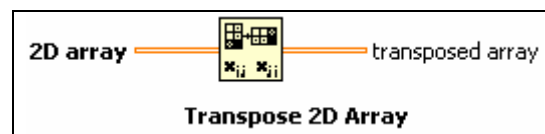


Figura 67

Esta função gera a matriz transposta de uma dada matriz. O exemplo a seguir mostra a matriz original e a sua transposta, fazendo a primeira linha ser a primeira coluna e a segunda linha ser a segunda coluna.

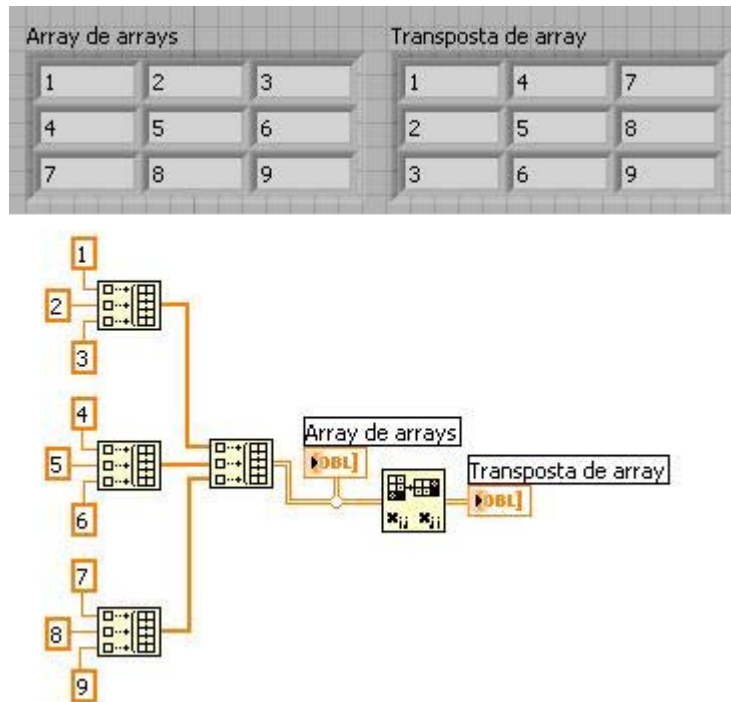


Figura 68 – Matriz original com a sua respectiva matriz transposta e o diagrama de bloco responsável por executar essa função

4.7.4.5. Array Max & Min

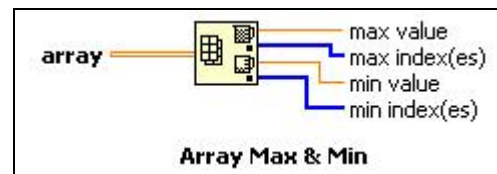


Figura 69

A função *Array Max & Min* indica o valor máximo e mínimo encontrado no *array*, com os respectivos índices da cada valor.

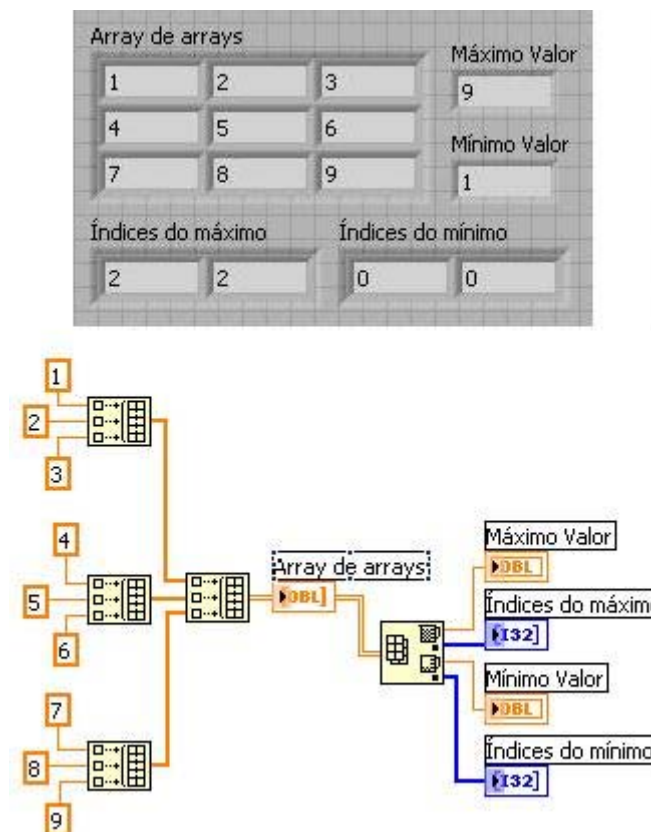


Figura 70 – o valor máximo deste *array* é o número 9 (índices 2 e 2) e o mínimo é o número 1 (índices 0 e 0)



Obs: Arrays também podem ser usados para formar tabelas de strings (cadeia de caracteres). As Figuras 71 e 72 mostram um exemplo de tabela utilizando uma “Table” encontrada na subpaleta “List & Table”, no painel frontal.

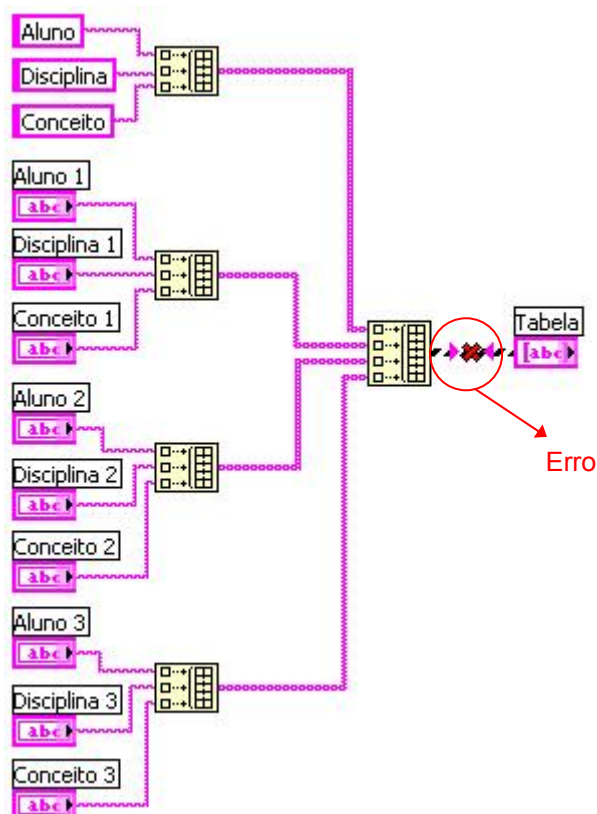


Figura 71 – A tabela utilizada é um controle e não pode ser usada para saída de dados, por isso ocorre um erro, demonstrado através do fio quebrado

