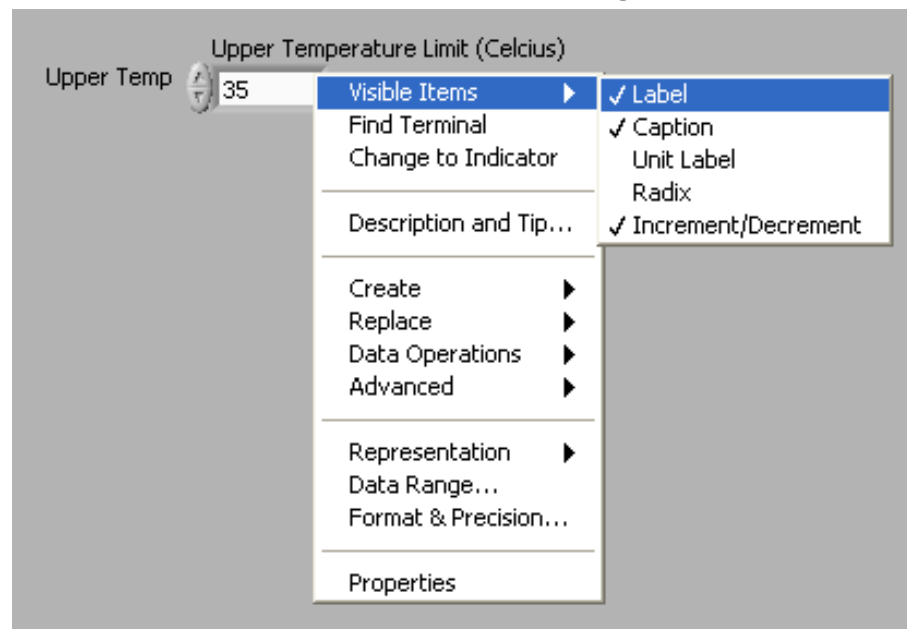


A. Projeto do Painel Frontal

- As entradas e saídas conduzem o projeto do painel frontal
- Obtenha as entradas das seguintes maneiras:
 - Adquirindo de um dispositivo
 - Lendo diretamente de um arquivo
 - Manipulando Controles
- Gere as saídas das seguintes maneiras:
 - Exibindo com indicadores
 - Armazenando em arquivo
 - Gerando dados para um dispositivo

A. Projeto do Painel Frontal– Labels/Captions

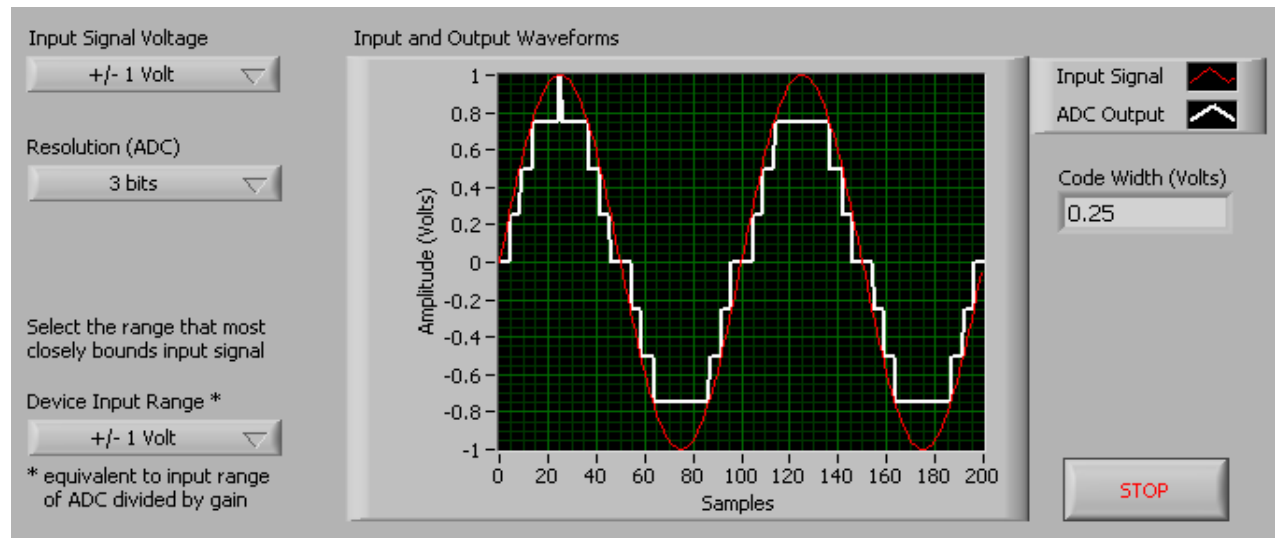
- Labels – Descrições Curtas
- Captions – Descrições Longas
- Captions não aparecem no diagrama de blocos



A. Projeto do Painel Frontal – Dicas de Cores

Comece com um esquema cinza

- Selecione um ou dois tons de cinza
- Adicione moderadamente cores de destaque para configurações importantes – em gráficos, botões de parada, etc



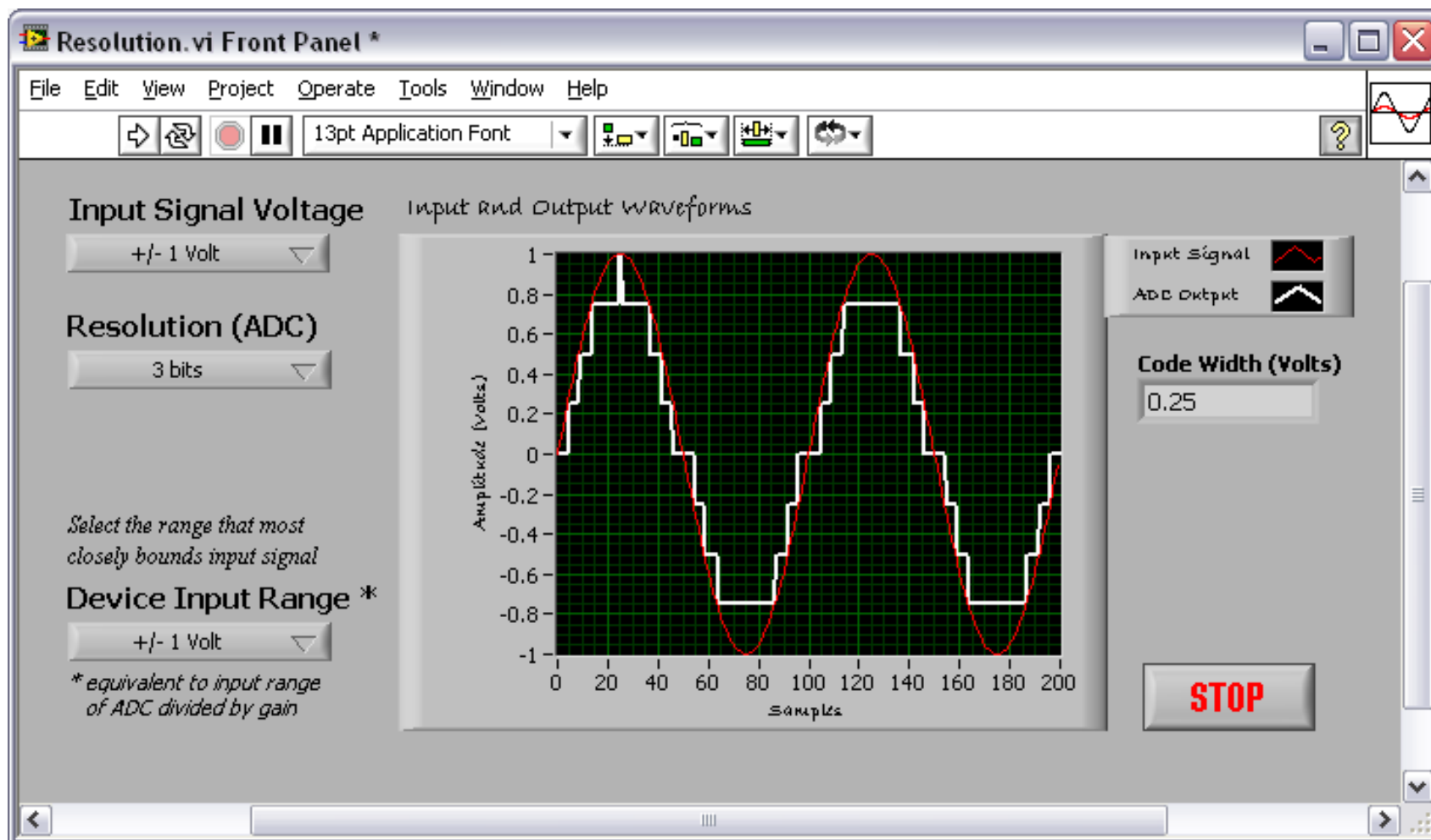
A. Projeto do Painel Frontal – Espaçamento & Alinhamento

| | |
|-----------------------|--------|
| New VI | Ctrl+N |
| New... | |
| Open... | Ctrl+O |
| Close | Ctrl+W |
| Close All | |
| Save | Ctrl+S |
| Save As... | |
| Save All | |
| Save with Options... | |
| Revert... | |
| Page Setup... | |
| Print... | |
| Print Window... | Ctrl+P |
| VI Properties... | Ctrl+I |
| Recently Opened Files | ► |
| Exit | Ctrl+Q |

| | |
|------------------------------|--------|
| New VI | Ctrl+N |
| <u>New...</u> | |
| <u>O</u> pen... | Ctrl+O |
| <u>C</u> lose | Ctrl+W |
| Close All | |
| <u>S</u> ave | Ctrl+S |
| Save <u>A</u> s... | |
| Save All | |
| Save <u>w</u> ith Options... | |
| <u>R</u> evert... | |
| Page Setup... | |
| Print... | |
| <u>P</u> rint Window... | Ctrl+P |
| VI Properties... | Ctrl+I |
| Recently Opened Files | ► |
| <u>E</u> xit | Ctrl+Q |

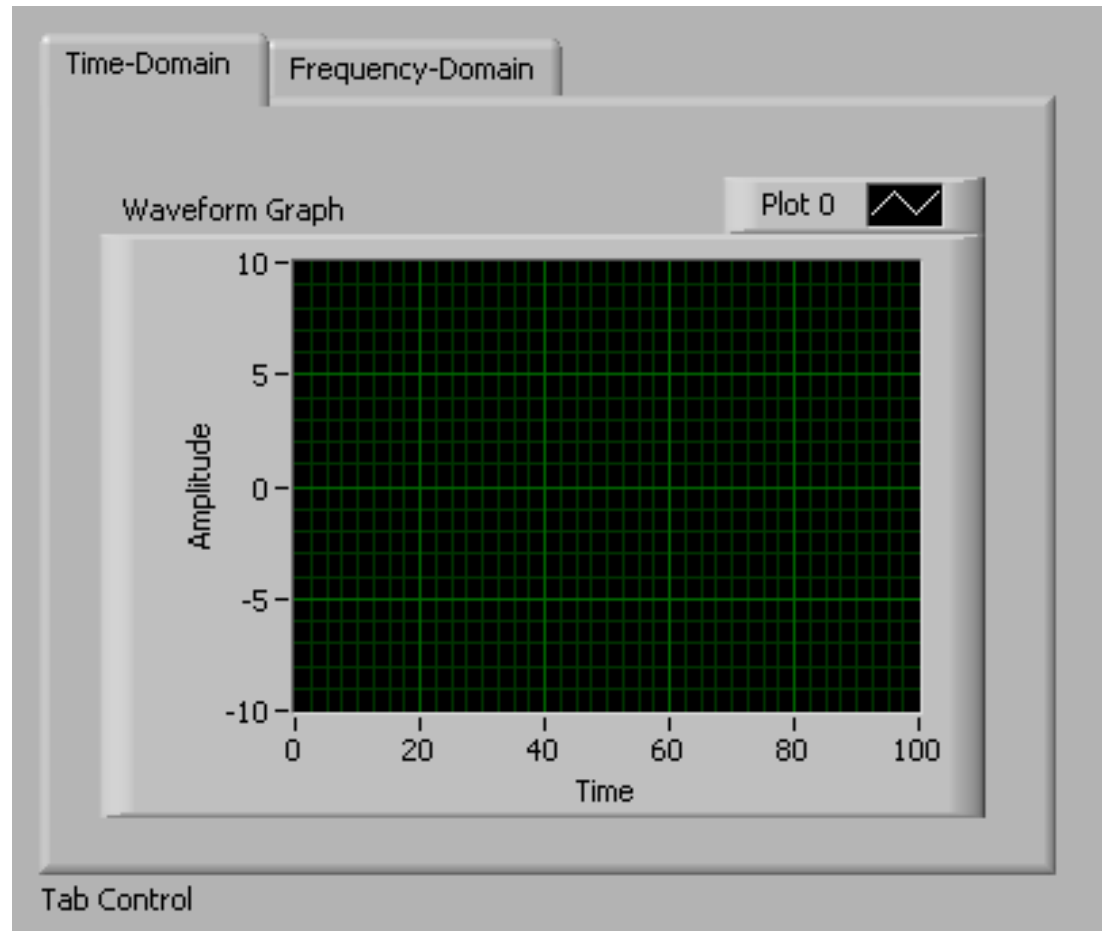
Projeto do Painel Frontal – Texto & Fontes

Um mal exemplo:



