



## PSI 3214 - LABORATÓRIO DE INSTRUMENTAÇÃO ELÉTRICA

### EXPERIÊNCIA 1 - DESENVOLVIMENTO DE INSTRUMENTOS VIRTUAIS

L.R.Y. /Revisão: V.H.N. e E.G.  
Ver\_1.1\_15.08.2015

#### 1. Objetivos

Desenvolver a capacidade de compreensão, análise e implementação de medições elétricas baseadas em instrumentos virtuais. Através das atividades práticas os alunos devem:

- Familiarizar-se com o conceito de instrumentação virtual;
- Conhecer os principais recursos da ferramenta LabVIEW;
- Elaborar programa de comunicação com osciloscópio e gerador;
- Desenvolver um programa de geração de formas de ondas.

#### 2. Instrumentos Virtuais

Atualmente, uma parte significativa das atividades de engenheiros e de cientistas envolve testes e ensaios baseados em medições de diversos parâmetros. Por exemplo, os fabricantes de celulares, automóveis ou aviões, antes de lançarem um novo produto, precisam realizar inúmeros testes controlados para a certificação e homologação junto aos órgãos como ANATEL<sup>1</sup>, DENATRAN<sup>2</sup>, ANAC<sup>3</sup> entre outros. Mesmo em atividades didáticas ou projeto dos alunos de graduação é comum a necessidade de coletar uma grande quantidade de dados para se conhecer a resposta em frequência de um circuito ou o comportamento de um dispositivo. A partir dos dados coletados é necessário construir gráficos, calcular estatísticas, realizar análises, entre outros. Essas atividades demandariam muito esforço e tempo se realizados manualmente. Felizmente, foram desenvolvidas soluções para automatizar o processo de coleta de dados, análise e visualização das informações, como veremos a seguir.

##### 2.1 Histórico

Em 1975 a IEEE<sup>4</sup> padronizou uma interface de comunicação entre equipamentos que foi denominada IEEE-488<sup>5</sup>, comumente chamado de GPIB (General Purpose Interface Bus). A

---

<sup>1</sup> Agência Nacional de Telecomunicações

<sup>2</sup> Departamento Nacional de Trânsito

<sup>3</sup> Agência Nacional de Aviação Civil

<sup>4</sup> Institute of Electrical and Electronic Engineer

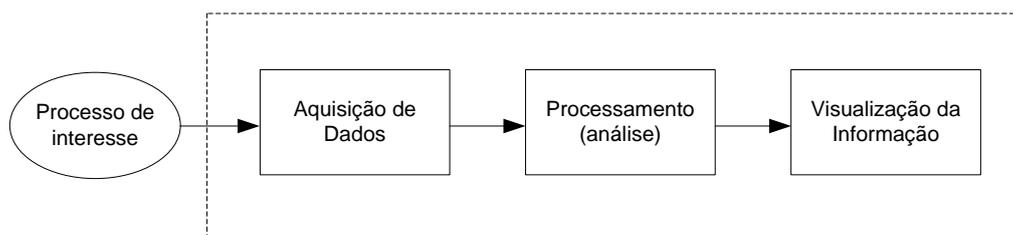
<sup>5</sup> <https://en.wikipedia.org/wiki/IEEE-488>

padronização de comunicação permitiu que os equipamentos de medições de diferentes fabricantes pudessem ser interconectados e os dados pudessem ser enviados para um computador. No início, os custos dos equipamentos eram elevados e havia a necessidade de programadores treinados para construir um sistema de medição automatizada.

Na década de 1980 o programa espacial americano estava em um período de grande avanço com o desenvolvimento da nave Columbia. A National Instruments (NI) foi contratada para produzir uma plataforma de desenvolvimento de sistemas baseado em uma linguagem programação gráfica. Essa plataforma foi denominada de LabVIEW (**L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench). Tornou-se uma ferramenta popular sendo largamente utilizada na academia, instituição de pesquisa e na indústria. A Escola Politécnica possui uma licença acadêmica do LabVIEW que permite que os alunos instalem programa em seus próprios computadores para estudo, desenvolvimento de atividades didáticas e de projetos.

## 2.2 Conceitos básicos

Um instrumento virtual tem como objetivo permitir a visualização de informações relacionada com um determinado processo de interesse conforme ilustrado na Figura 1.