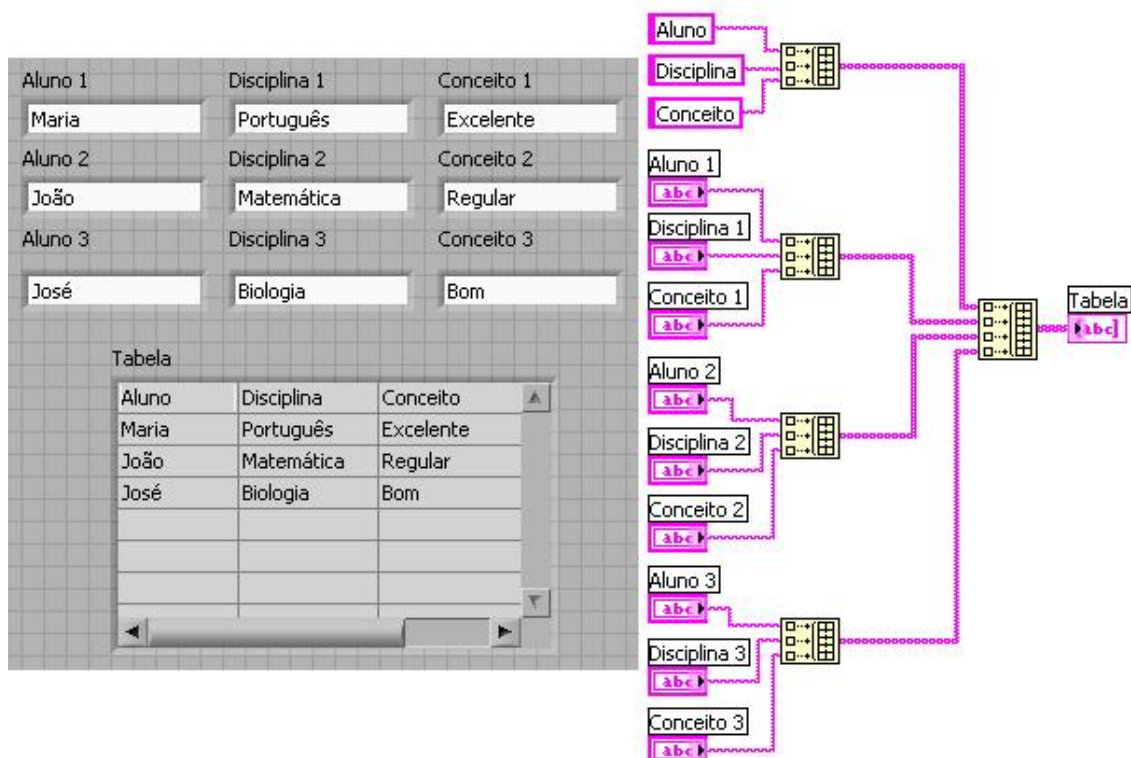


Figura 72 – O VI correto devido a alteração da Tabela de Controle para Indicador, através da seleção da opção *Change to Indicator*, no menu de atalho com o botão direito sobre o ícone

4.7.5. Subpaleta Cluster



Figura 73

A função *Cluster* agrupa ou desagrupa elementos. Em um cluster podem ser agrupados elementos de diferentes tipos de dados.

4.7.5.1. Bundle

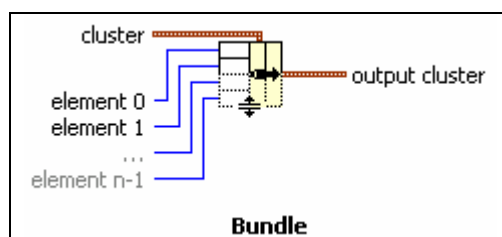


Figura 74

Esta função agrupa elementos, transformando-os em um cluster.

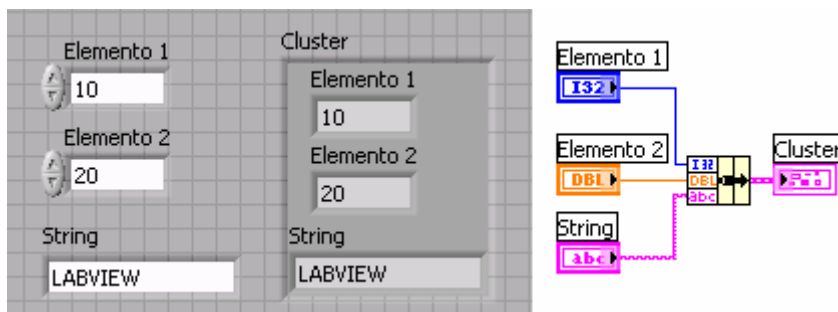


Figura 75

4.7.5.2. Unbundle

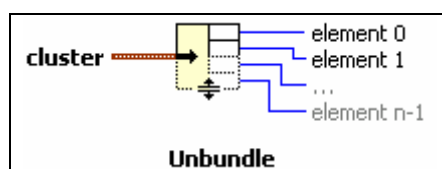


Figura 76

Esta função desagrupa o cluster separando cada elemento. A função *SI/Config.vi*, por exemplo, tem um terminal chamado de *Sound Format*, o qual é um cluster. Caso se queira manipular cada elemento deste terminal, a idéia é usar a função *Unbundle*.

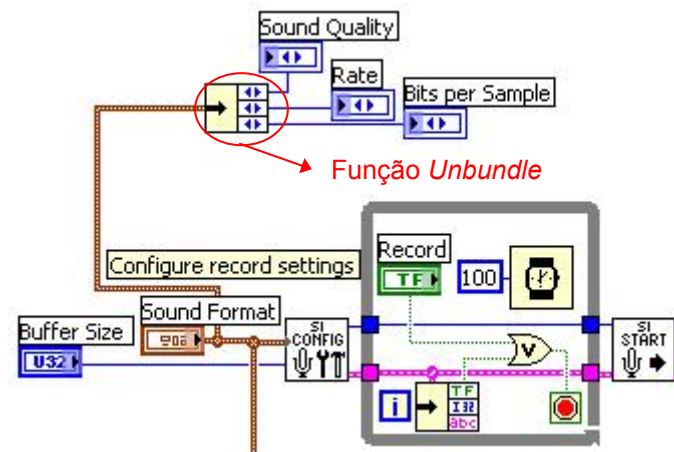


Figura 77

4.7.5.3. Build Cluster Array

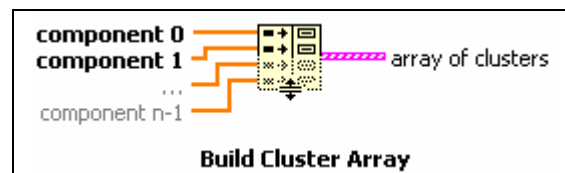


Figura 78

Esta função forma um cluster de arrays.

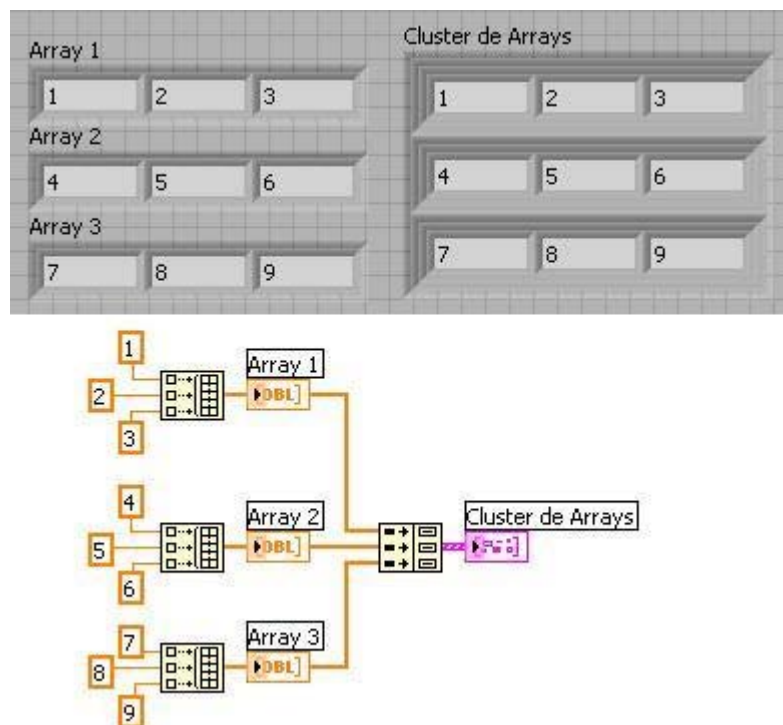


Figura 79

4.7.6. Subpaleta Comparison

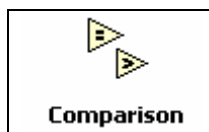


Figura 80

Estas funções fazem comparações entre valores *booleanos*, *strings*, numéricos, *arrays* e *clusters*.