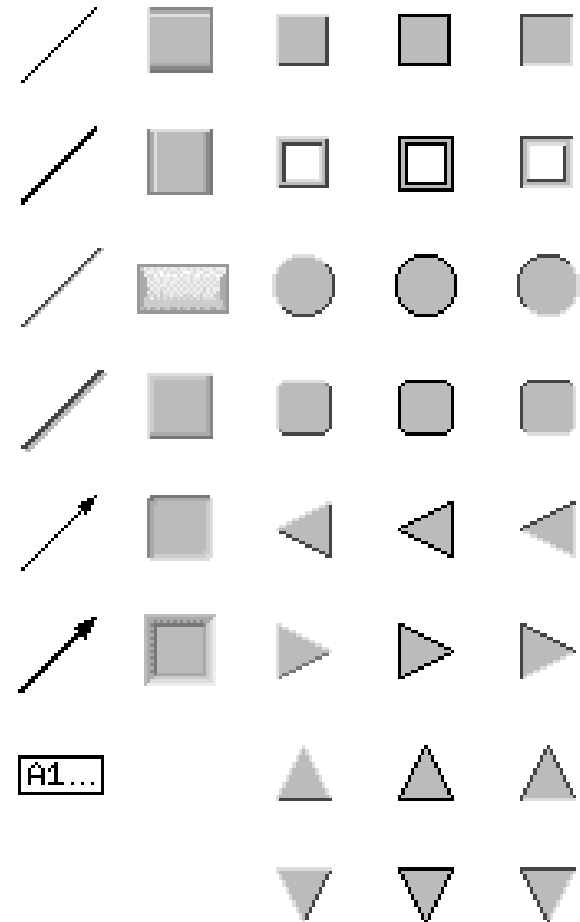
A. Projeto do Painel Frontal – Tab Controls

Use tab controls para sobrepor controles e indicadores do painel frontal em uma área menor



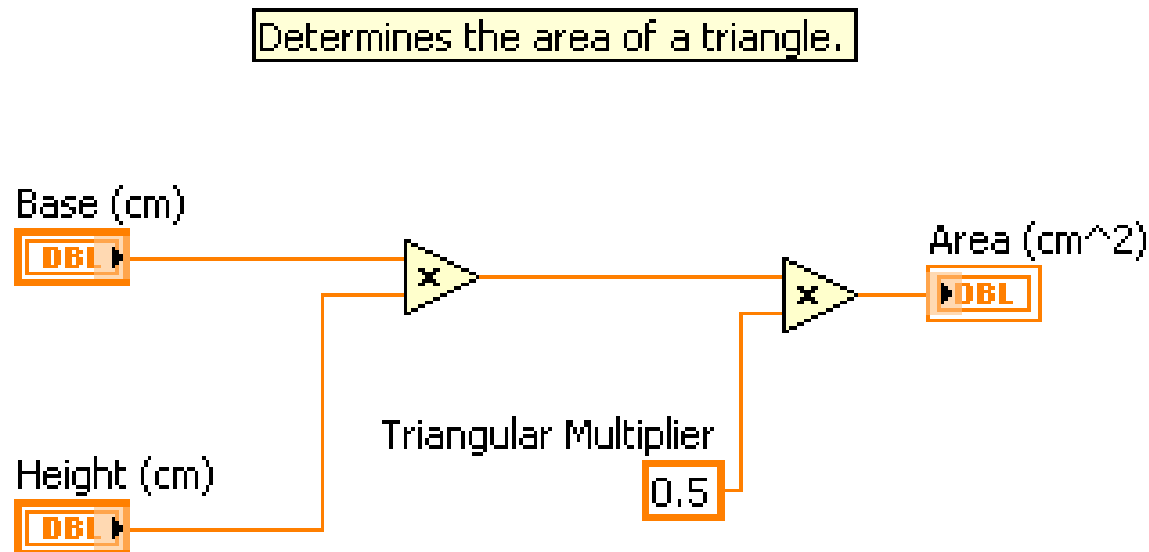
A. Projeto do Painel Frontal – Decorações

- Use decorações para agrupar ou separar visualmente objetos no painel frontal, com caixas, linhas e flechas
- Estes objetos são somente para decoração e não exibem dados



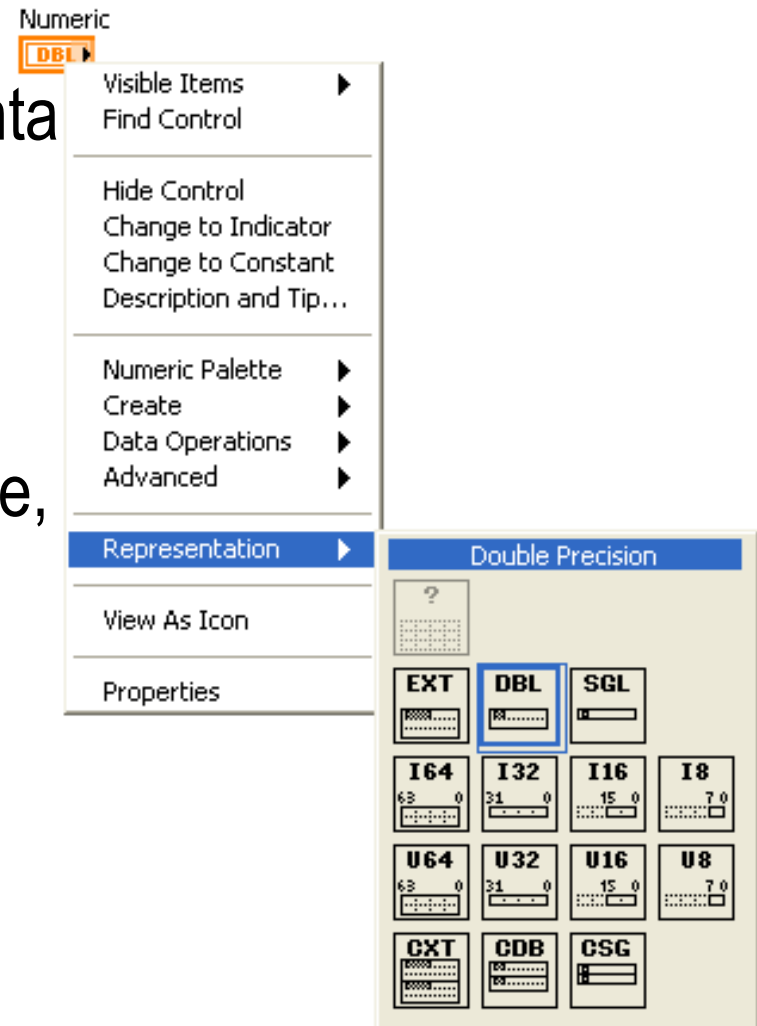
B. Tipo de Dados – Terminais

Os Terminais visualizam as informações da comunicação sobre os tipos de dados representados



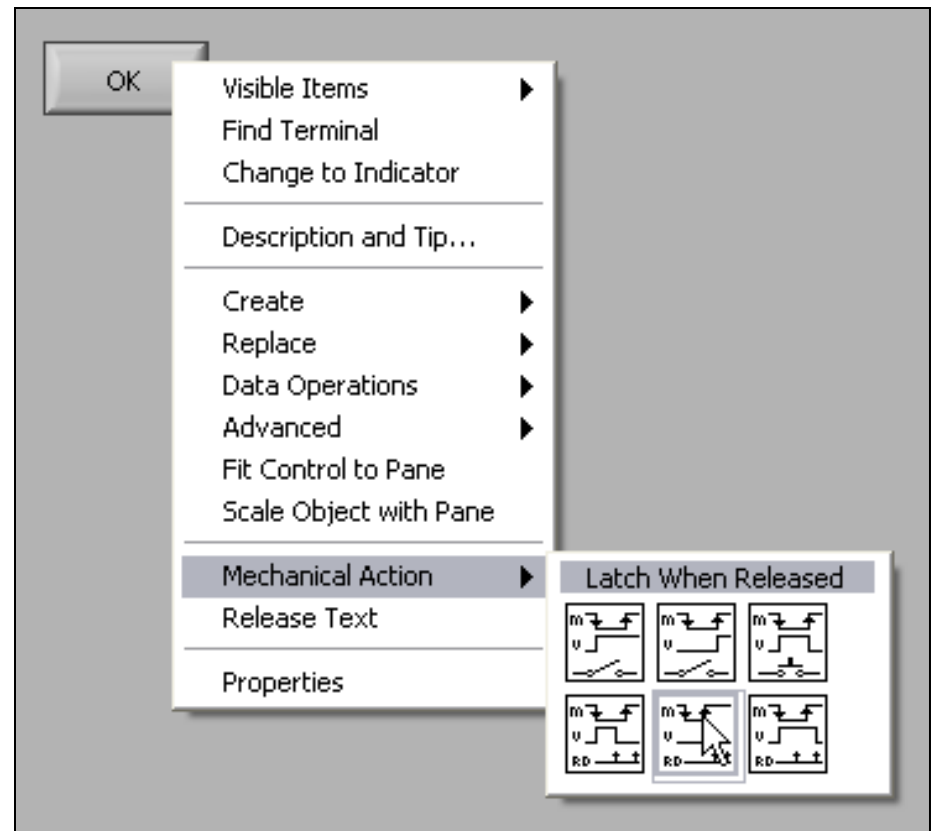
B. Tipos de Dados – Numéricos

- O tipo de dado numérico representa números de vários tipos
- Para mudar a representação de um numérico, clique com o direito no controle, indicador ou constante, e selecione **Representation** a partir do shortcut menu



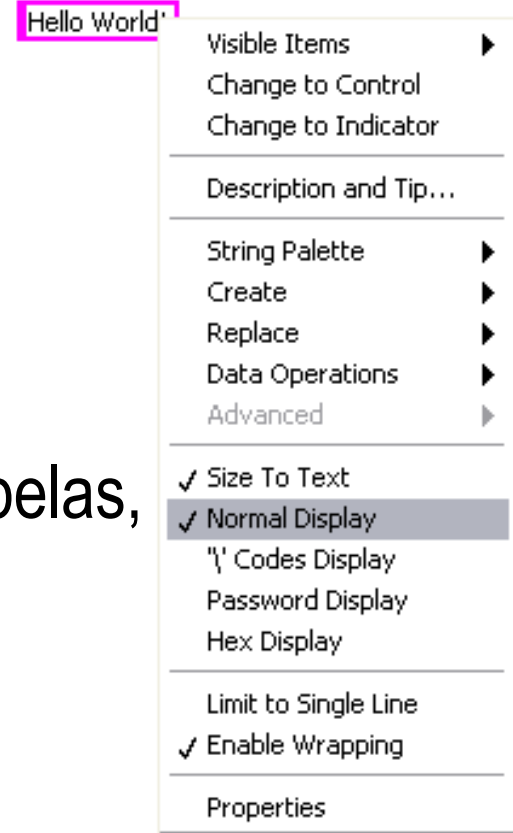
B. Tipos de Dados – Boleano

- O comportamento de um controle booleano é especificado pela ação mecânica
- No LabVIEW, o tipo de dado booleano é representado com a cor verde



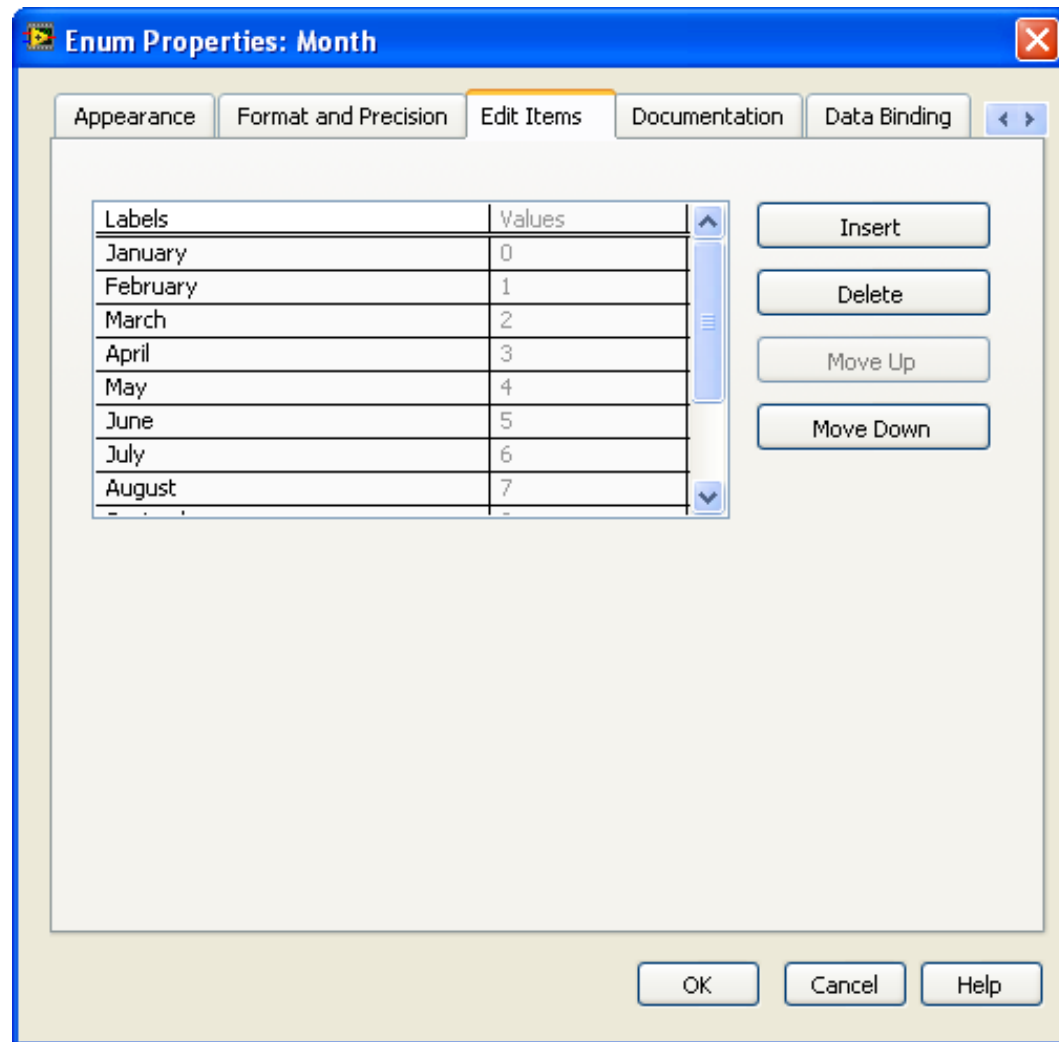
B. Tipos de Dados – String

- Uma sequência de caracteres ASCII visualizáveis ou não
- No painel frontal, strings aparecem como tabelas, caixas de entrada de texto e títulos
- Altere o tipo de exibição a partir do short-cut menu: Normal, '/' Codes, Hex e Password
- Edite e manipule strings com as funções de String no diagrama de blocos
- No LabVIEW, o tipo de dado string é representado em Cor-de-Rosa