**Figura 1** – Elementos básicos de um Instrumento Virtual.

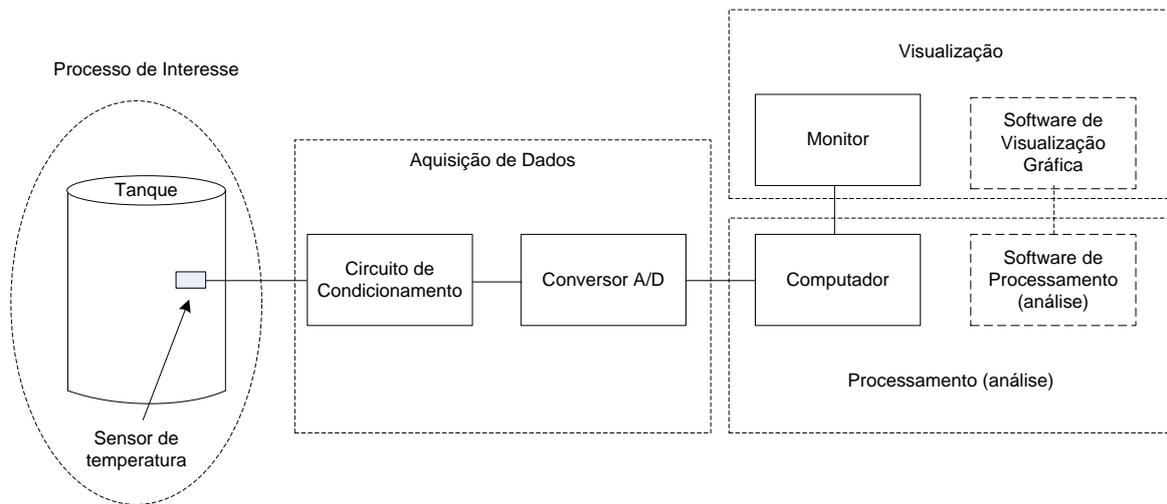
O processo de interesse pode ser um dispositivo, circuito, equipamento, planta industrial entre outros. O bloco de aquisição de dados é responsável pela transformação de sinais coletados junto ao processo de interesse em dados digitais e armazenamento em uma memória. O bloco de processamento ou análise realiza operações matemáticas nos dados coletados a fim de extrair informações que se deseja obter (por exemplo, médias, histogramas, componentes espectrais<sup>6</sup> entre outros). O último bloco de visualização de informação é responsável pela apresentação do resultado em forma de tabela, gráfico, mostradores numéricos, barra de níveis ou ponteiros, que podem ser escolhidos conforme a necessidade.

**Exemplo 1:** Vamos ver a seguir um exemplo de aplicação de Instrumentação Virtual. A **Figura 2a** mostra uma representação esquemática de um sistema de monitoramento de temperatura de um tanque. Veja que todos os elementos básicos que vimos na Figura 1 estão presentes (caixa pontilhada). A temperatura do tanque é capturada por meio de um sensor de temperatura que

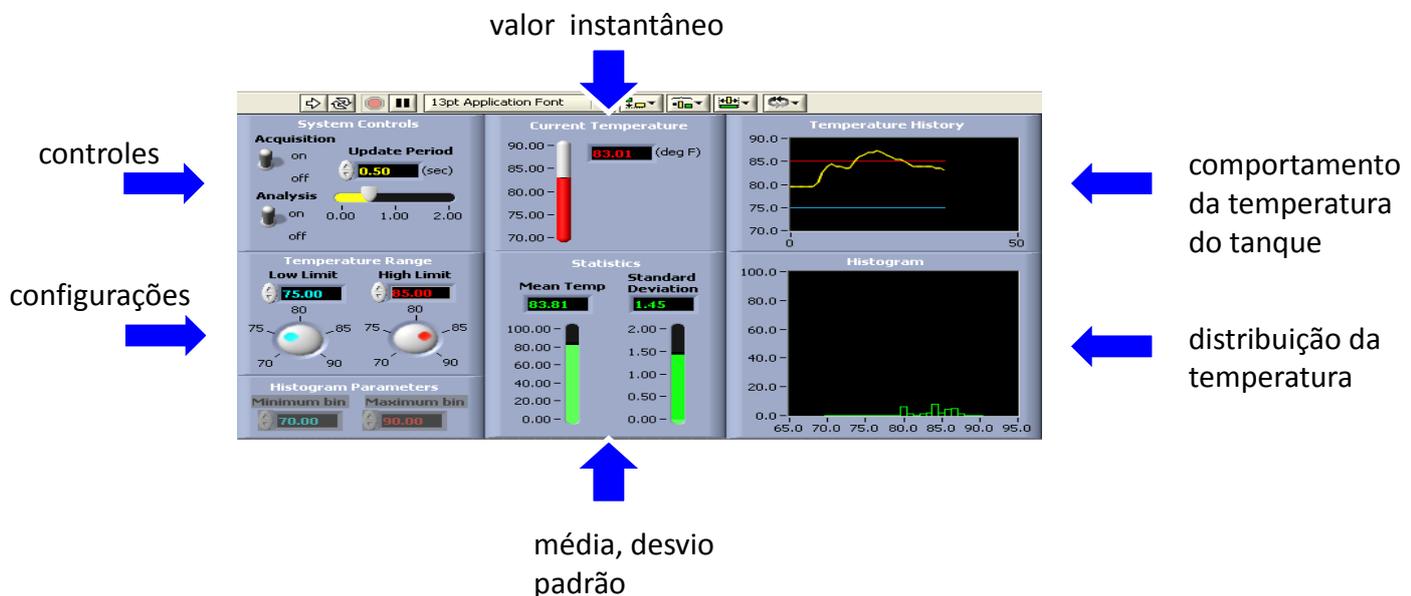
---

<sup>6</sup> Um sinal é composto por um ou mais componentes espectrais. Uma onda senoidal é composta por um único componente, enquanto que uma onda quadrada é formada por infinitos componentes.