4.7.6.1. Greater?

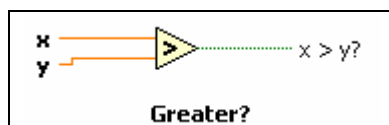
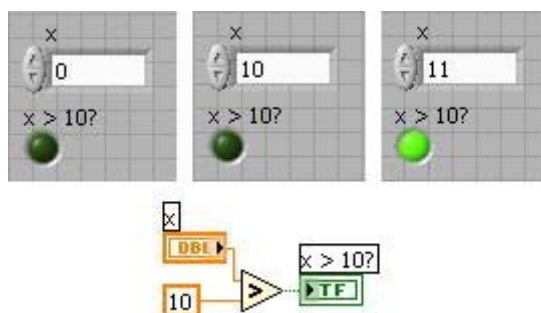


Figura 81

Esta função faz uma comparação entre os valores de entrada **x** e **y**. Se **x** for maior que **y**, o LED acende, caso não seja o LED permanece apagado.

Figura 82 – VI que compara se o valor de **x** é maior que o número escolhido, que é 10

4.7.6.2. Greater Than 0?



Figura 83

Esta função faz uma comparação entre o valor de entrada **x** e o número 0. Se **x** for maior que 0, o LED acende, caso não seja o LED permanece apagado.

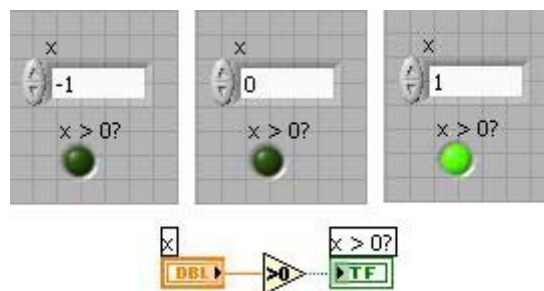


Figura 84 – VI que compara se o valor de x é maior que 0

4.7.6.3. Select

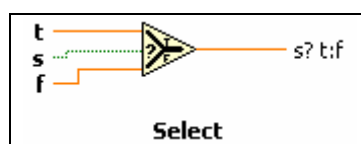


Figura 85

A função *Select* mostra o valor conectado à entrada **t** (**true – verdadeiro**) ou a entrada **f** (**false – falso**), dependendo do valor de **s**. Se **s** é verdadeiro, esta função mostra o valor conectado a **t**. Se **s** é falso, esta função mostra o valor conectado a **f**.

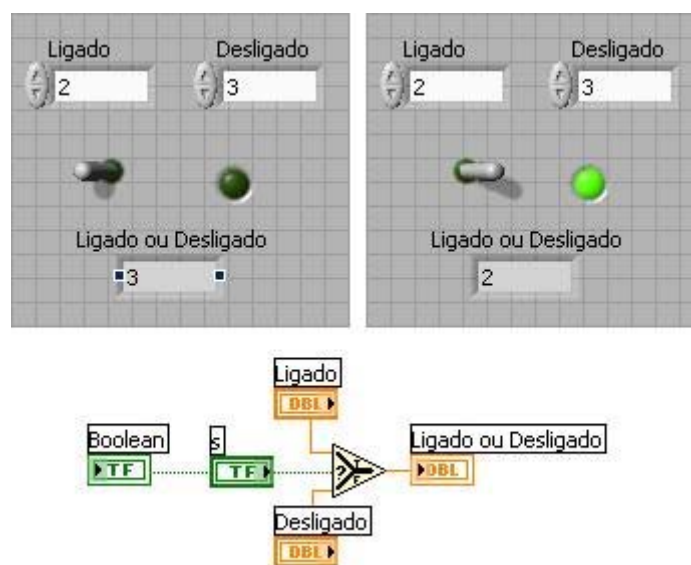


Figura 86 – Quando o botão está desligado, ele é FALSO. Quando ele está ligado, ele é VERDADEIRO

4.7.6.4. Max & Min



Figura 87

Esta função compara os valores de entrada **x** e **y**, e indica o maior e o menor valor.

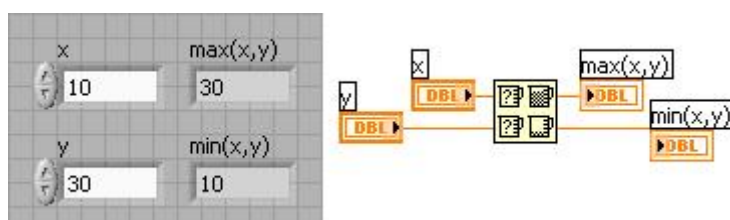


Figura 88 – VI que indica o valor máximo e mínimo

4.7.7. Subpaleta Time & Dialog



Figura 89

Os VI's de tempo, diálogo e erro podem ser usados para manipular a velocidade com que uma operação é executada, para criar caixas de diálogo, para indicar algo quando uma operação for executada.

4.7.7.1. Wait (ms)

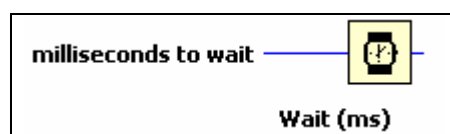


Figura 90

Basta indicar, através de uma constante, em quanto tempo se deseja atrasar a execução de uma tarefa, em milissegundos.

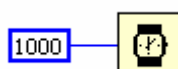


Figura 91 – Foi usada uma constante com valor igual a 1000, ou seja, a tarefa será atrasada em 1000 milissegundos (1 segundo)

4.7.7.2. One Button Dialog

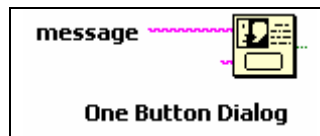


Figura 92

Esta função emite uma mensagem, de acordo com a constante *string*.

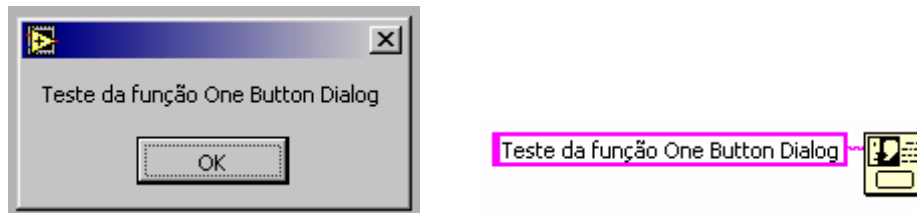


Figura 93 – Mensagem que é mostrada quando o programa é executado



Obs: Ao usar esta função é indicado incluir no VI a função Wait com um tempo de atraso da operação de 3000 milissegundos, para que se tenha tempo suficiente de clicar no botão OK da mensagem.

