



## Tutorial 1: Introdução ao Ambiente de Programação LabVIEW

Rodrigo dos Anjos Souza, Elisabete Galeazzo

### 1 Teoria I:

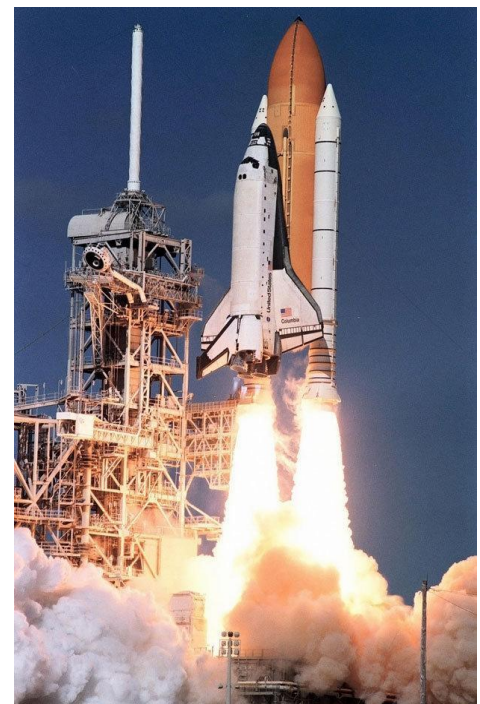
#### 1.1 OBJETIVOS:

- Familiarização com o conceito de Instrumentação Virtual
- Compreensão das principais ferramentas do LabVIEW
- Familiarização com as ferramentas de depuração do LabVIEW

#### 1.2 VISÃO GERAL:

##### 1.2.1 História:

Na década de 1980, o programa espacial americano encontrava-se em um período de grande avanço com o desenvolvimento da nave Columbia. Com a enorme quantidade de sensores e computadores operando na nave, os cientistas e engenheiros da NASA se depararam com o seguinte problema: como tornar a programação de sistemas de grande complexidade e, principalmente, a validação de suas subestruturas, mais eficientes? Dentre as soluções encontradas na época, destacou-se o LabVIEW (**L**aboratory **V**irtual **I**nstrument **E**ngineering **W**orkbench) oferecido pela recém-criada National Instruments (NI). A plataforma do LabVIEW, diferente de linguagens sequenciais (como a linguagem C), foca no fluxo de dados utilizados em um sistema e não a sequência de operações lógicas que um computador deverá seguir.



### 1.2.2 Instrumentação Virtual:

Atualmente, uma parte significativa das atividades de engenheiros e de cientistas envolve testes e ensaios baseados em medições de diversos parâmetros. Atividades como lançar um foguete, um avião ou testar um carro, por exemplo, requerem inúmeros ensaios controlados para certificação e homologação junto a órgãos como AEB, DENATRAN e ANAC. Mesmo em atividades didáticas ou projetos de alunos de graduação, é comum a coleta e a análise de grande quantidade de dados para avaliar o desempenho do sistema em estudo, como por exemplo a resposta em frequência de um circuito eletrônico. Deste modo, foram criadas ferramentas de instrumentação virtual, dentre as quais se destaca o LabVIEW, que reduzem o tempo de coleta de dados, análise e visualização de informações.

### 1.2.3 LabVIEW na disciplina PSI 3214:

Ao longo da disciplina de PSI 3214 será realizado um projeto extraclasse que envolverá a aquisição de dados (através de sensores quando possível, ou através de seleção de banco de dados) e um computador para processamento e visualização de tais informações. Assim, é importante saber como é a estrutura geral de um projeto que efetua a aquisição e armazenamento de dados de um processo, e como o LabVIEW se encaixa nesse panorama. Neste contexto, o projeto deve ter o seu fluxo de dados dividido em 4 partes de interesse: **Processo** em estudo, **Aquisição** de dados (por exemplo, a aquisição poderia ser feita com microcontroladores e conversores AD, se fosse possível realizá-los), **Análise** dos dados, **Visualização** de dados, como indicado na Figura 2.