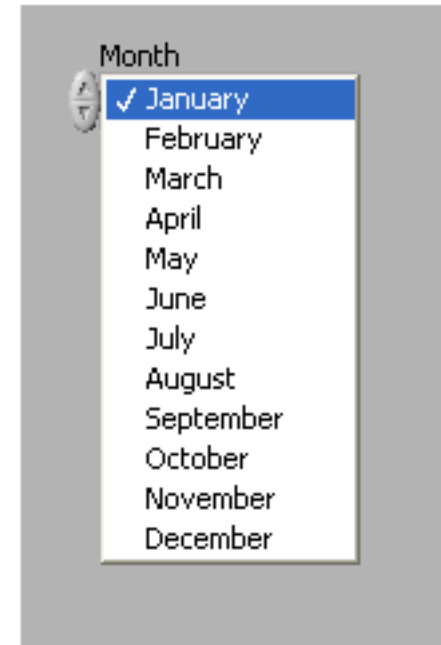
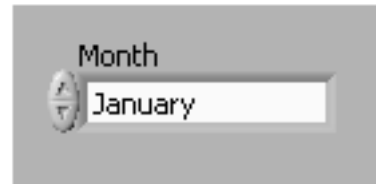
B. Tipos de Dados – Enum

Um **enum** representa um par de valores, um string e um numérico, onde o **enum** pode ser um valor de uma lista de valores definidos



B. Tipos de Dados – Enum

- Enum: indicador, constante ou controle enumerado
- Enums são muito úteis porque é mais fácil manipular números do que strings no diagrama de blocos



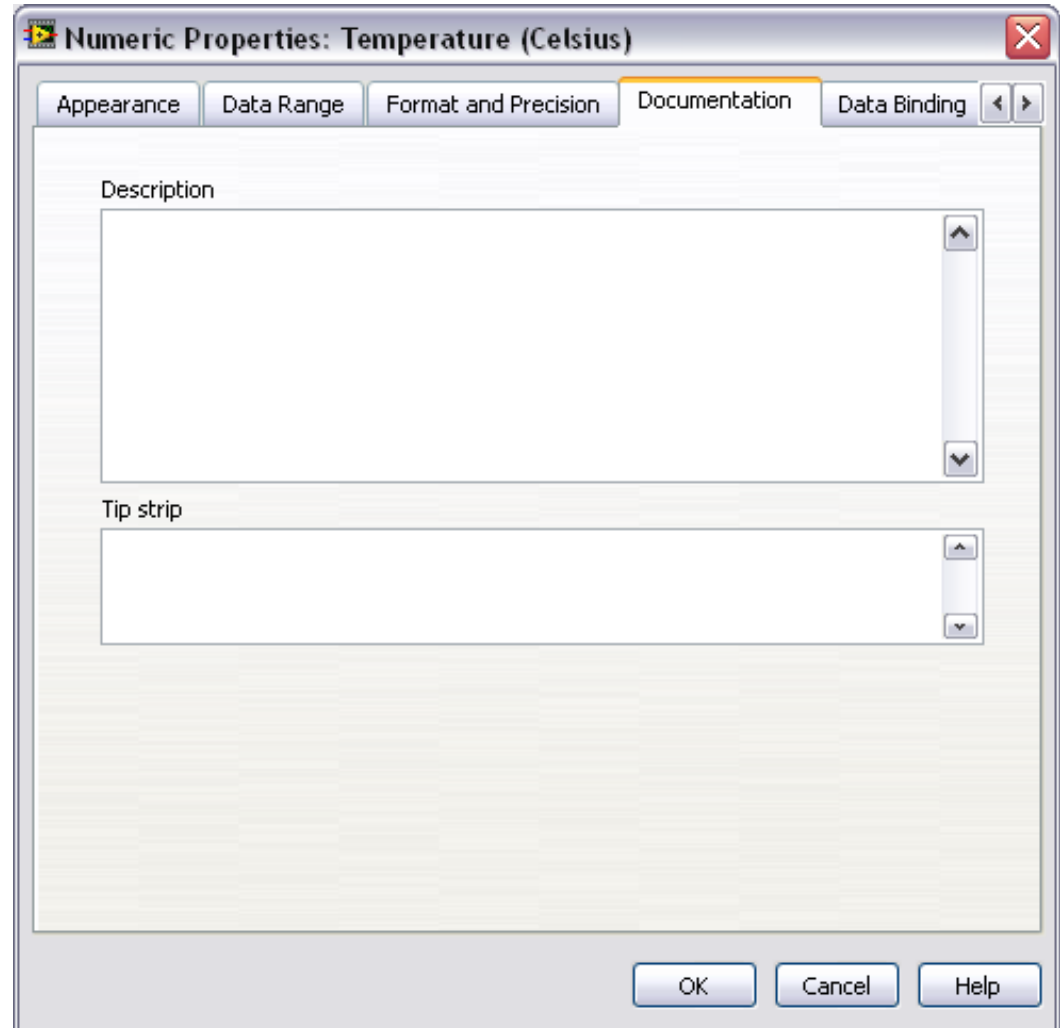
B. Tipos de Dados – Dinâmicos



- Armazena a informação gerada ou adquirida por um VI Expresso
- VIs que não são Expressos não aceitam tipo de dado dinâmico:
 - Para usar uma função ou VI nativa na análise ou processamento de um tipo de dado dinâmico, você deve converter o tipo de dado
 - Entradas ou Indicadores de dados Numéricos, waveform ou booleanos automaticamente convertem o dado dinâmico quando conectados
- No LabVIEW, o tipo de dado Dinâmico é representado com a cor **azul escuro**

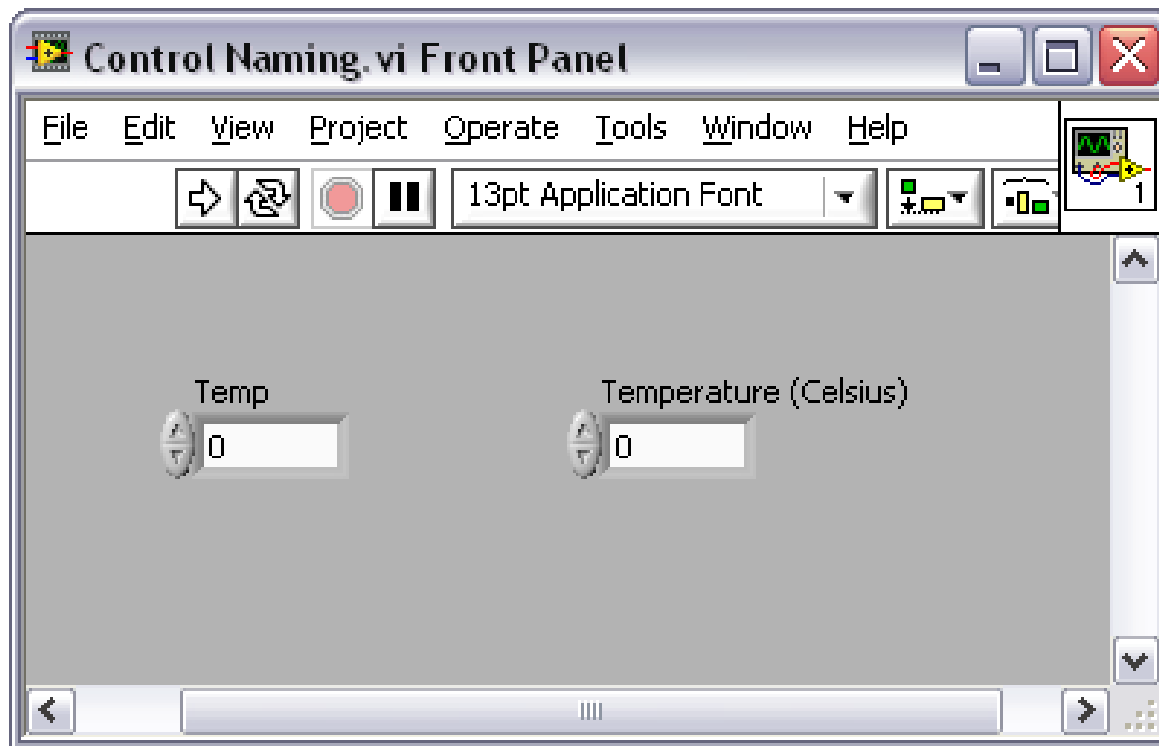
C. Documentando um Código – Painel Frontal

- Tip Strips
- Descrições
- Propriedades do VI
- Um bom Projeto



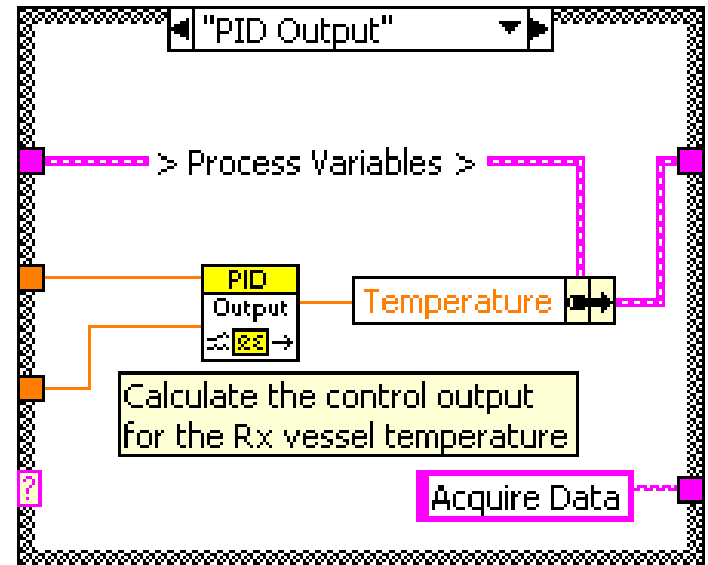
C. Documentando um Código – Nomeando

Fornecer nomes descritivos aos controles e indicadores aumenta a usabilidade do painel frontal

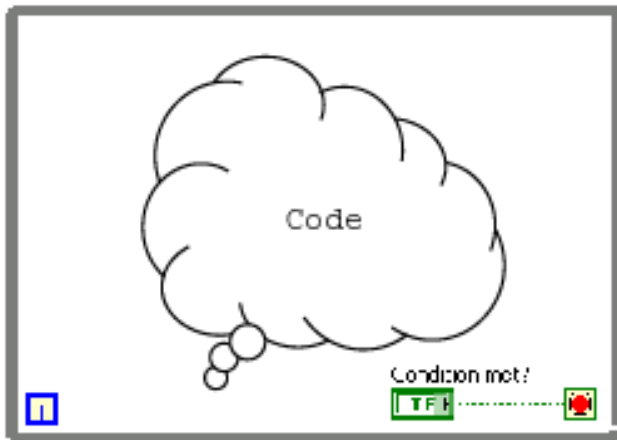


C. Documentando o Código – Diagrama de Blocos

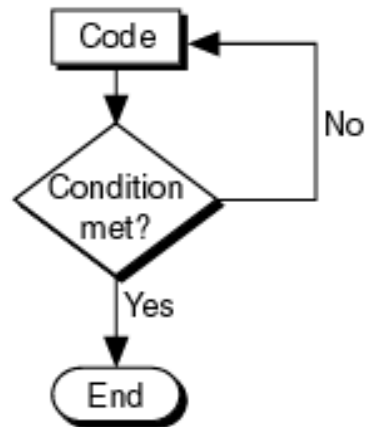
- Use comentários no diagrama de blocos para:
 - Descrever Algoritmos
 - Informar os dados contido nos fios
- Use a ferramenta de nomeação para inserir um texto livre a partir da paleta **Functions**