4.7.7.3. Two Button Dialog



Figura 94

Esta função gera uma mensagem com botões, que resultam um valor booleano, julgando verdadeiro ou falso.

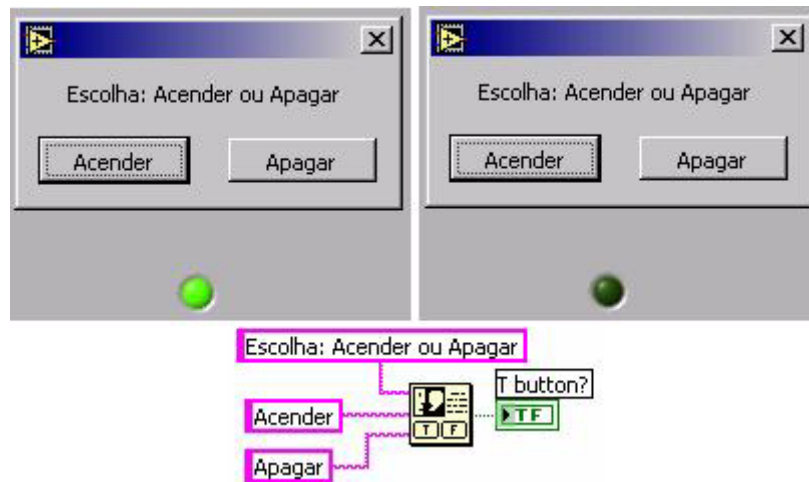


Figura 95 – Acende ou apaga o LED

Outra idéia é trocar as cores, por exemplo, Verde ou Vermelho.



Figura 96 – Cor verde ou vermelha

Atividades

Tente fazer as seguintes atividades:

- VI que execute as quatro operações (soma, subtração, multiplicação e divisão) simultaneamente;
- VI que multiplique quatro números simultaneamente;
- VI que faça conversão de temperaturas, usando as relações:
 - $T[^\circ\text{F}] = T[^\circ\text{C}] * 1,8 + 32$
 - $T[\text{K}] = T[^\circ\text{C}] + 273$



Obs: As respostas estão no final da apostila.

5. Gráficos

Os gráficos podem ser usados para mostrar sinais gerados no *LabVIEW* ou aquisitados de um equipamento real. Existem vários tipos de gráficos, os quais são encontrados no painel frontal, *Controls Palette*>>*Graph*. A Figura 97 mostra as principais partes de um gráfico.

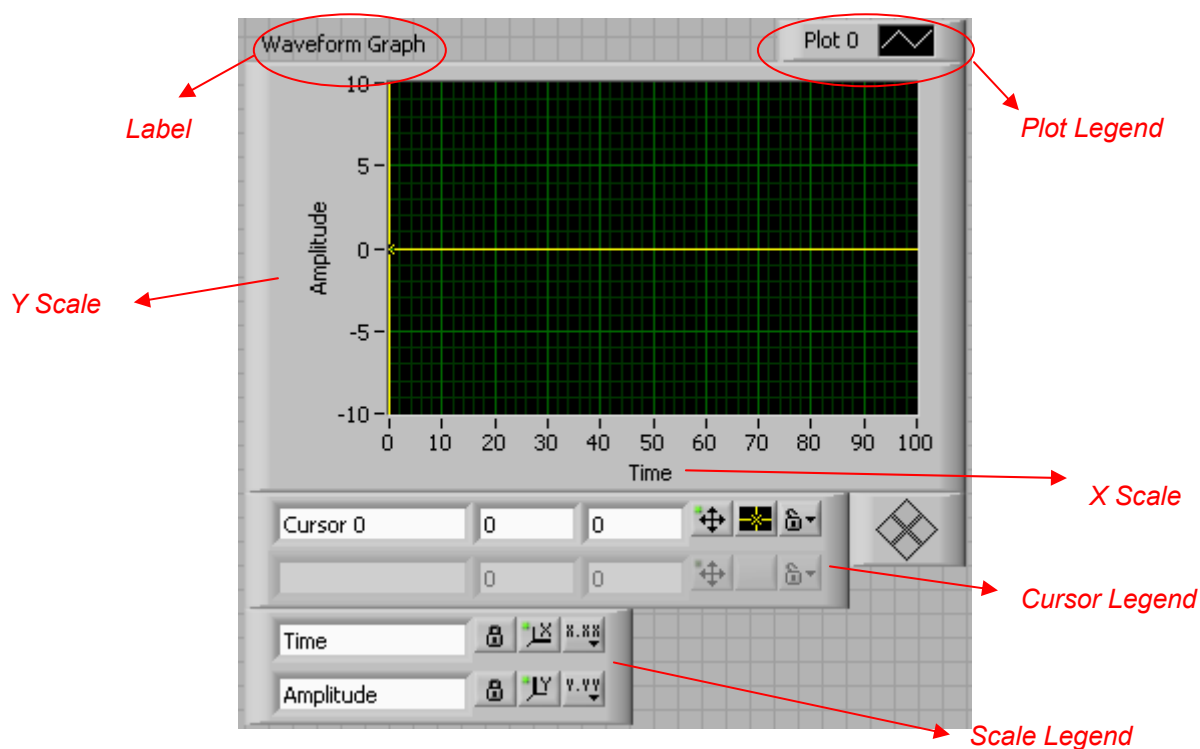


Figura 97

Todos esses componentes podem ficar visíveis ou não. A configuração pode ser feita clicando-se com o botão direito do mouse sobre o gráfico. No menu de atalho aparecem várias opções de configuração, como mostrado na Figura 98.

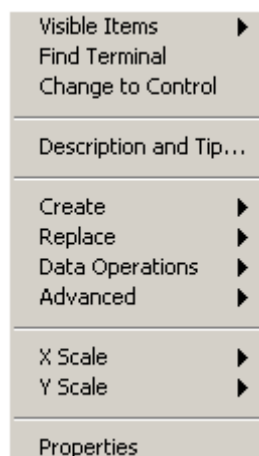


Figura 98 – Menu de atalho de gráficos.

- **Visible Items** – Nesta opção, pode-se escolher os itens que devem permanecer visíveis, tais como, *Label*, *Plot Legend*, *Scale Legend*, *Cursos Legend*, *X Scale*, *Y Scale*, entre outros.
- **Find Terminal** – Esta opção localiza o terminal correspondente no diagrama de bloco.
- **Change to Control** – Esta opção transforma o gráfico para um controle, que inicialmente é um indicador.
- **Description and Tip...** – Abre uma janela na qual se insere comentários, isto é, a descrição e o tipo do gráfico. Quando o VI é executado, pode-se ver esses comentários clicando com o botão direito do mouse sobre o gráfico.
- **Create** – Cria uma *Local Variable*, *Property Node*, *Reference* e *Invoke Node*.
- **Replace** – Mostra a paleta de controles para que se possa substituir o gráfico atual por outro.
- **Data Operations** – Executa operações referentes aos dados do gráfico, tais como copiar e colar os dados, limpar o gráfico.
- **Advanced** – Neste item é possível ocultar o objeto, clicando em *Hide Indicator*.
- **X Scale** – Nesta opção configura-se a escala de X como auto-escala ou não.
- **Y Scale** – Nesta opção configura-se a escala de Y como auto-escala ou não.

- **Properties** – As configurações de gráfico podem ser feitas também na janela *Properties*, mostrada na Figura 99.