produz uma pequena tensão elétrica nos seus terminais. Esse sinal é amplificado pelo circuito de condicionamento de sinais. O conversor analógico-digital transforma o sinal elétrico em dados digitais. O computador lê os dados, processa e apresenta os resultados na tela do monitor. As informações deste sistema de monitoramento de temperatura são mostradas através de uma interface gráfica (**Figura 2b**) onde é possível visualizar a temperatura atual, comportamento estatístico, histórico da temperatura, histograma entre outros. O sistema permite também a configuração de diversos parâmetros de monitoramento que podem ser ajustados de acordo com o objetivo do acompanhamento.



(a) Diagrama de blocos do sistema.



(b) Painel de monitoramento.

**Figura 2** – Exemplo de sistema de monitoramento de temperatura de um tanque.

### 3. LabVIEW

Agora que já temos uma ideia do que é um Instrumento Virtual vamos agora conhecer como se faz para construir o nosso próprio Instrumento Virtual.

O LabVIEW é um ambiente de desenvolvimento de instrumentos virtuais. Foi construído de tal forma que um usuário que não seja um especialista em programação consiga implementar um software rapidamente, atendendo as suas necessidades de medições, análise e visualização de resultados. A programação é feita de uma forma visual através de desenhos, diagramas e conexões entre os blocos de funções tornando o processo de construção de software intuitivo.

#### 3.1 Componentes básicos

A Figura 3 mostra os componentes básicos do LabVIEW.



Figura 3 – Componentes do LabVIEW.

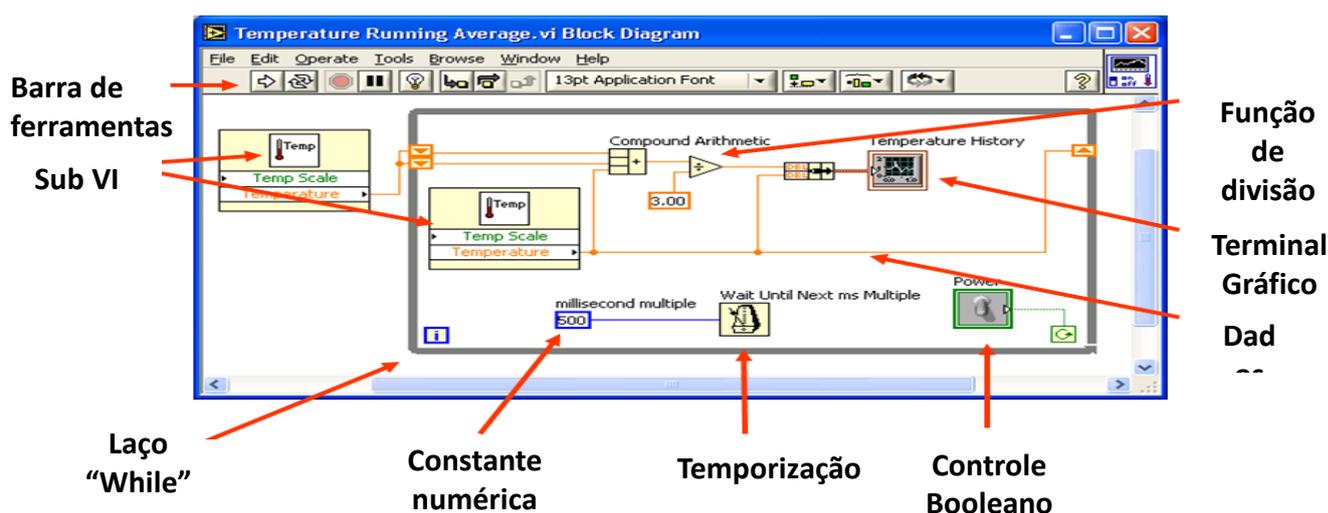
#### Programação G:

O LabVIEW utiliza uma linguagem de programação gráfica. Ao invés de linhas de códigos em texto, o programa é construído por meio de diagramas de blocos conectados por linhas. Trata-se de uma programação orientada pelo fluxo de dados<sup>7</sup> e baseada numa estrutura hierárquica. Permite a implementação de sistemas complexos que englobem aquisição e manipulação de dados ou ainda o controle de equipamentos por meio do computador.

Vamos ver como fica o programa para o caso do **Exemplo 1** baseado na linguagem de programação gráfica, utilizando LabVIEW como mostrado na Figura 4. Não se assuste, mais adiante, vamos falar passo-a-passo sobre cada um dos elementos que apareceram no diagrama de blocos. Repare no diagrama estão presentes vários elementos conhecidos na programação convencional. Existem dois blocos de temperatura (SubVI) sendo que um refere-se à

<sup>7</sup> Os blocos (operações ou funções) atuam sobre os dados assim que ficam disponíveis. O fluxo dos dados é distribuído graficamente através de fios que conectam a saída de um bloco na entrada do próximo.

temperatura atual e outro refere-se aos dados históricos de temperatura armazenados na memória. Observe também que há um laço de repetição (“While”) onde os dados são processados. Os dados “fluem” por meio dos fios vermelhos. A visualização do resultado é feita através de um terminal gráfico que gera a curva de temperatura automaticamente. Há também um controle do tipo booleano que tem a função de ativação/desativação do sistema de monitoramento. Procure imaginar como você escreveria um programa em C ou Java e compare com a programação G.