D. Loop While



Loop While do LabVIEW



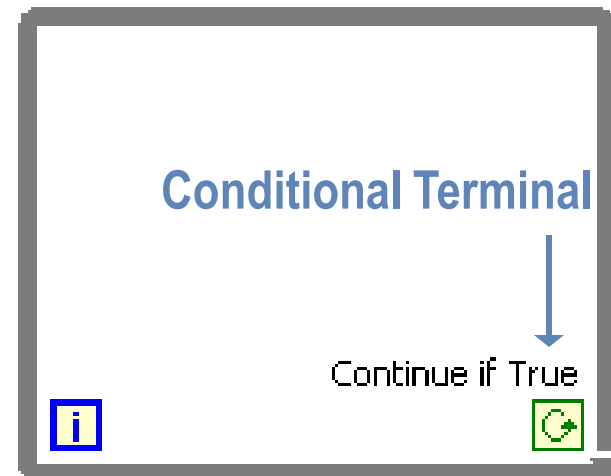
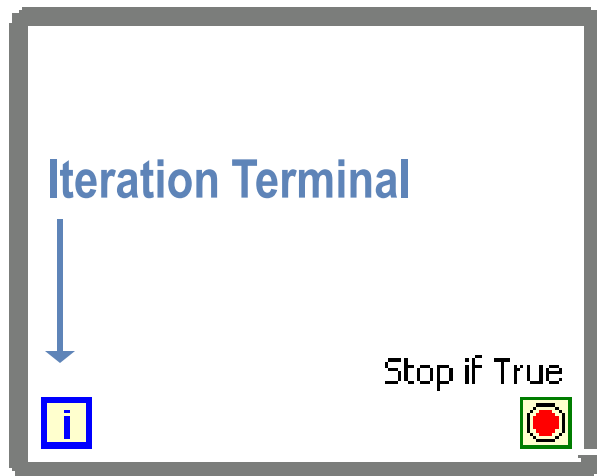
Fluxograma

```
Repeat (code);  
Until Condition met;  
End;
```

Pseudo Código

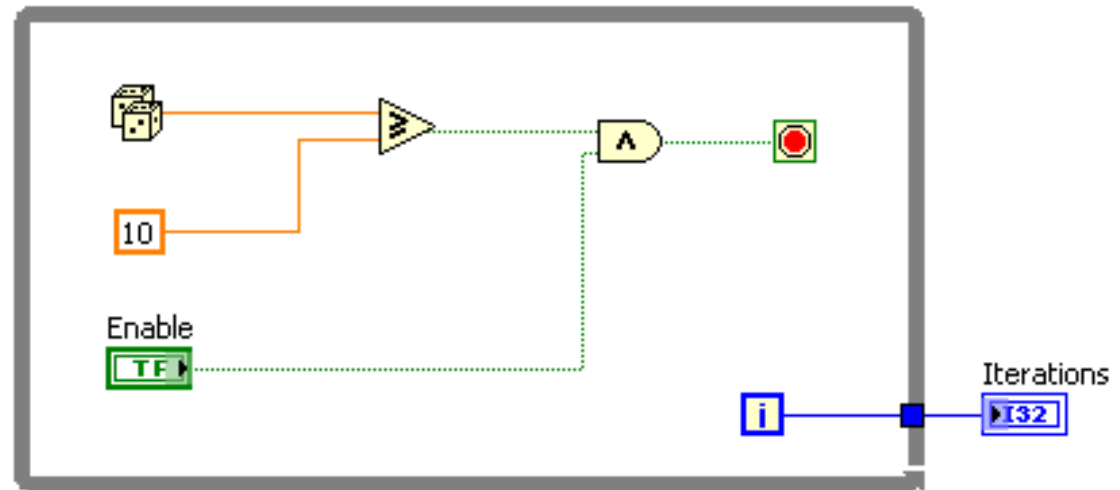
D. Loop While

- Terminal de Interação: retorna o número de vezes que o loop executa, iniciado em zero
- Terminal Condicional: define quando o loop pára



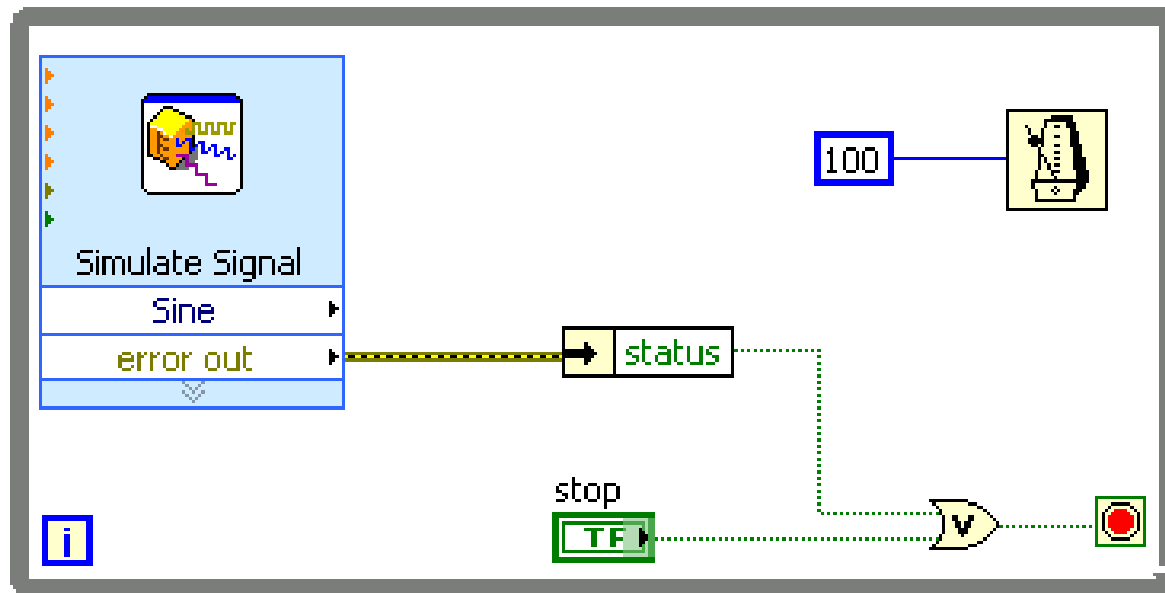
D. Loop While – Túneis

- Túneis transferem dados para dentro ou fora das estruturas
- O túnel adota a cor do tipo de dado que a ele está conectado
- O dado é passado para fora do loop após o final da execução
- Quando um túnel estiver passando dados para dentro do loop, esse somente o executará após todos os dados chegarem

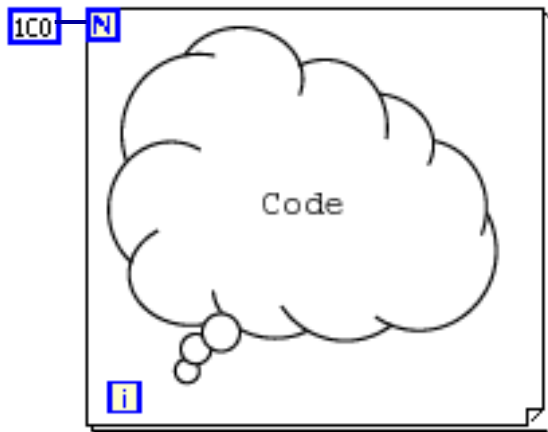


D. Loop While – Verificação e Gerenciamento de Erros

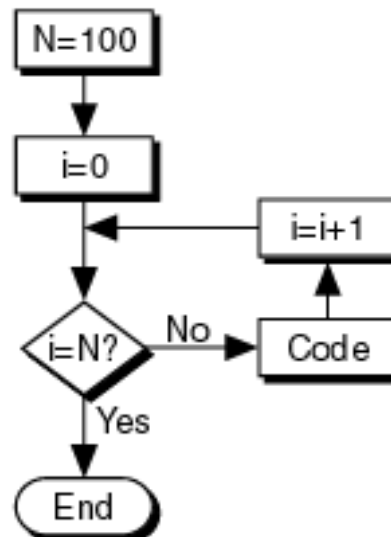
Use um cluster de erro para parar o Loop While se acontecer algum erro.



E. Loop For



Loop For do LabVIEW



Fluxograma

```
N=100;  
i=0;  
Until i=N:  
    Repeat (code;i=i+1);  
End;
```

Pseudo Código

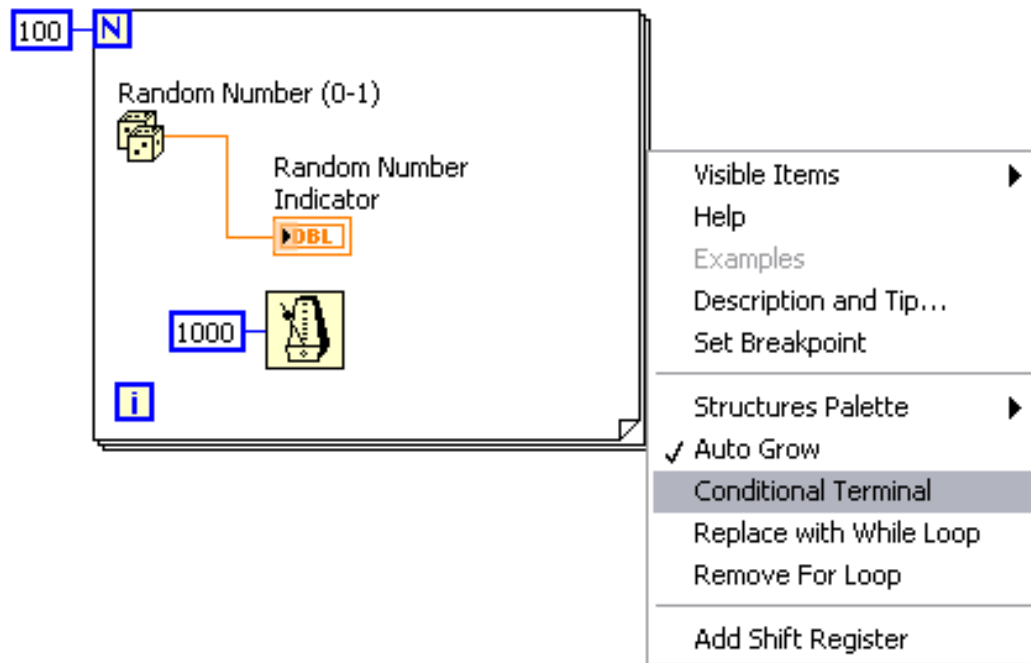
E. Loop For

- Crie um Loop For da mesma maneira que você cria um Loop While
- Se você precisar substituir um Loop While existente por um Loop For, clique com o direito na borda do Loop While e selecione **Replace with For Loop** a partir do shortcut menu
- O valor no terminal de contagem (um terminal de entrada) indica quantas vezes deve-se repetir o subdiagrama



E. Loop For – Terminal Condicional

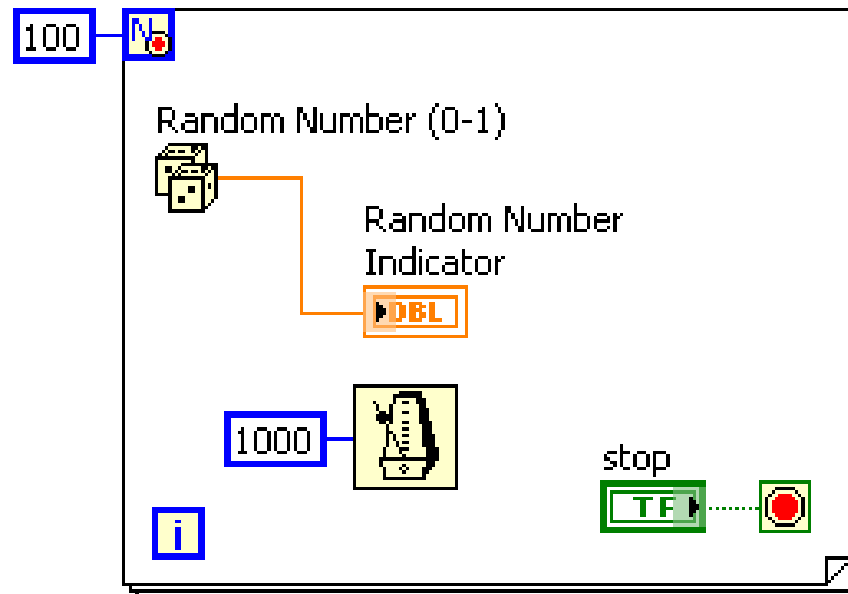
Você pode adicionar um terminal condicional para configurar o Loop For para parar quando uma condição booleana ou um erro ocorrer



E. Loop For – Terminal Condicional

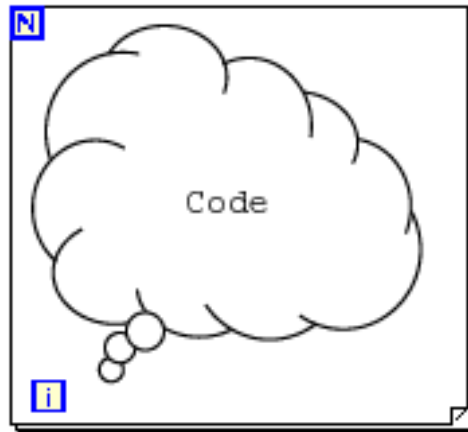
O Loop For configurado para saída condicional possuem:

- Ponto vermelho próximo ao terminal de quantidade de iterações
- Terminal Condicional no canto direito inferior



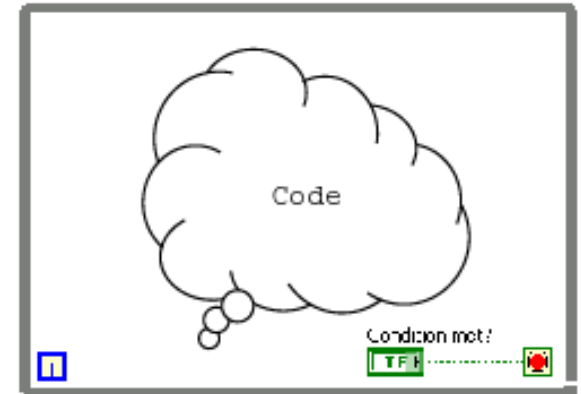
E. Comparação Loop For/Loop While

Loop For



- Executa um determinado número de vezes
- Pode executar “zero vezes”
- Os túneis automaticamente geram um array de dados

Loop While



- Pára a execução somente se um valor no terminal condicional atender à condição
- Precisa executar pelo menos uma vez
- Os túneis automaticamente fornecem o último valor

E. Loop For– Conversão Numérica

- Se você conectar um valor numérico ponto flutuante ao terminal de quantidade de iterações , o LabVIEW converte o maior valor numérico em um inteiro de 32 bits
- Um Loop For pode executar somente um número inteiro de vezes

Double-Precision
Floating Point

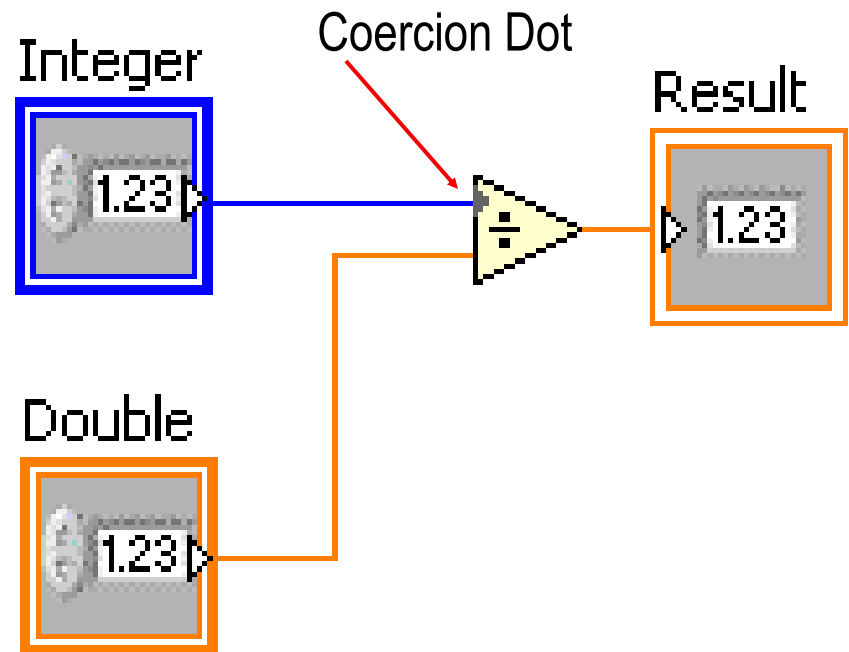


32-Bit Signed Integer

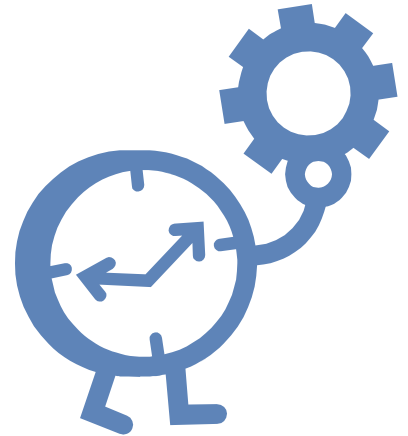


E. Loop For– Conversão Numérica

- Normalmente, quando você conecta diferentes tipos de representação às entradas da função, a função retorna o valor de saída no maior formato
- O LabVIEW escolhe a representação que utiliza mais bits



F. Temporizando um VI



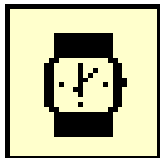
Por que você precisa temporizar um VI?

- Controlar a frequência que o loop executa
- Provê tempo ao processador para completar outras tarefas, como processar a interface de usuário

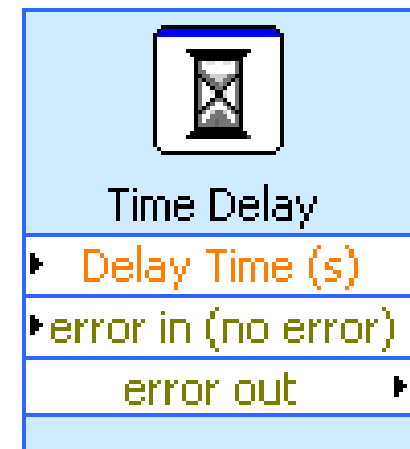
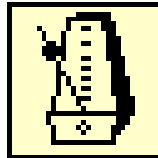
F. Temporizando um VI – Funções de Espera

- Uma função de espera dentro de um loop permite que o VI pare por uma certa quantidade de tempo
- Permite que o processador execute outras tarefas durante o tempo de espera
- Utiliza o clock em milisegundos do sistema operacional

Wait (ms)

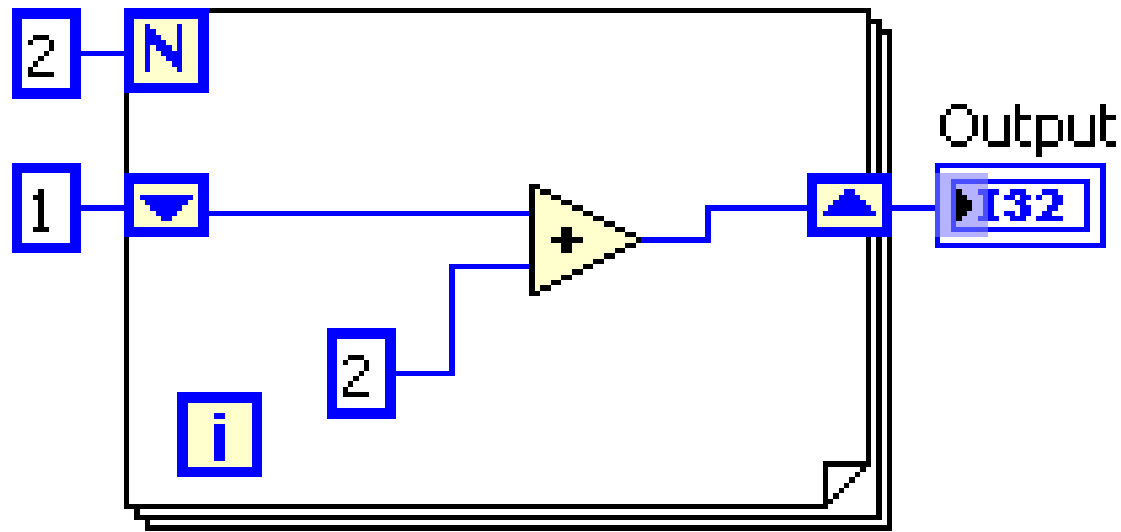


Wait Until
Next ms Multiple



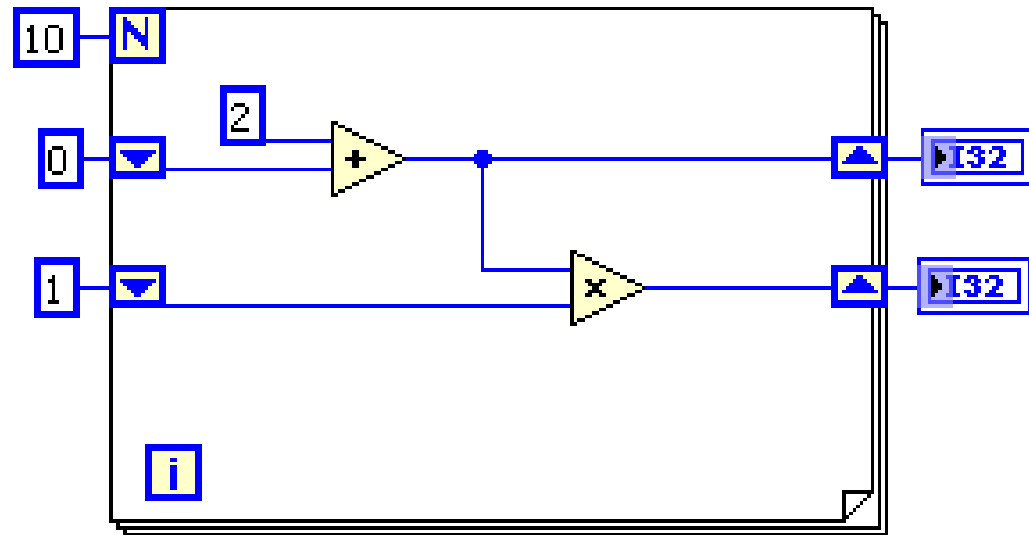
G. Transferência Interativa de Dados

- Na programação com Loops, freqüentemente é necessário saber valores de dados das iterações anteriores do loop
- Os Shift registers transferem valores de uma iteração do loop para a próxima



G. Transferência Interativa de Dados – Shift Registers

- Clique com o direito na borda do loop e selecione **Add Shift Register** a partir do shortcut menu
- O shift register armazena os dados ao final da iteração
- O shift register fornece o dado armazenado no início da próxima iteração

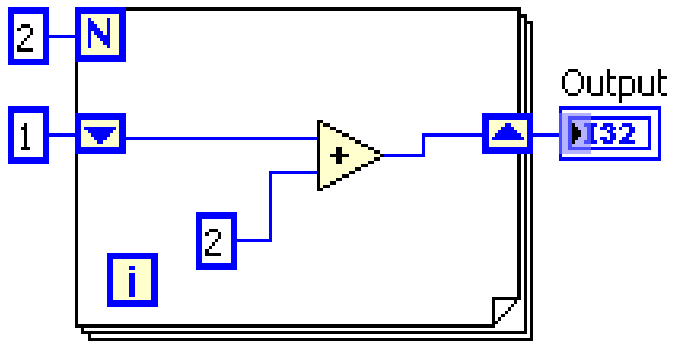
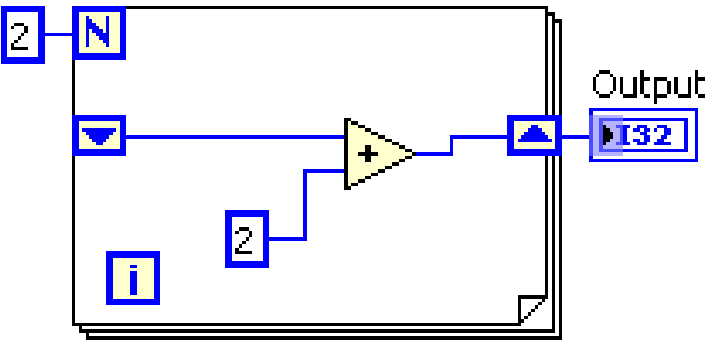


G. Transferência Interativa de Dados – Inicializando

Executa

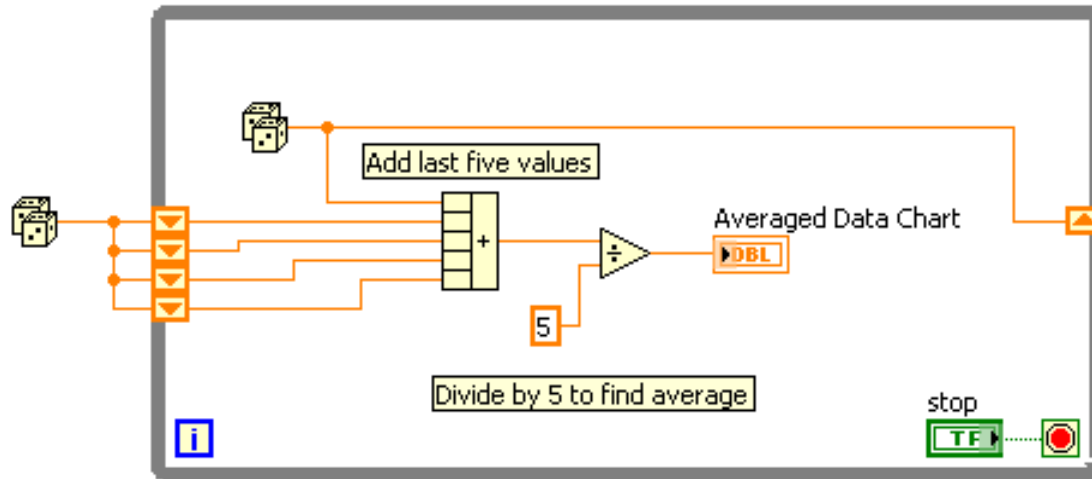
Finaliza VI

Executa novamente

| Diagrama de Blocos | 1ª Execução | 2ª Execução |
|--|-------------|-------------|
| <p>Shift Register Inicializado</p>  | Output = 5 | Output = 5 |
| <p>Shift Register Não Inicializado</p>  | Output = 4 | Output = 8 |