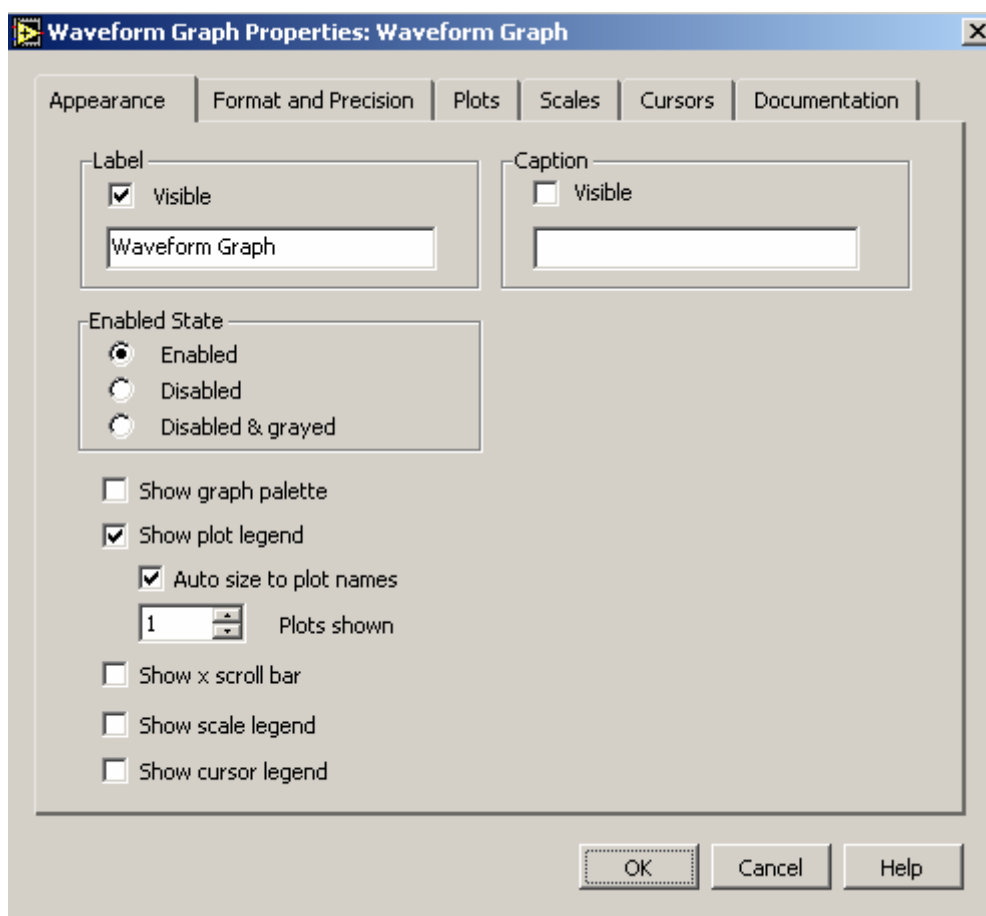


Figura 99

5.1. Waveform Graph

A Figura 100 mostra um VI que utiliza o *Waveform Graph*. Neste exemplo foi simulado um sinal que apresenta uma forma de onda senoidal (seno). A função que “gera o seno” é a *Sine Waveform* e encontra-se no diagrama de blocos. Nela podem ser configuradas a frequência, a amplitude e a fase do sinal, entre outras.

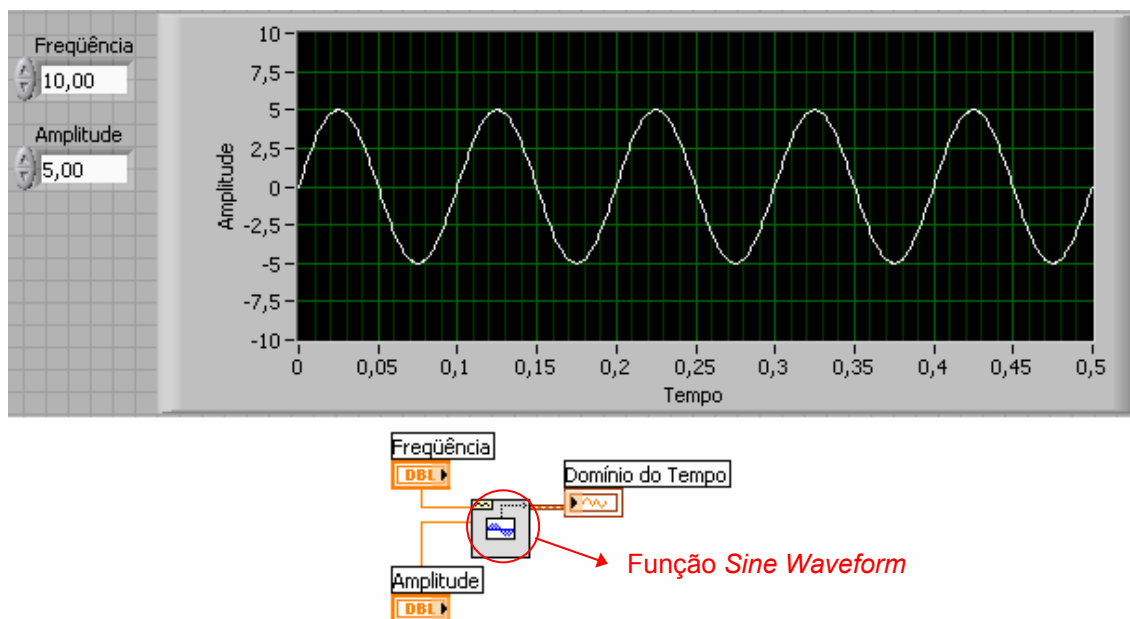


Figura 100 – Exemplo de Waveform Graph.



Obs: Há outras formas de onda no LabVIEW, a onda quadrada, dente de serra e onda triangular.

É possível visualizar dois sinais no mesmo gráfico. Um exemplo é gerar duas formas de onda defasadas de 90° . Para isso é necessário criar dois sinais, depois conectá-los a um *Build Array* e por fim conectar o *array* criado a um gráfico. A Figura 101 mostra o VI.