**Figura 4** – Programa elaborado em linguagem gráfica do LabVIEW referente ao sistema de monitoramento de temperatura de tanque (Exemplo 1). Por conveniência um programa desse tipo é chamado de VI. Um SubVI é um outro VI que é chamado dentro do programa (equivalente a uma subrotina).

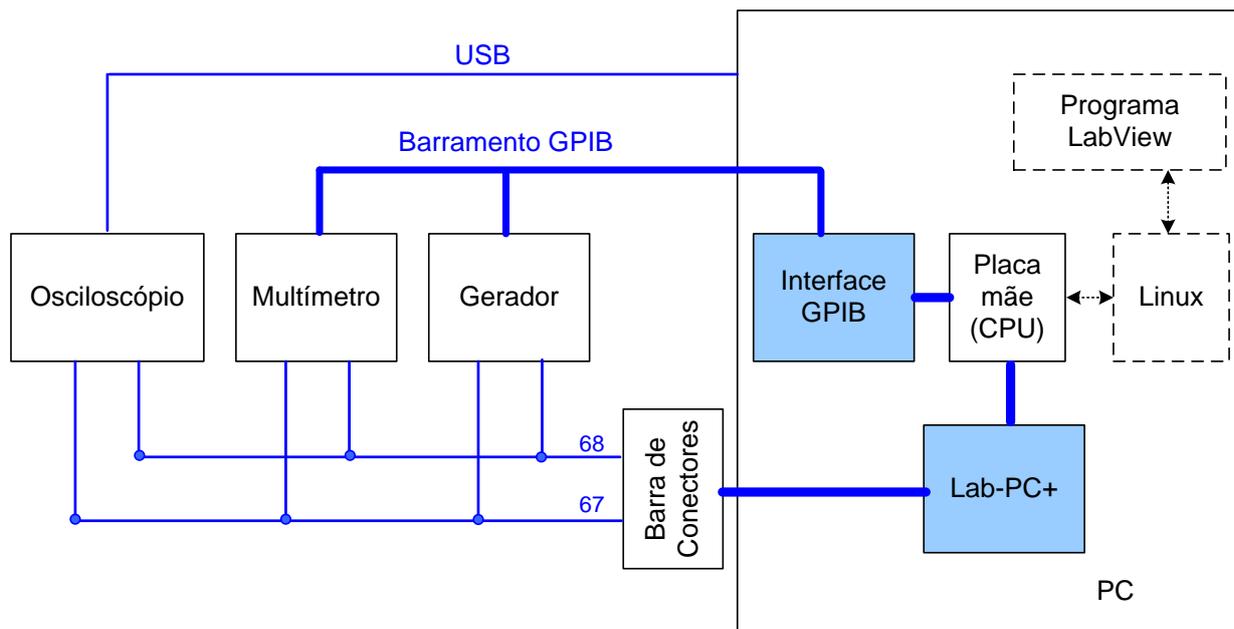
### Interface com o Hardware:

Uma das dificuldades dos programadores é realizar a comunicação entre o computador e os diferentes tipos de hardware. O LabVIEW possui um suporte para a maioria dos equipamentos e dispositivos utilizados em medições e instrumentação de uma forma geral, incluindo:

- Instrumentos científicos;
- Dispositivos para aquisição de dados;
- Sensores;
- Câmeras;
- Motores e atuadores.

Por exemplo, os instrumentos e o sistema de aquisição de dados (A/D) das bancadas do Laboratório de Instrumentação (C1-06) estão integrados ao computador conforme ilustrado na

Figura 5. A comunicação entre o PC e os equipamentos é feita por meios de *plugins*<sup>8</sup> e *device drivers*<sup>9</sup> que são incorporados ao LabVIEW.



**Figura 5** – Exemplo de integração de instrumentos e placa de aquisição de dados – bancadas do Laboratório de Instrumentação Elétrica (C1-06).

### Bibliotecas de código técnico e análise

O LabVIEW incorpora centenas de funções e algoritmos específicos necessários em aplicações científicas e de engenharia, incluindo as seguintes:

- Manipulação de strings, arrays e formas de onda;
- Processamento de sinal, incluindo filtros, janelamento, análise espectral, transformadas de Laplace, Transformada de Fourier e Transformada Z.
- Análise matemática, incluindo ajuste de curvatura, estatísticas, equações diferenciais, álgebra linear e interpolação;
- Comunicação, incluindo protocolos para comunicação em alto nível, HTTP, SMTP, FTP, TCP, UDP, Serial e Bluetooth;
- Geração de relatórios, E/S de arquivos e conectividade com banco de dados;
- Aplicações mais especializadas<sup>10</sup>, como:
  - Projeto e simulação de sistemas de controle
  - Análise de som e vibração
  - Visão de máquina e processamento de imagem
  - Rádio Frequência e comunicação

<sup>8</sup> Componente de software que permite que uma extensão de software seja incorporada ao software aplicativo.