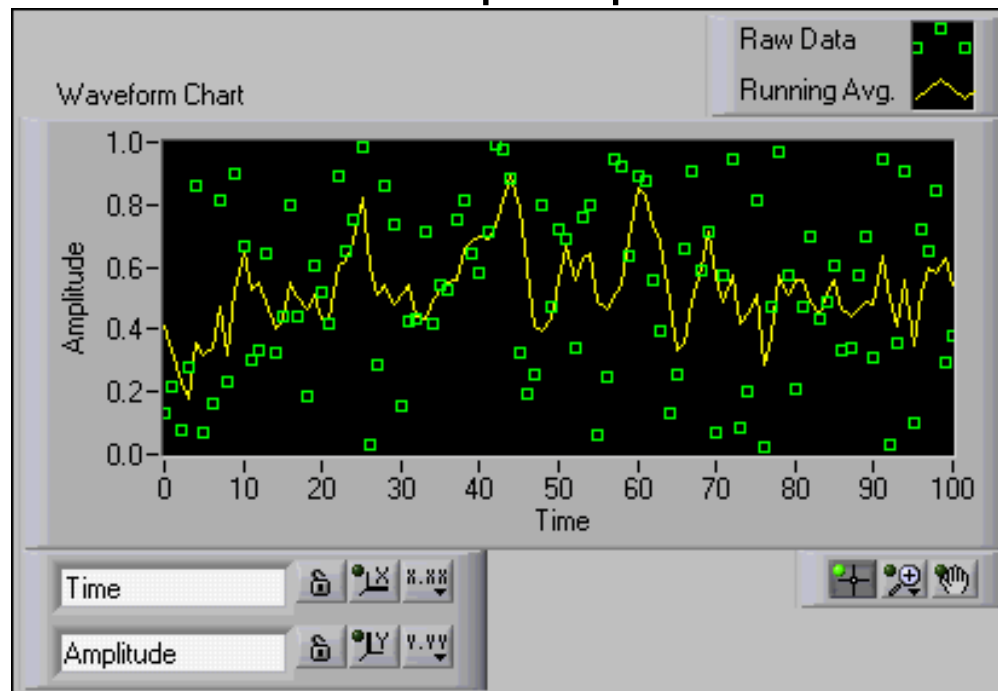
G. Transferência Interativa de Dados– Shift Registers Empilhados

- Shift registers empilhados armazenam valores de múltiplas iterações anteriores e carregam estes valores para as próximas iterações
- Clique com o direito no shift register da esquerda e selecione **Add Element** a partir do shortcut menu



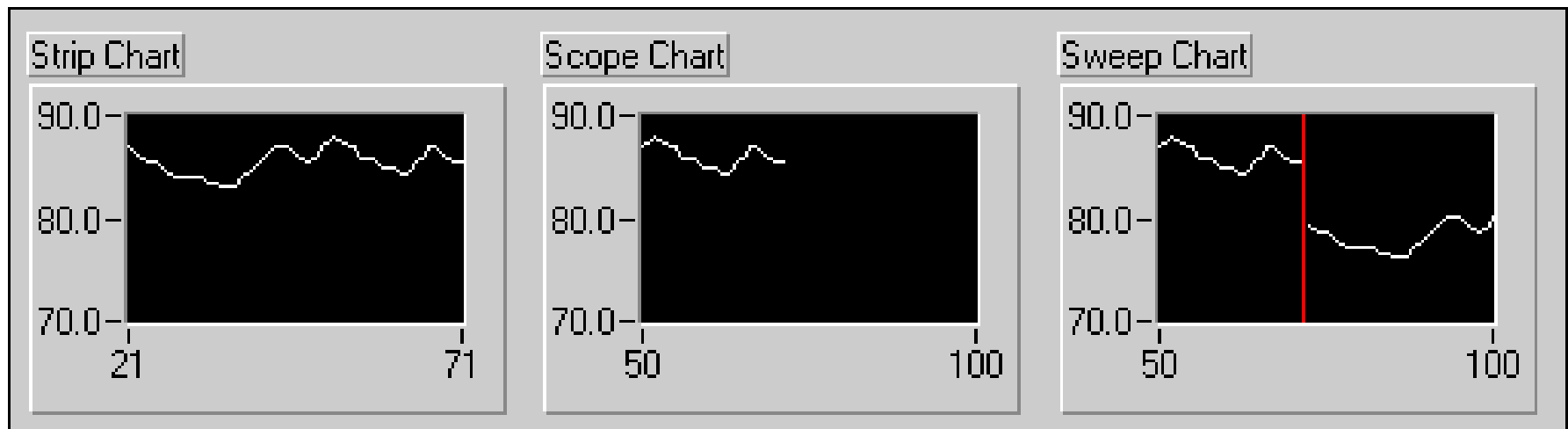
H. Plotando Dados – Waveform Chart

- Um tipo especial de indicador numérico que mostra um ou mais dados em um gráfico, tipicamente adquiridos a uma taxa constante
- Exibe um único ou múltiplos plots

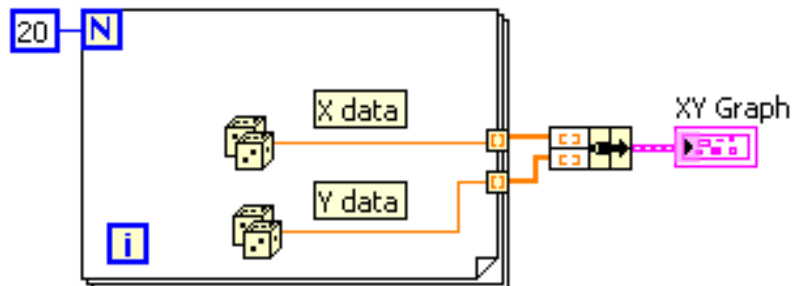
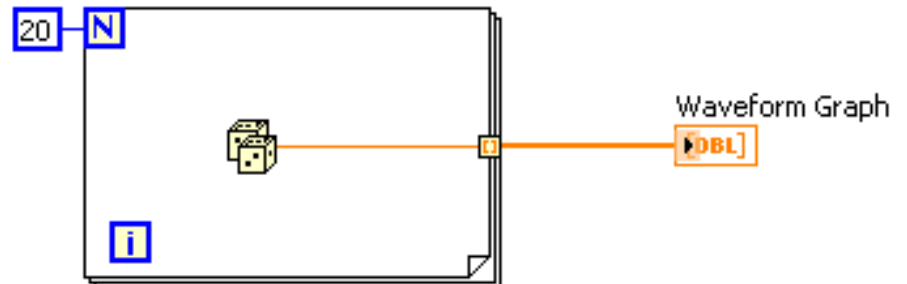
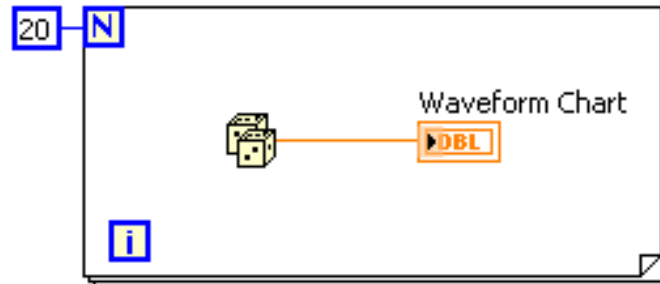
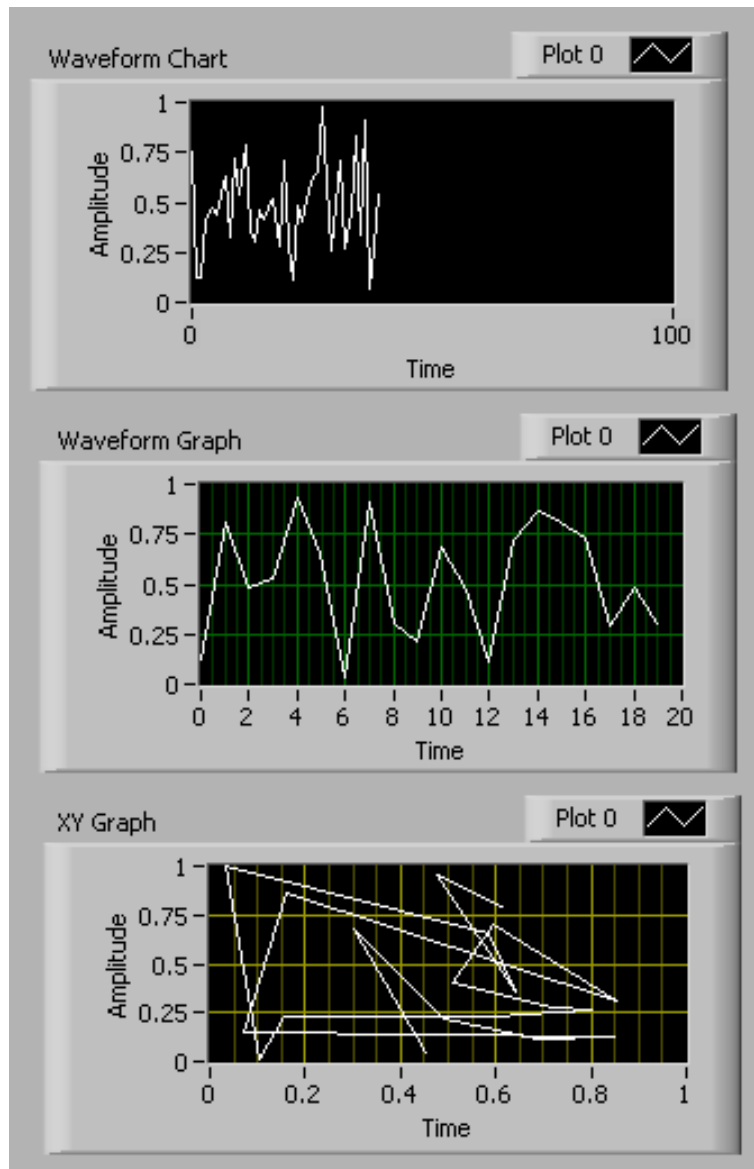


H. Plotando Dados – Modos de atualização do Chart

- Clique com o direito no chart e selecione **Advanced»Update Mode** a partir do shortcut menu
- O modo *Strip chart* é o modo de atualização padrão
- Os modos *Scope chart* e *Sweep chart* exibem plots significativamente mais rápidos do que o modo *Strip Chart*

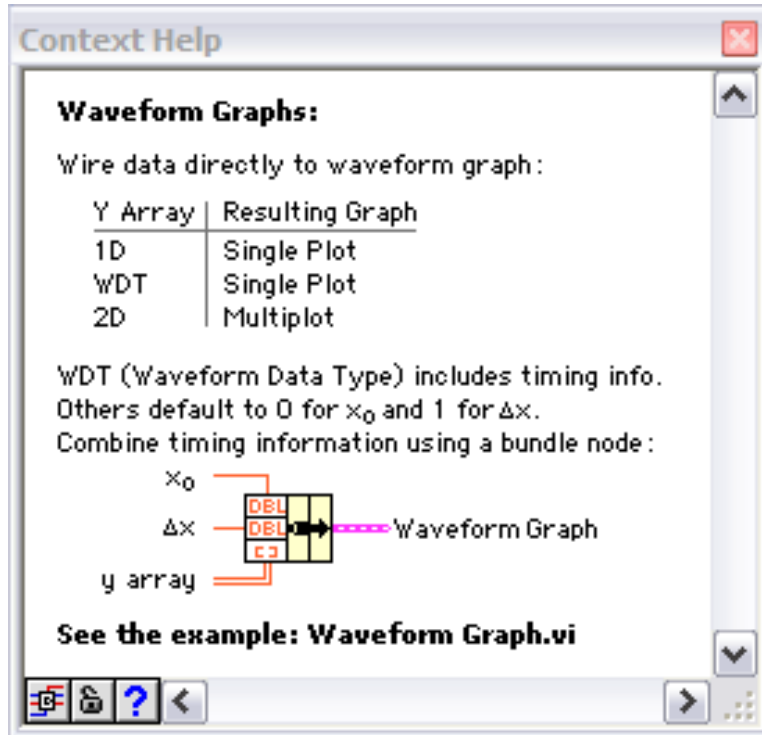


H. Plotando Dados



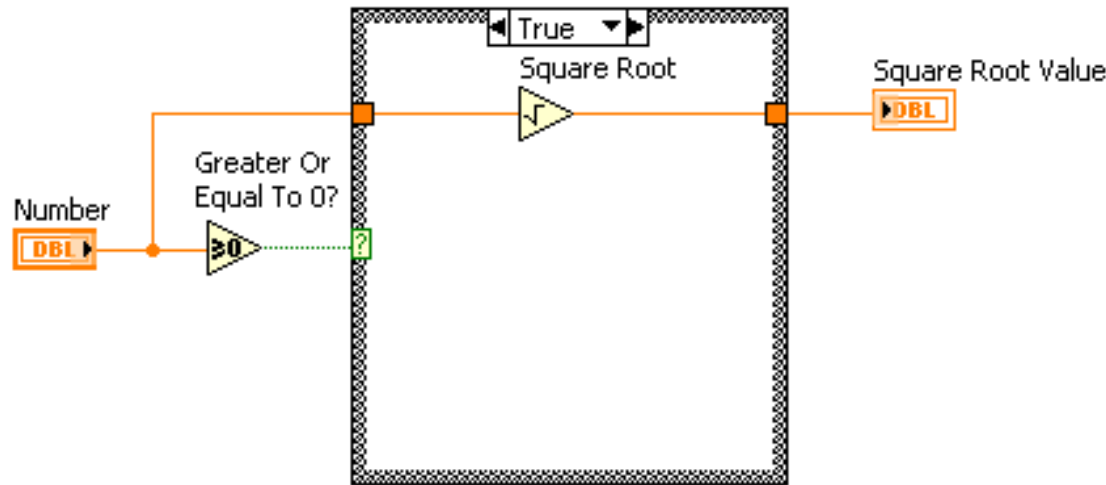
H. Plotando Dados– Waveform Graphs

Use a janela Context Help para determinar como conectar múltiplos gráficos em *Waveform Graphs* e Gráficos XY