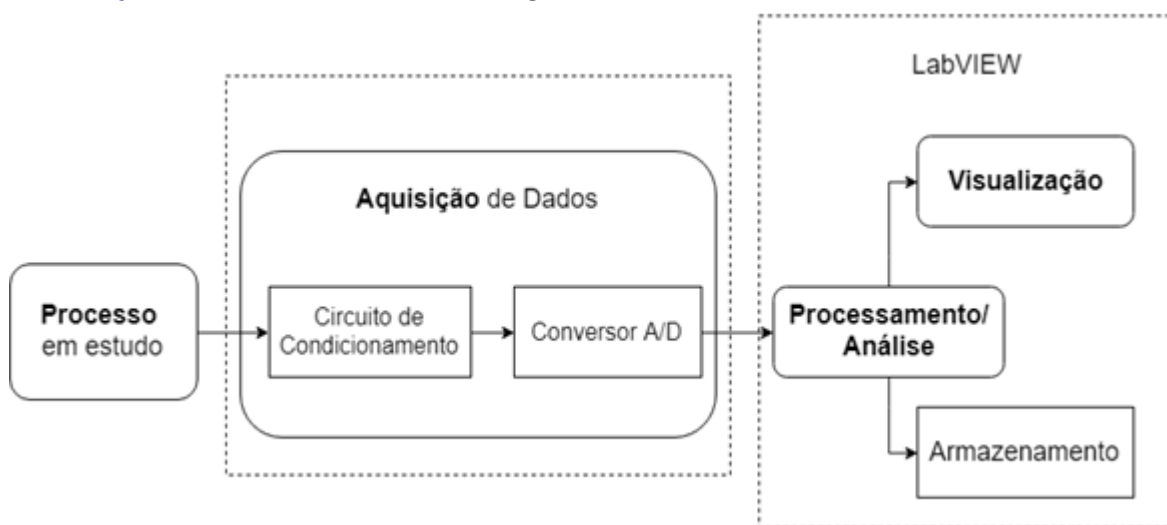


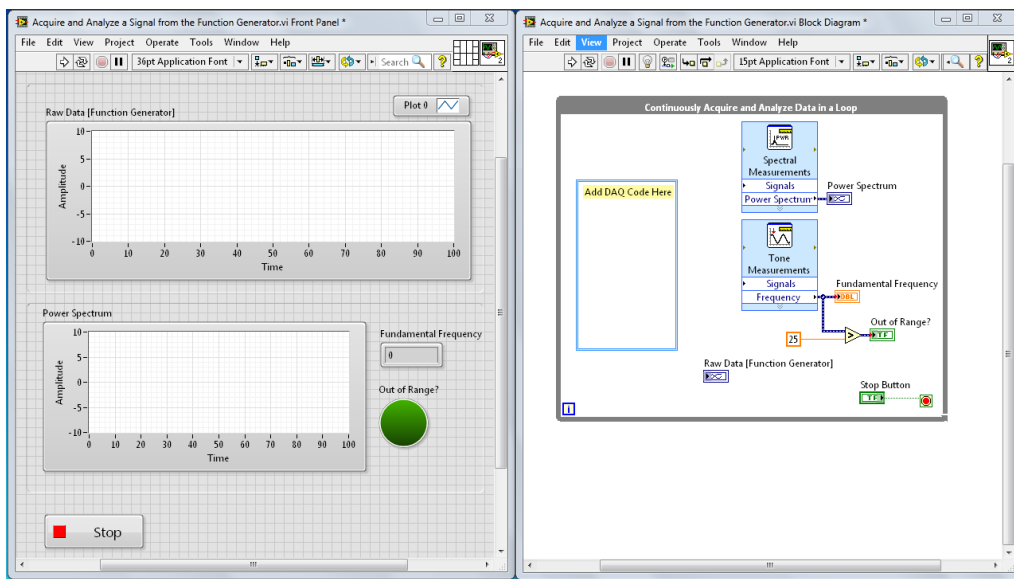
Figura 2 – Diagrama de blocos do projeto.