<sup>9</sup> Componente de software que permite que um software aplicativo se comunique com um hardware por meio de chamadas de serviço. O programador não precisa ter conhecimento de detalhes do protocolo de comunicação do equipamento, pois o *device driver* faz o papel de tradutor.

<sup>10</sup> Pode depender de *Addons* complementares ao pacote de LabVIEW adquirido.

## Componentes e Ferramentas de Interface com Usuário e Relatórios

Ao mesmo tempo em que você constrói a lógica de operação do programa é elaborada a interface com o usuário numa janela chamada de “Painel Frontal”. Conforme mostrada na Figura 4, podem ser colocadas na interface com o usuário os controles interativos, como gráficos, relógios e tabelas, para visualizar seus dados adquiridos. Também são disponibilizadas ferramentas para salvar os dados em arquivos ou bancos de dados, ou gerar relatórios automaticamente.

## Abstração de tecnologia

Você pode desenvolver o seu software sem se preocupar com os detalhes do hardware final onde o programa feito em LabVIEW será executado. A plataforma de hardware pode ser uma CPU com múltiplos núcleos, um microcontrolador ou uma FPGA<sup>11</sup>. As complexidades envolvidas nas operações com o hardware serão absorvidas pelo LabVIEW.

## Modelos de computação

O LabVIEW permite selecionar a abordagem mais eficiente para solucionar os problemas computacionais. Os modelos de computação admitidos são os seguintes:

- Fluxo de dados é o modelo de computação padrão para o LabVIEW.
- Diagramas de estados fornecem uma abstração de nível mais elevado para aplicações baseadas em estados.
- Diagramas de simulação são um modo familiar de modelar e analisar sistemas dinâmicos.
- Formula Node insere fórmulas matemáticas simples em seu código de linguagem G.

## 3.2 Programação em LabVIEW

### Painel Frontal e Diagrama de Blocos:

Qualquer programa feito em LabVIEW é chamado de instrumento virtual (VI – *virtual instrument*) já que sua aparência e operação assemelham-se às de instrumentos reais. Um VI, assim como um programa usual, é composto por um conjunto de instruções que fazem a manipulação e fluxo dos dados, e por uma interface com o usuário, na qual se encontram as entradas e saídas necessárias.

Basicamente pode-se identificar em um VI duas partes que o compõem:

- *Diagrama de blocos* – é a estrutura do programa propriamente dita que contém o código fonte construído de forma gráfica;
- *Painel frontal* – constitui a interface com o usuário, que apresenta de forma visual todos os controles, gráficos e indicadores na forma de uma tela que simula o painel físico de um instrumento. Ele pode ser formado por botões, *leds*, *knobs* e indicadores que permitem a interação pelo *mouse* ou pelo teclado do computador.

---

<sup>11</sup> Field Programmable Gate Array – circuito integrado em que o usuário pode programar as funções lógicas do hardware de forma a atender as suas necessidades específicas.

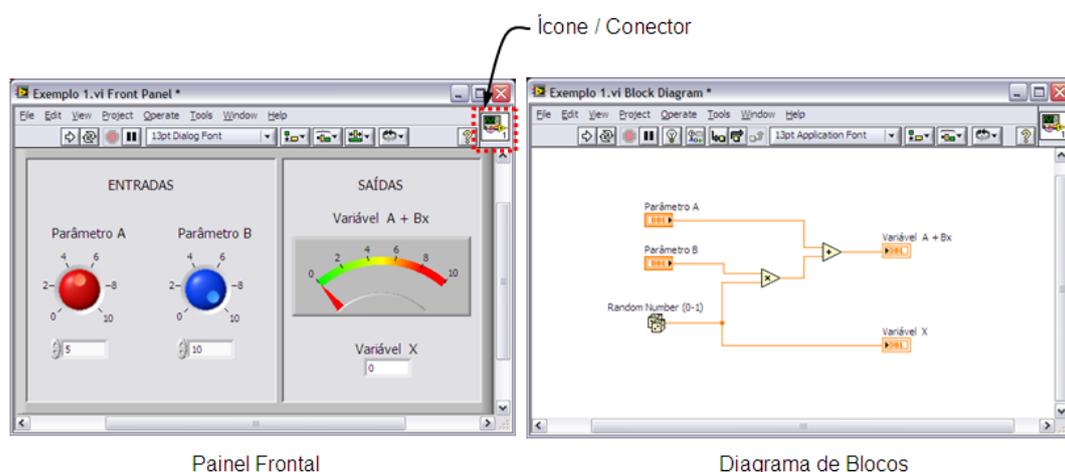
Pode-se ainda encapsular um VI inteiro (isto é, diagrama de blocos + painel frontal) em um módulo re-utilizável dentro de um outro VI. Esse módulo encapsulado constituirá um subVI. Assim sendo, podemos incluir um terceiro ingrediente componente de um VI genérico:

- *Ícone/conector* – define as entradas e saídas do VI acessíveis à conexão quando este é utilizado como um subVI dentro de um outro VI (é análogo às definições de entrada e saída para se usar uma rotina como função dentro de um programa numa linguagem convencional). O ícone é a definição da aparência gráfica que se deseja que este VI tenha no diagrama de blocos quando usado como um subVI.