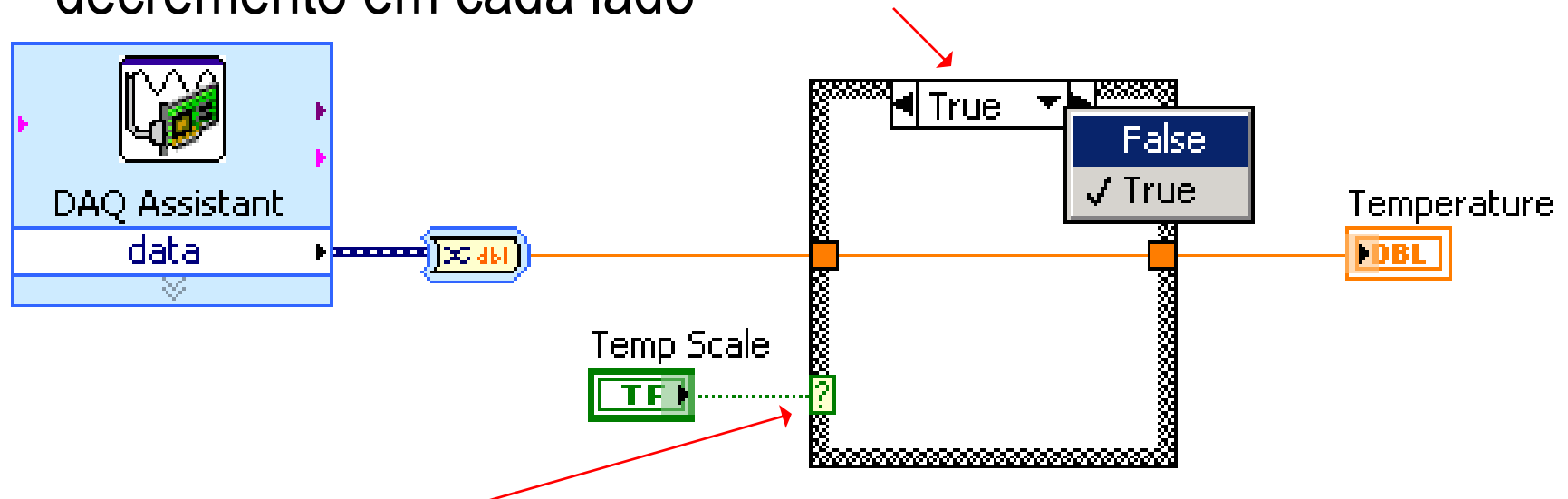
I. Estruturas Case

- Possuem dois ou mais subdiagramas ou casos
- Exibe e executa somente um subdiagrama por vez
- Um valor de entrada determina qual subdiagrama será executado
- Declarações semelhantes ao *case* ou *if...then...else* em linguagens de programação baseadas em texto



I. Estruturas Case

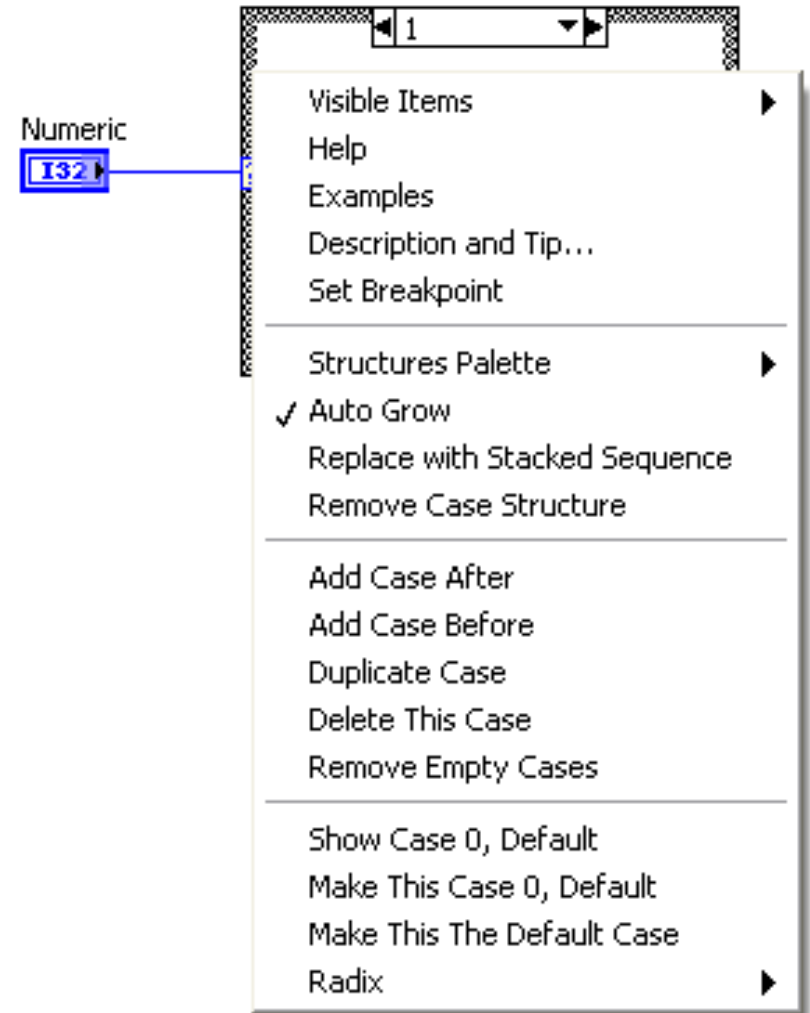
- O nome do case contém a identificação de seleção do caso atual no centro e os botões de incremento e decremento em cada lado



- Terminal de Seleção: Conecte um valor de entrada, ou seletor, para determinar qual caso irá executar

I. Estruturas Case – Caso Padrão

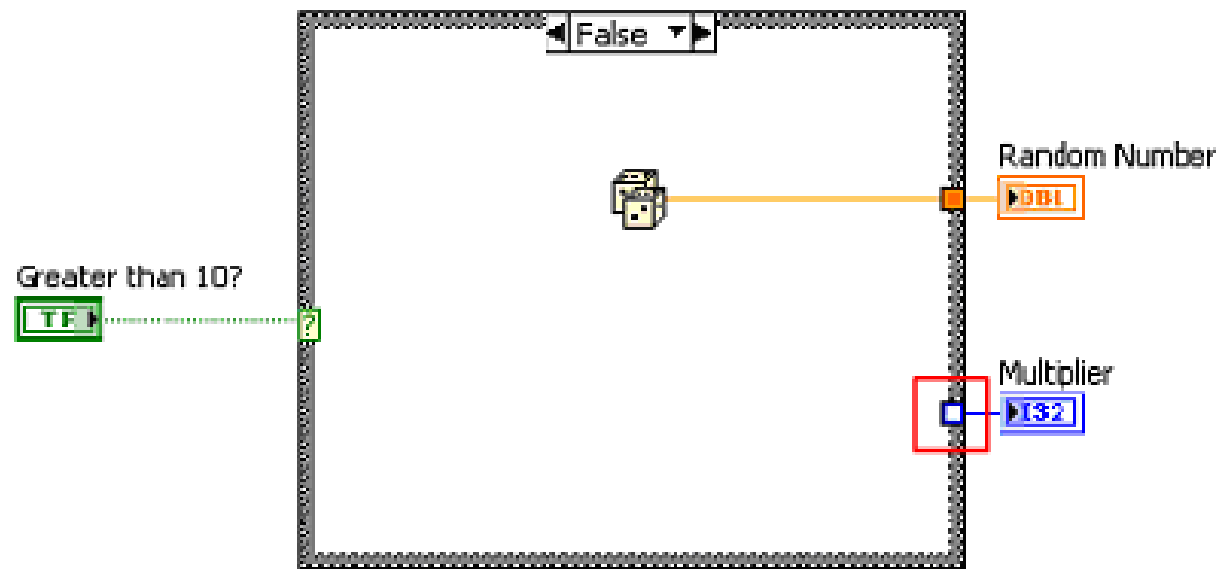
- Você pode especificar um caso padrão para cada Estrutura Case
 - Se você especificou casos para os valores 1, 2 e 3, mas teve um valor de entrada 4, a Estrutura Case executa o caso padrão
- Clique com o direito na borda da Estrutura Case para adicionar, duplicar, remover ou reorganizar os casos e também para selecionar o caso padrão



I. Estruturas Case – Túneis de Entrada & Saída

Você pode criar múltiplos túneis de entrada e saída

- As entradas estão disponíveis para todos os casos
- Você deve definir valores em cada túnel de saída para todos os casos



I. Estruturas Case – Use Default if Unwired

Os valores Default são:

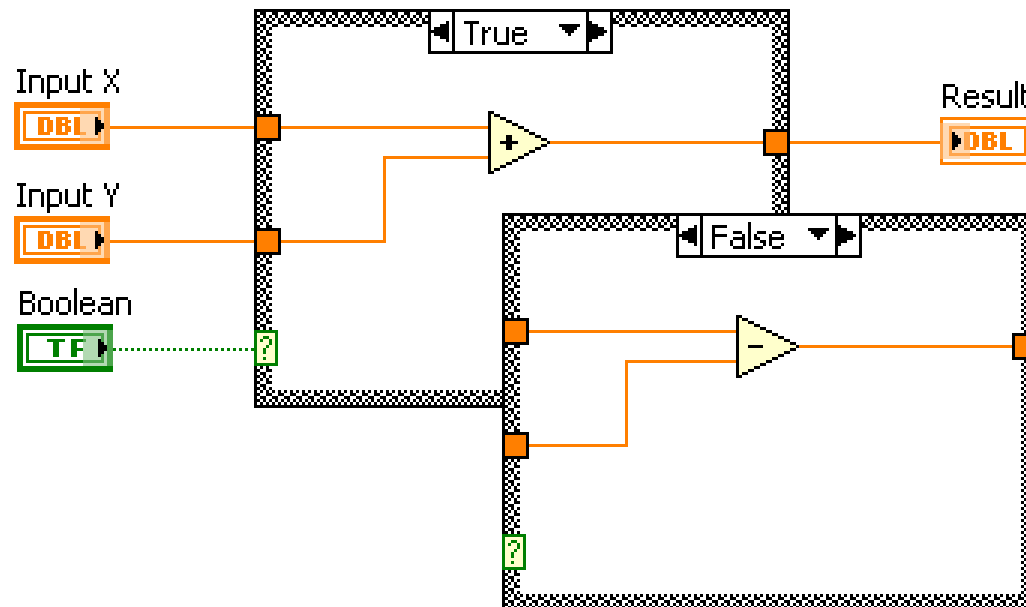
| Tipo de Dado | Valor Padrão |
|--------------|--------------|
| Numérico | 0 |
| Booleano | FALSE |
| String | Empty |

Evite usar a opção *Use Default If Unwired*

- Adiciona um nível de complexibilidade ao código
- Complica a depuração do código