O projeto que será desenvolvido durante a disciplina em 2020 será parcialmente baseado nas etapas indicadas na Figura 2 (neste ano não teremos a aquisição física dos dados com sensores). Em tais etapas, os grupos de trabalho desenvolverão e aplicarão suas capacidades de análise de circuitos, bem como de programação em LabVIEW para visualização e processamento dos dados.

### 1.2.4 O que é LabVIEW?

O LabVIEW é uma ferramenta de desenvolvimento que faz uso de programação gráfica (denominada **G**) para criação de instrumentos virtuais (**Virtual Instruments, VIs**). Tais instrumentos se comunicam com hardwares específicos da National Instruments ou que possuem suporte dela, com o intuito de acelerar os processos de aquisição, análise e controle de sistemas.

Ele apresenta um ambiente de programação contendo dois painéis com funcionalidades diferentes. Um deles é denominado Painel Frontal (**Front Panel**), que é a interface com o usuário, e o outro denominado Diagrama de Blocos, onde é realizada a programação (**Block Diagram**), como ilustrado a seguir.



### 1.2.5 Primeiros passos:

Ao abrir o LabVIEW note que nos deparamos com a tela ao lado. Nela podemos criar um Instrumento Virtual (ou um VI), ou um projeto, ou seja, um conglomerado de VI's e outros arquivos auxiliares utilizados para programação de Instrumentos virtuais mais complexos.

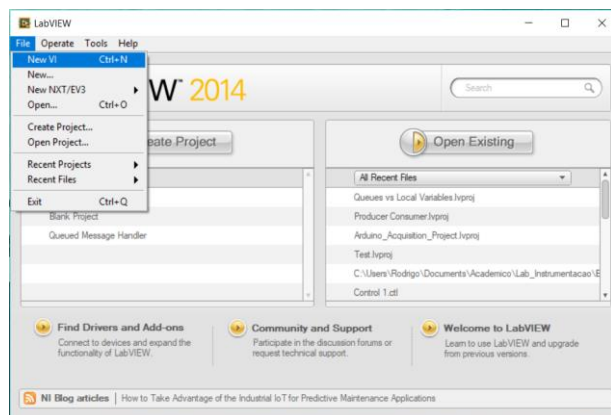


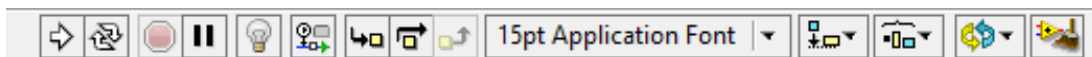
Figura 3 - Tela inicial do LabVIEW

### 1.3 MODOS DE EDIÇÃO:

Como já mencionado, ao criar um novo VI (Figura 3) observe que serão abertas duas janelas, uma delas denominada **Painel Frontal**, ambiente onde é programada a **interface de usuário**, e outra janela denominada **Diagrama de Blocos**, onde toda a programação relacionada à **análise e ao processamento** de dados será realizada. Todos os elementos criados no **Painel Frontal** possuem referências no **Diagrama de Blocos**, pelas quais se **lê um valor** fornecido pelo usuário (**Controle**) ou se **mostra um resultado** para o usuário (**Indicador**). É importante lembrar-se dessa divisão entre as duas janelas, uma vez que a quantidade de ferramentas que podem ser utilizadas pelo usuário é muito grande. Por exemplo, se desejarmos realizar uma operação sobre os dados adquiridos para montagem de um vetor de valores espectrais, devemos ir ao **Diagrama de Blocos**, pois neste caso estamos operando sobre os dados. Se desejarmos apresentar tais informações espectrais a algum usuário, devemos ir ao **Painel Frontal**. Para maior compreensão, cada janela se encontra descrita em detalhes a seguir:

#### 1.3.1 Diagrama de Blocos:

- **Barra de Ferramentas:** A barra de ferramentas é acessível no topo da janela e possui as funções básicas de execução e depuração do programa, indicadas abaixo.



- **Ferramentas de Execução:**



**Compilar e Executar o Programa:** Compila o programa e o executa uma vez. Este será o principal método para se executar o programa.