### Exemplo 2:

Para entender como funciona a programação em LabVIEW vamos fazer um programa simples. Considere que as entradas sejam os parâmetros **A** e **B**, números inteiros que podem variar de 0 a 10. As saídas são as variáveis **X** e **A + BX** sendo que o parâmetro **B** deve ser multiplicado por um número aleatório (real) variando na faixa de **0** a **1**. O software deve permitir que usuário selecione o valor dos parâmetros **A** e **B**. O resultado, **A + BX**, deve ser mostrado em um visualizador analógico do tipo ponteiro, com fundo de escala igual a **10**. O valor da variável **X** deve ser mostrado num campo numérico. Como seria um programa que atenda essa especificação em C, Java ou Python?

Fazer esse programa em LabVIEW é muito simples como pode ser visto na Figura 6. A janela do lado esquerdo é chamada de “**Painel Frontal**”, onde o usuário irá interagir com o programa. Uma vez que o VI estiver pronto essa será a parte visível do programa. A janela do lado direito é o “**Diagrama de Blocos**”, onde se encontra a parte lógica do VI, que em geral fica oculto para o usuário. Durante a fase de programação trabalha-se concomitantemente em ambas janelas, que estão completamente integradas. Modificações feitas em uma janela são refletidas imediatamente na outra janela.



**Figura 6** – Exemplo de programação em LabVIEW.

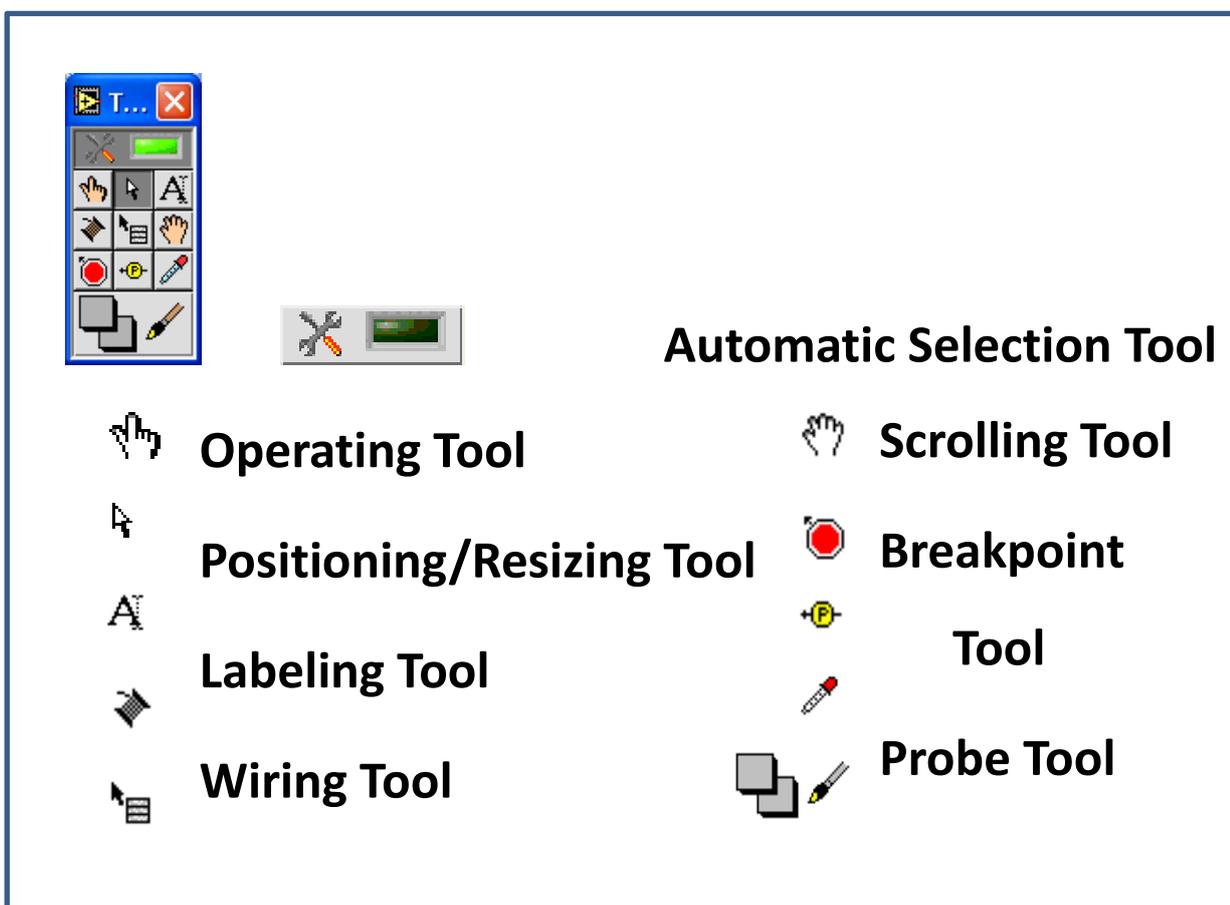
## **Pallets<sup>12</sup>:**

Para a edição do programa como mostrado na Figura 6 precisamos conhecer os recursos dos chamados de *palettes*. São três tipos:

- *Tools palette* (disponível em ambas janelas)
- *Controls palette* (disponível somente na janela do painel frontal)
- *Functions palette* (disponível somente na janela do diagrama de blocos)

### *Tools palette:*

Para abrir selecione na barra de ferramentas: **View >> Tool Pallet.**



**Figura 7** – Tools pallet - ferramentas de uso geral para operar e modificar os objetos do painel frontal e diagrama de blocos.

### *Control Palette:*

Para abrir selecione na barra de ferramentas: **View >> Control Pallet** (ou também com o botão direito do mouse na área de trabalho do painel frontal). A Figura 8 mostra um exemplo de seleção

<sup>12</sup> Paletas em português. Por uma questão de conveniência utilizaremos toda nomenclatura de programação em inglês apenas, com tradução caso necessário.

dessa funcionalidade. Veja que uma grande variedade de objetos de controle e de visualização são disponibilizados. Há muitas possibilidades de escolha para construir o seu instrumento virtual.

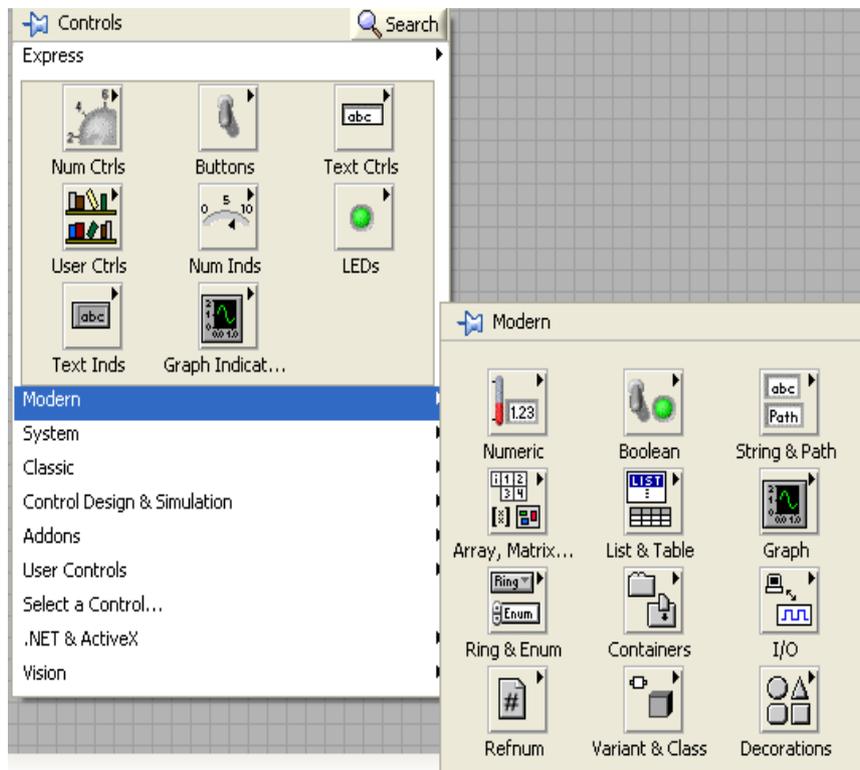


Figura 8 – Control Pallet – Objetos de controle e visualização são disponibilizados. Há muitas possibilidades de escolha para construir o seu instrumento virtual.

#### *Function Palette:*

Para abrir selecione na barra de ferramentas: **View >> Functions Pallet** (ou também com o botão direito do mouse na área de trabalho do diagrama de blocos). A Figura 9 mostra um exemplo de seleção dessa funcionalidade. Veja que uma grande variedade de objetos de funções são disponibilizadas.