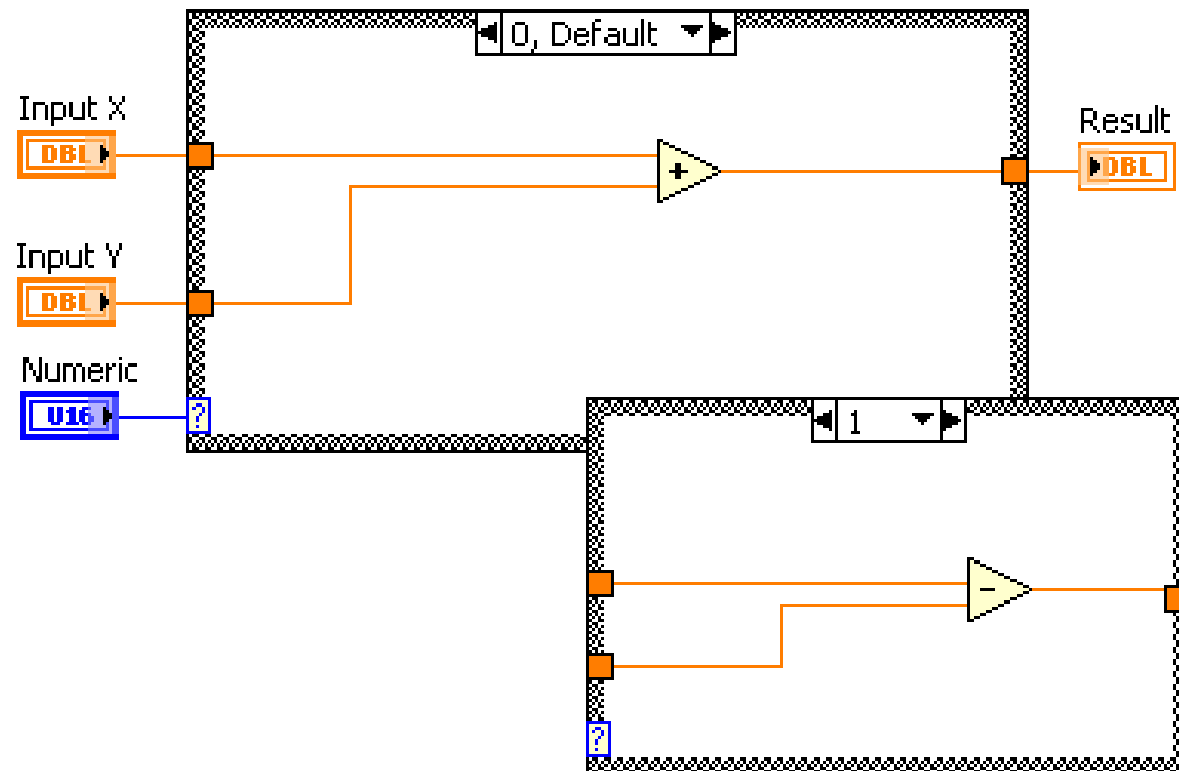
I. Estruturas Case – Boleano

- Entradas booleanas criam dois casos: True e False



I. Estruturas Case – Inteiro

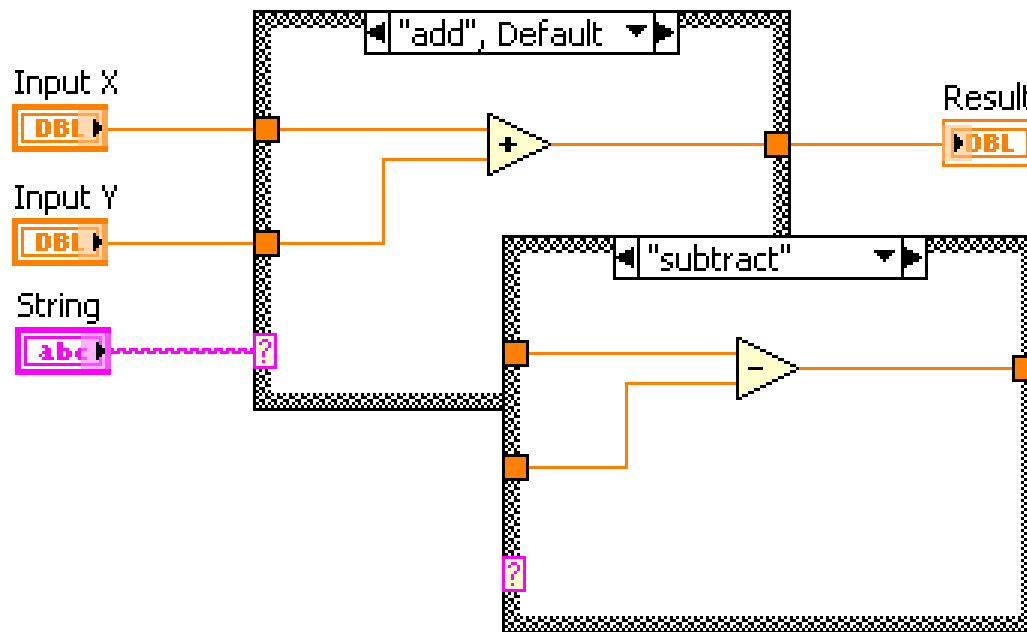
- Adicione um caso para cada inteiro que for necessário
- Inteiros sem um caso definido executam o caso padrão



INSTITUTO MAUÁ DE TECNOLOGIA

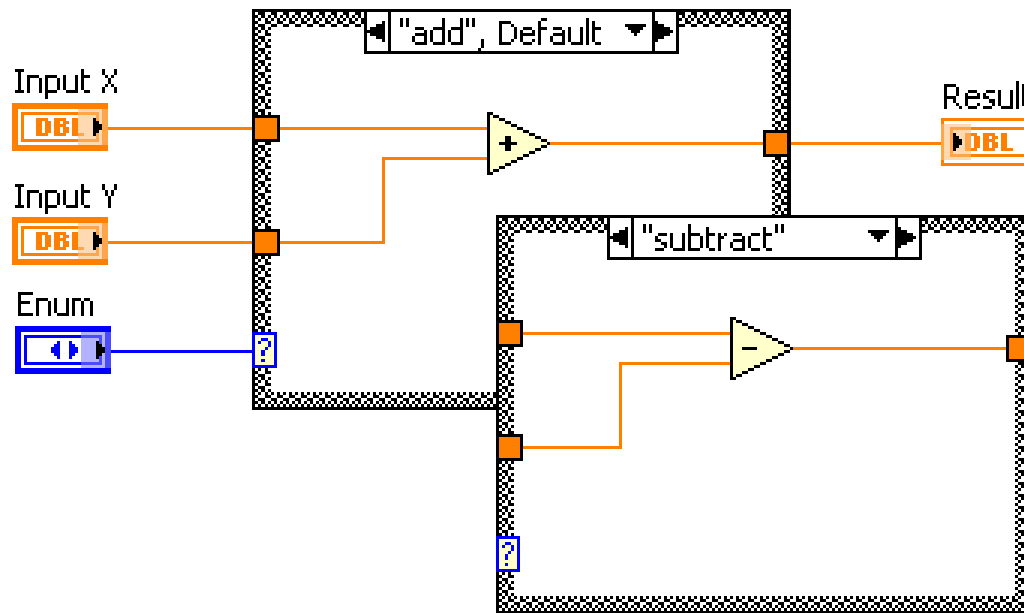
I. Estruturas Case – String

- Adicione um caso para cada string necessário
- Strings sem um caso definido executam o caso padrão



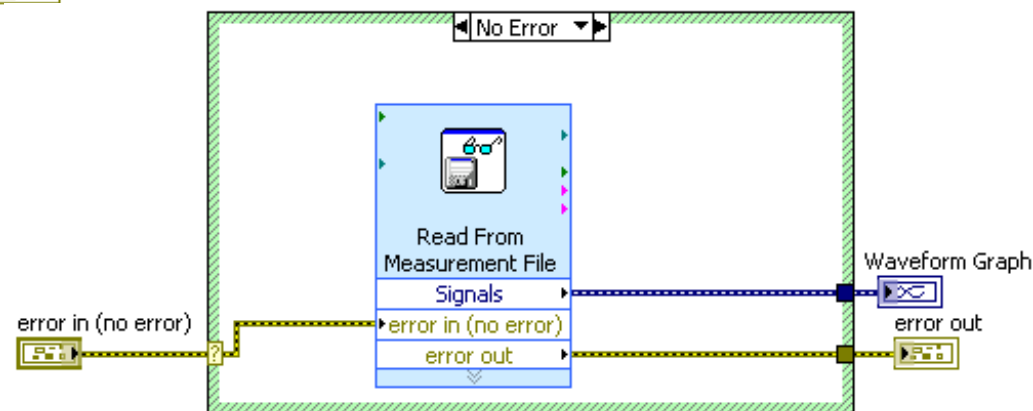
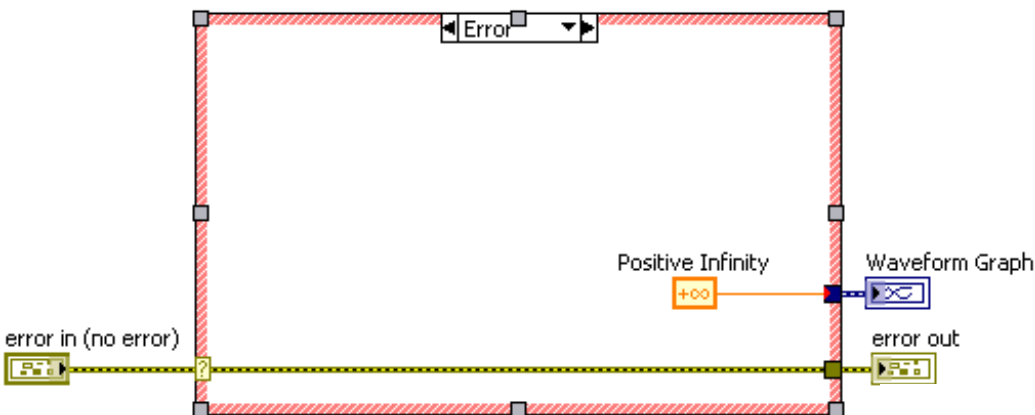
I. Estruturas Case – Enum

- Provê ao usuário uma lista de itens para selecionar
- O seletor de caso exibe um caso para cada item do controle tipo enumerado



I. Estruturas Case – Verificação e Gerenciamento de Erro

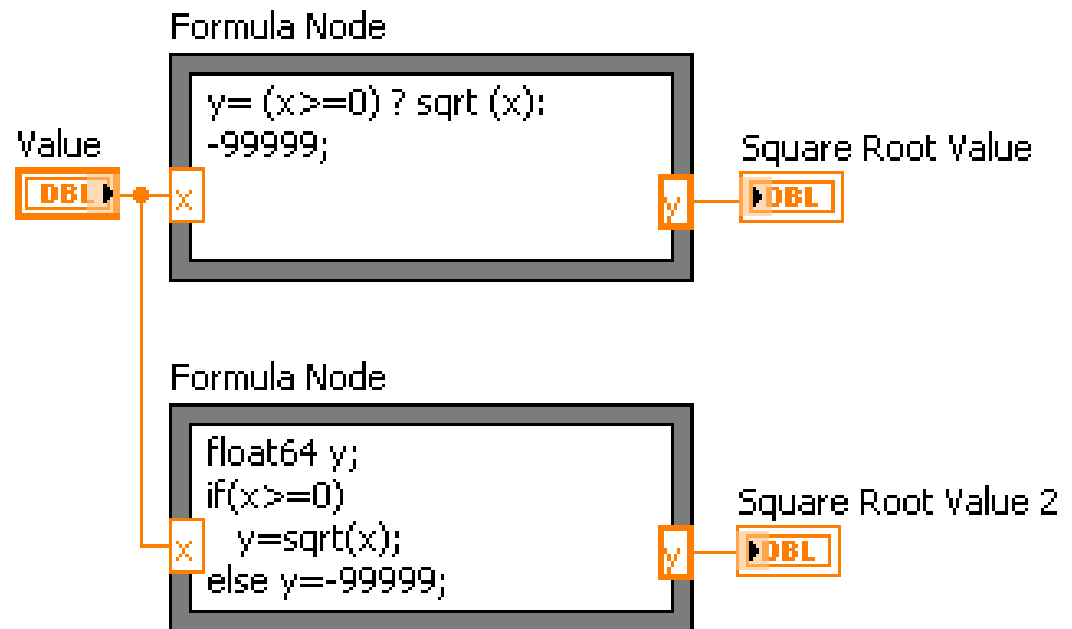
Use a Estrutura Case dentro de VIs para executar um código se não houver erro e pular a execução se houver erro



INSTITUTO MAUÁ DE TECNOLOGIA

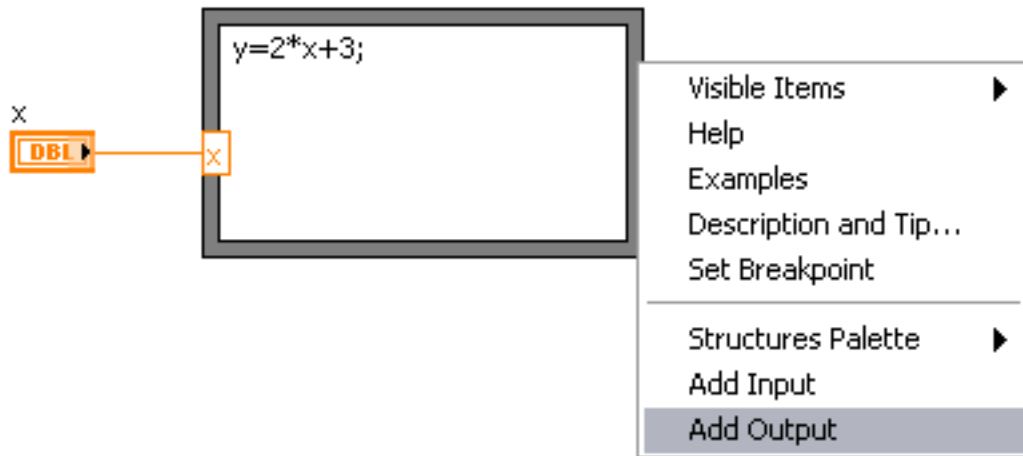
J. Formula Node

- Baseado em texto para desempenhar operações matemáticas no diagrama de blocos
- Aceita versões de comandos como *if*, *while loops*, *for loops* e *do loops* que são familiares para programadores em C



J. Formula Node

- Clique com o botão direito na borda e selecione **Add Input** ou **Add Output**
- Insira a variável de entrada ou saída
- Entre com a equação na estrutura
- Termine cada comando com um ponto-e-vírgula (;)



J. Formulas

- A VI Expresso *Formula* utiliza uma interface de calculadora para criar equações matemáticas
- Executa mais funções matemáticas do que uma calculadora científica