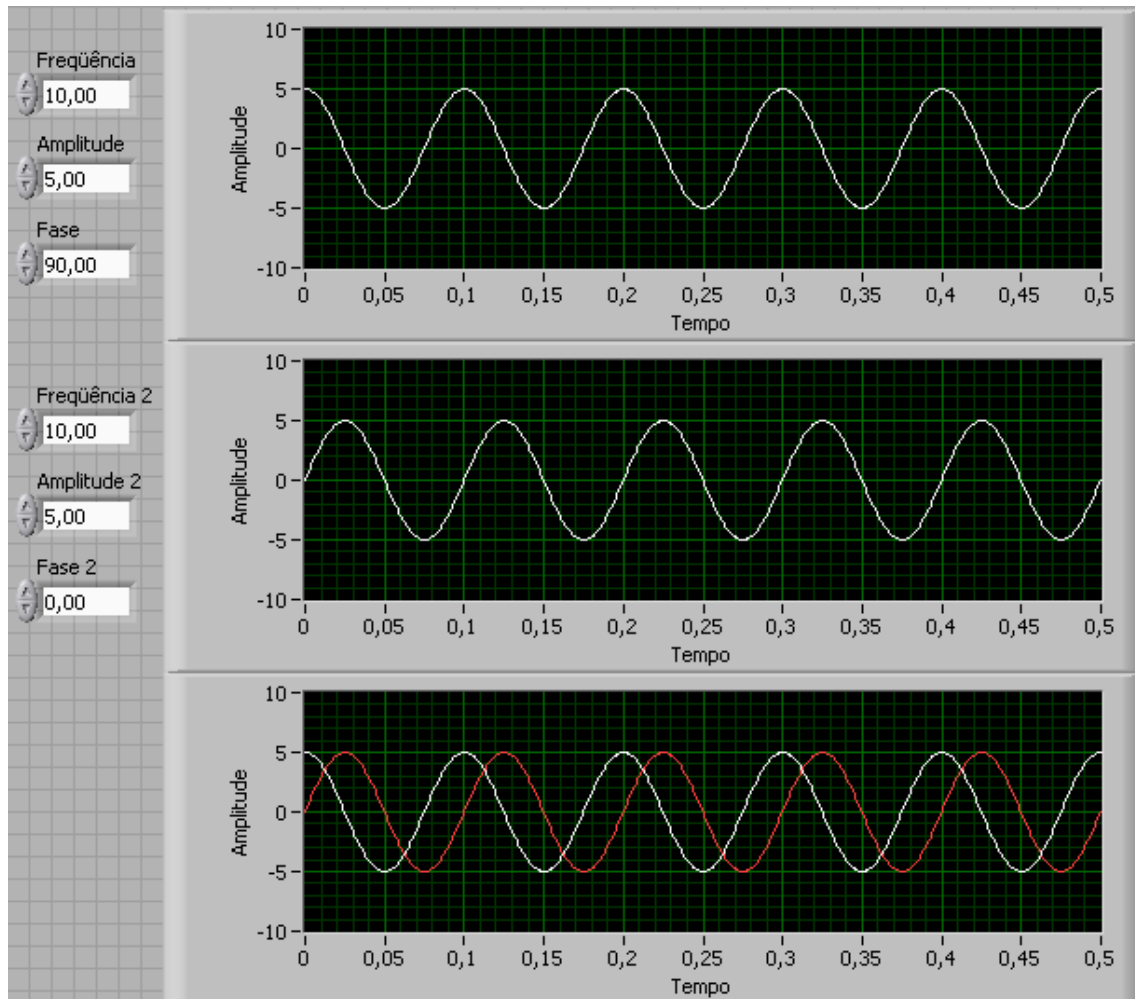


Figura 101 – VI de senos defasados de 90°

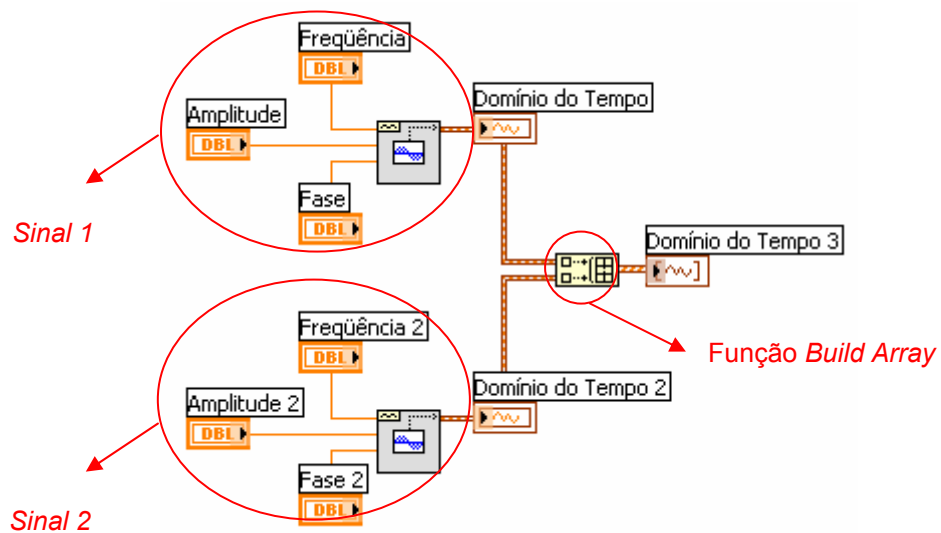


Figura 102 – Diagrama de bloco

Neste caso, em que se têm dois sinais no mesmo gráfico, é interessante manter visível o *Plot Legend*, pois ele identifica cada sinal. Basta redimensioná-lo para o número de sinais e editar os nomes originais *Plot 0* e *Plot 1*.



Obs: Não são obrigatórios os dois gráficos que mostram os sinais isolados. Estes foram inseridos somente por uma questão didática, para que se visualize cada sinal isoladamente.

5.2. XY Graph

O *XY Graph* gera um gráfico que relaciona os valores de X com os valores de Y, ou seja, cada ponto pertencente à curva contém uma coordenada em X e uma em Y. É obrigatório o uso de um cluster do tipo *Bundle*, com duas entradas, sendo que cada uma deve ser um *array*. A Figura 103 mostra um exemplo de *XY Graph*.