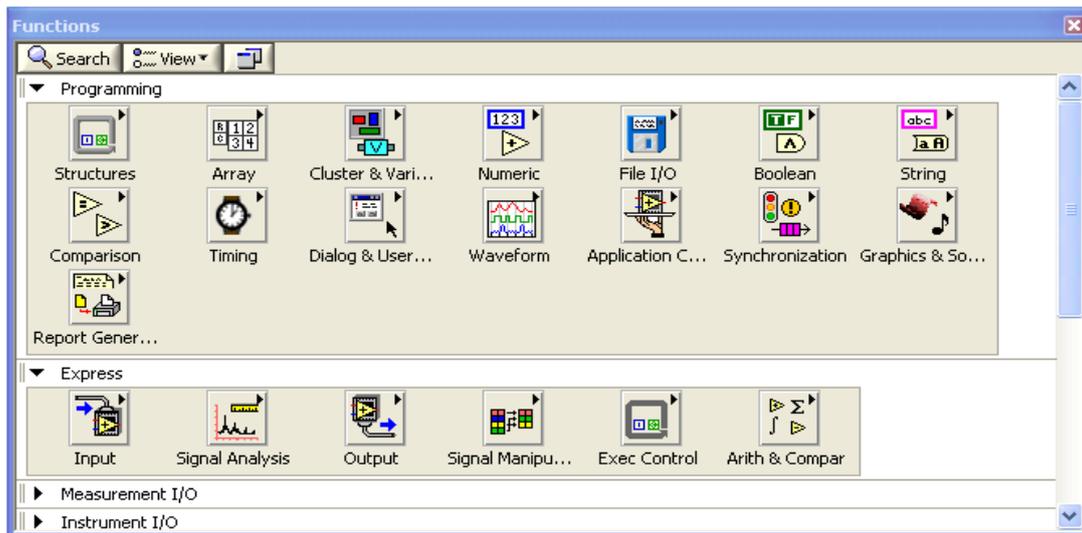


Figura 9 – Function Pallet – Objetos de funções por categorias. Ao clicar um determinado ícone será aberta outra janela com o leque opções.

### Context Help (Ajuda Contextual)

É um recurso muito útil para entender o funcionamento dos objetos selecionados. Ao apontar para qualquer objeto (ícone) e selecionar essa função (“show context help”) serão apresentadas detalhes operacionais desse objeto (exemplo na Figura 10).

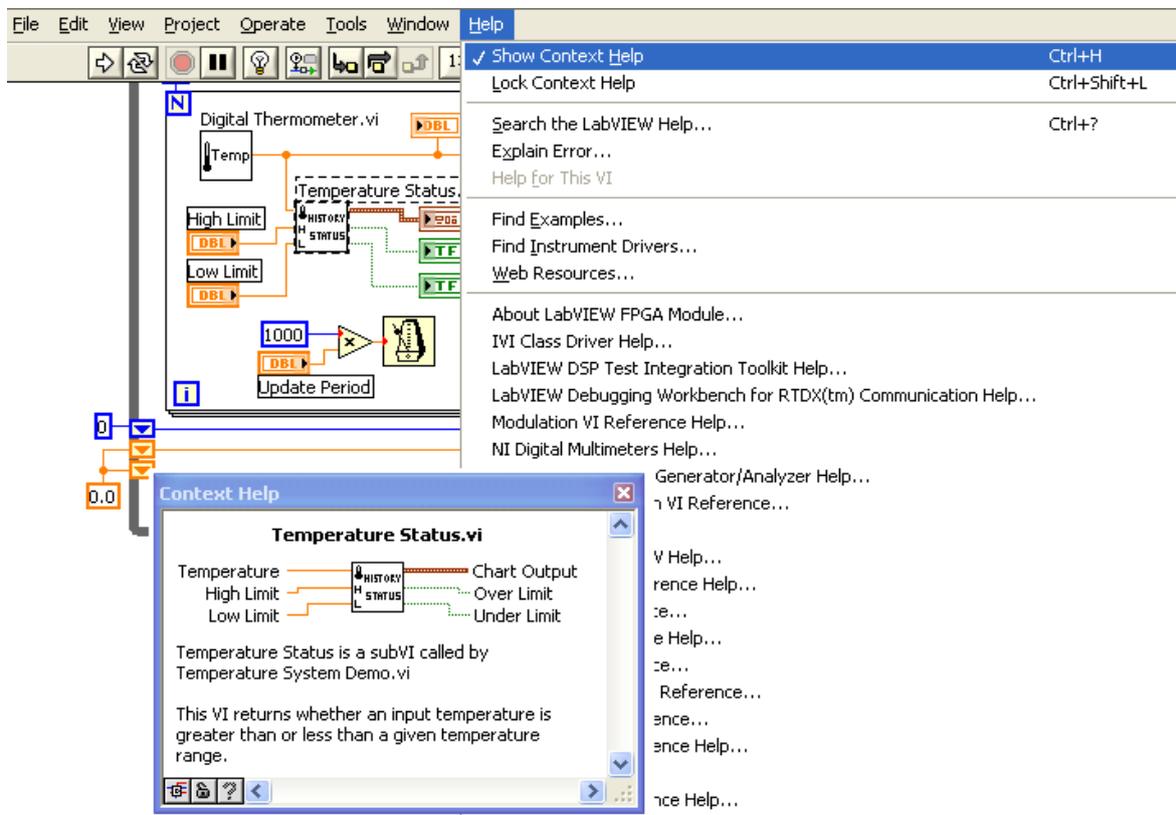


Figura 10 – Ajuda Contextual.

## Barra de Estados

Terminou a programação? Então chegou a hora de testar o programa. As funções relacionadas com a execução, parada e o estado do programa encontram-se na “Barra de Estados”.