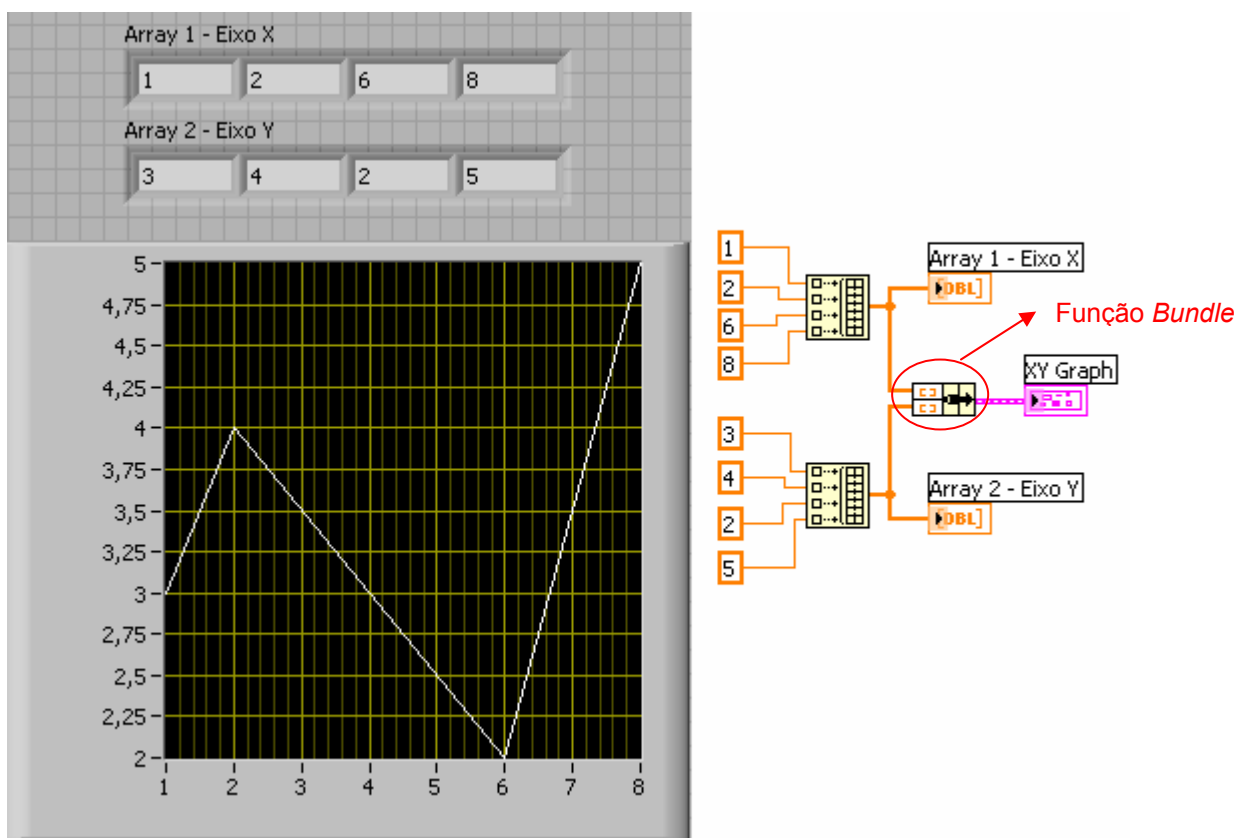


Figura 103

6. Programação Estruturada

6.1. Laços (Loops)

Laços são construções que permitem executar uma seqüência de declarações mais de uma vez. Existem duas formas básicas de construção de laços: *While Loop* e *For Loop*. A principal diferença entre esses dois tipos de laços é como a repetição é controlada. O código em laço *while* é repetido uma quantidade indefinida de vezes até que alguma condição especificada pelo usuário seja satisfeita. Em contraste, o código em laço *for* é repetido uma quantidade determinada de vezes, e o número de repetições é conhecido antes do início do laço.

6.1.1. While Loop

É usado quando se deseja que uma operação se repita enquanto uma determinada condição é verdadeira.

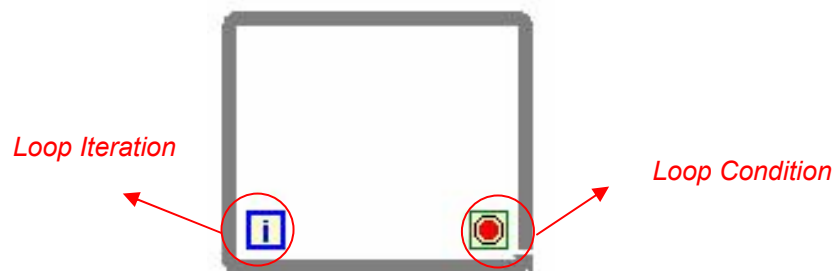


Figura 104 – While Loop

Contém dois terminais:

- **Loop Condition:** A ele é conectado um botão de valor *booleano*. O *LabVIEW* confere o estado deste terminal ao término de cada iteração, se o seu valor for *true* (verdadeiro), ele para a execução do programa.
- **Loop Iteration:** Indica o número de vezes que o subdiagrama foi executado e o valor mínimo, sempre será zero.

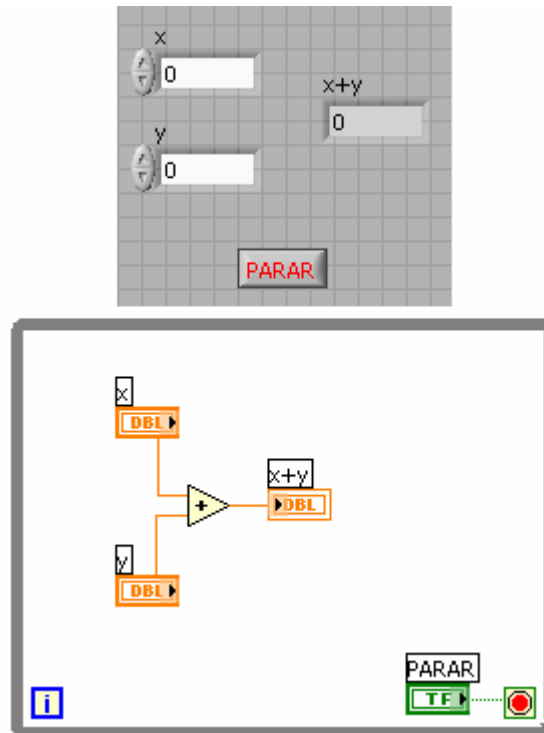


Figura 105 – O *While Loop* faz com que o programa seja executado infinitas vezes, até que o botão conectado ao terminal condicional seja acionado.

6.1.2.For Loop

É usado quando se deseja que uma operação se repita enquanto um determinado número de vezes.

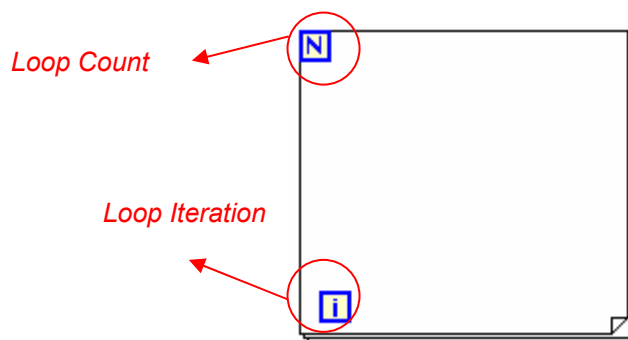


Figura 106 – For Loop

Contém dois terminais:

- **Loop Count:** Contém o número de vezes que deve ser executado o subdiagrama criado dentro da estrutura. O valor do contador se fixará externamente.
- **Loop Iteration:** Indica o número de vezes que a estrutura foi executada: 0 durante a primeira iteração, e 1 durante a segunda e assim até N-1.

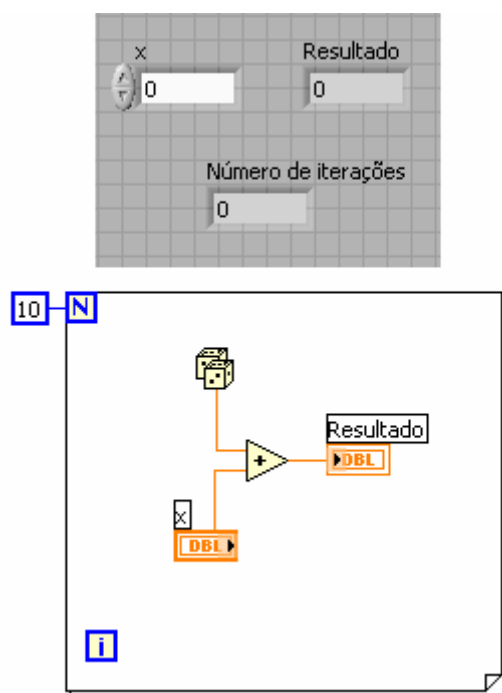


Figura 107 – O For Loop executa o programa o número de vezes indicado no terminal do contador e pára.



Obs: Quando usar While Loop ou For Loop, deve-se executar o programa clicando no botão Run, e não mais no botão Run Continuously, como mostra a Figura 108.



Figura 108

6.2. Estrutura de Decisão

6.2.1. Case Structure

É similar ao *if, then, else*, das linguagens de programação baseadas em texto e possui dois ou mais subdiagramas ou casos. Somente um subdiagrama é visível no diagrama de bloco de um VI, e uma estrutura executa um caso de cada vez.

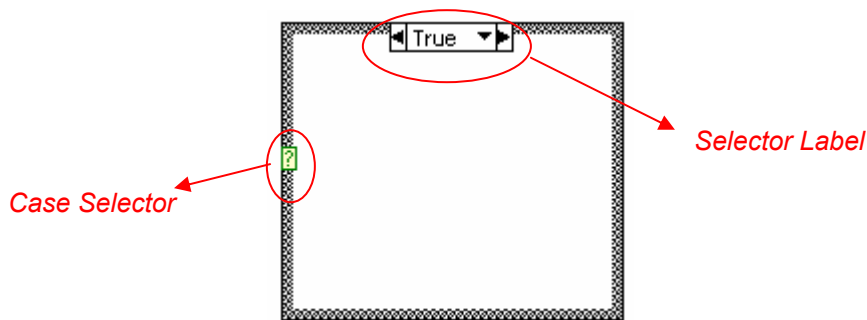


Figura 109 – Case Structure

Contém dois terminais:

- **Selector Label:** Contém os casos a serem executados de acordo com a condição conectada ao *Case Selector*.
- **Case Selector:** Nele é conectada uma condição.

O exemplo mostrado a seguir possui apenas dois casos, *True* ou *False* (ver Figura 110) e tem como objetivo calcular a raiz quadrada de um valor inserido pelo usuário. Caso este número seja maior ou igual a zero, a raiz quadrada é calculada, no entanto, se este número for menor que zero, a raiz quadrada não é calculada e uma mensagem é mostrada alertando que não existe raiz real para esse valor.

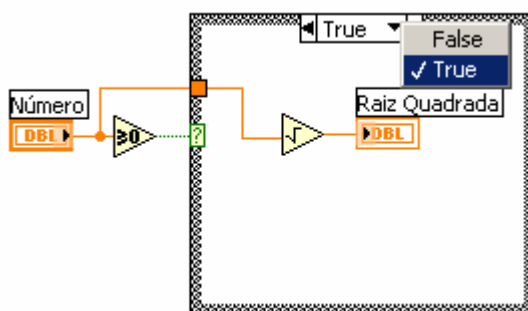


Figura 110 – Case Structure com apenas dois casos

Para isso conectou-se um controle à função “*Greater Or Equal To 0?*”, a qual foi conectada ao *Case Selector* do *Case Structure*, para que a condição seja analisada (ver Figura 111). Caso seja verdade (*true*) a raiz quadrada é calculada e mostrada no indicador “Raiz Quadrada” (ver Figura 112).

Se for falso (*false*), a raiz quadrada não é calculada. A mensagem mostrada corresponde à função *One Button Dialog*, e foi conectada a uma

constante de tipo de dado *String*. Usou-se a função *Wait (ms)* conectada a uma constante de valor 5000, para atrasar a execução desse subdiagrama em 5000 milissegundos (ver Figuras 113 e 114).

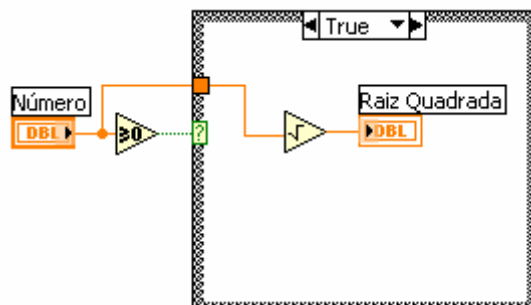


Figura 111

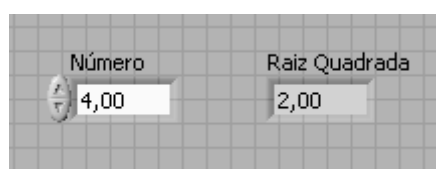


Figura 112

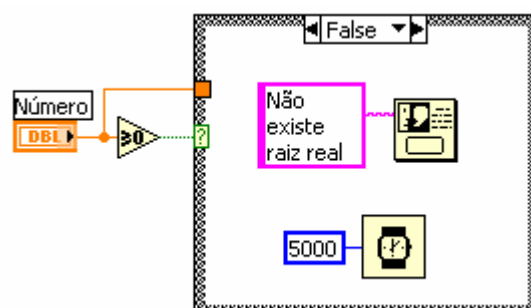


Figura 113

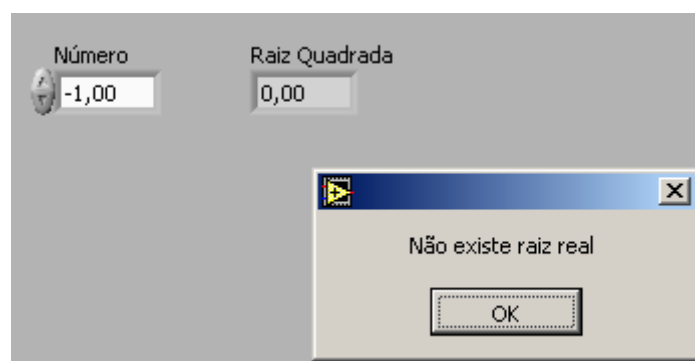
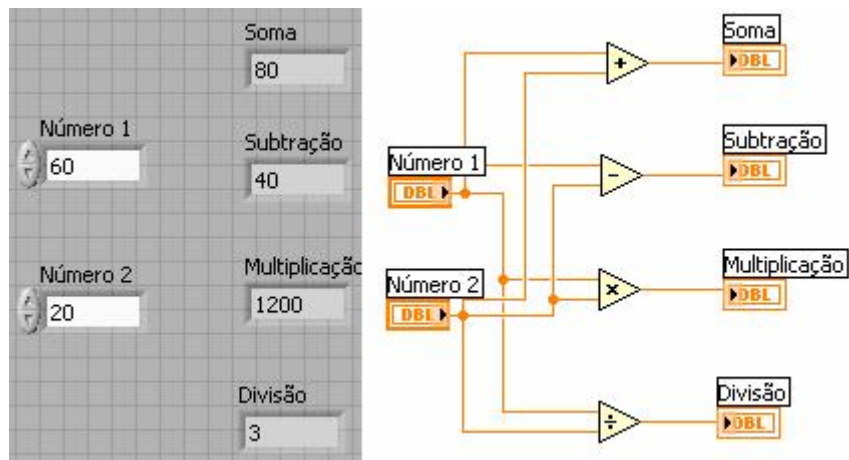


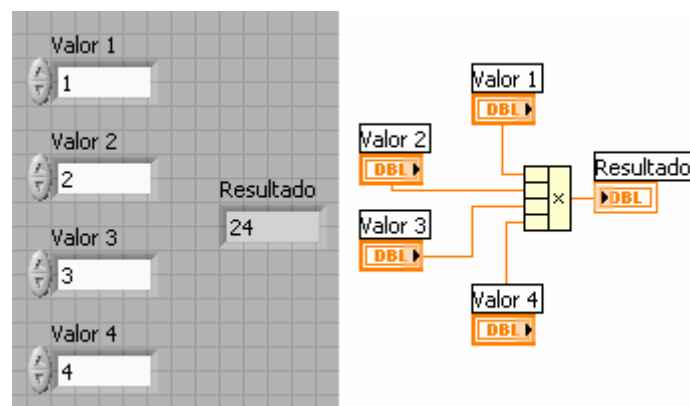
Figura 114

Respostas

- VI que execute as quatro operações (soma, subtração, multiplicação e divisão) simultaneamente.



- VI que multiplique quatro números simultaneamente.



- VI que faça conversão de temperaturas, usando as relações:
 - $T[^\circ\text{F}] = T[^\circ\text{C}] * 1,8 + 32$
 - $T[\text{K}] = T[^\circ\text{C}] + 273$