**Compilar e Executar Continuamente o Programa:** Compila o programa uma vez e o executa continuamente até que se encontre algum erro ou que seja abortado. Quase nunca utilizado.



**Abortar o Programa:** Aborta quaisquer operações sendo realizadas pelo programa. Tal botão é apenas emergencial e **não deve ser utilizado** para se parar o programa. Seu uso indevido e contínuo pode levar vazamentos de memória e até a perda de dados previamente salvos.

- **Ferramentas de Depuração:**



**Pausar o Programa:** Pausa o fluxo de dados pelo programa. Tal ferramenta pode ser usada para se averiguar valores de fios e avaliar se estão corretos ou não. Junto às ferramentas de passo e de sondagem, é um poderoso meio de depuração.



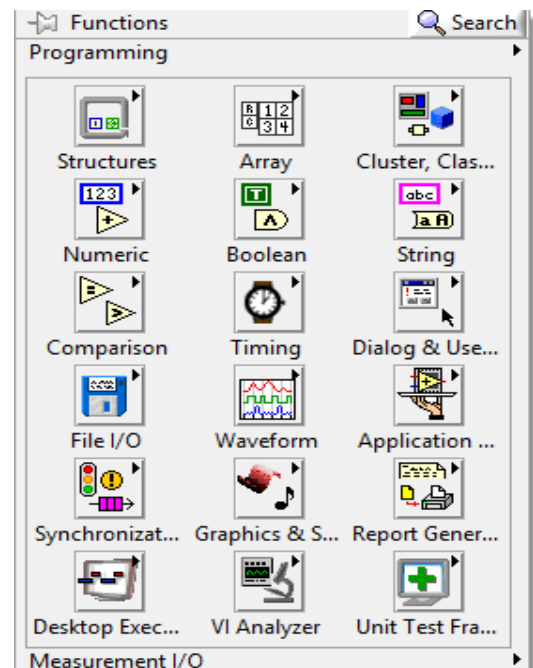
**Realce da Execução:** Realça os conectores do programa, de modo que se pode visualizar o fluxo de dados, permitindo a identificar a origem de um erro ou de um valor incorreto retornado pela execução do código. Este é o meio mais intuitivo e fácil de depurar um programa em LabVIEW.

- **Paleta de Funções\*:**

A paleta de funções é o espaço no qual se encontram **todas** as funções de programação oferecidas pelo LabVIEW. Nela existem alguns blocos que valem menção aqui por sua utilização:

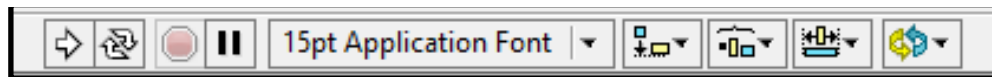
- Estruturas: **Laço For, Laço While;**
- Vetores: **Criação de Vetores, Manipulação de Vetores;**
- Numérico: **Manipulação algébrica;**
- Boolean: **Manipulação de Boolean;**
- Strings: **Manipulação de Strings;**
- Comparação: **Operações lógicas generalizadas;**

\* Para abrir a **Paleta de Funções**, basta clicar com o **botão direito** do mouse em qualquer lugar da janela do **Diagrama de Blocos**.



### 1.3.2 Painel Frontal:

- **Barra de Ferramentas:** Assim como na janela do **Diagrama de Blocos**, o **Painel Frontal** também possui uma barra de ferramentas na parte superior da janela.



É interessante reparar que esta **não possui ferramentas de depuração**, já que tais opções dizem respeito à integridade dos dados. Assim, se mantém a divisão de que o **Painel Frontal** diz respeito somente à interface de usuário.

- **Paleta de Controle\*:** A paleta de controles contém quaisquer elementos que digam respeito à interface de usuário. Nela, além de controles específicos de robótica e visão computacional, há 4 estilos de interface que podem ser utilizadas: *Modern*, *Silver*, *System* e *Classic*. Entre essas, a *Silver* é a mais utilizada em programas modernos. Os blocos de destaque desta paleta são ilustrados na Figura 4 e destacados na sequência:

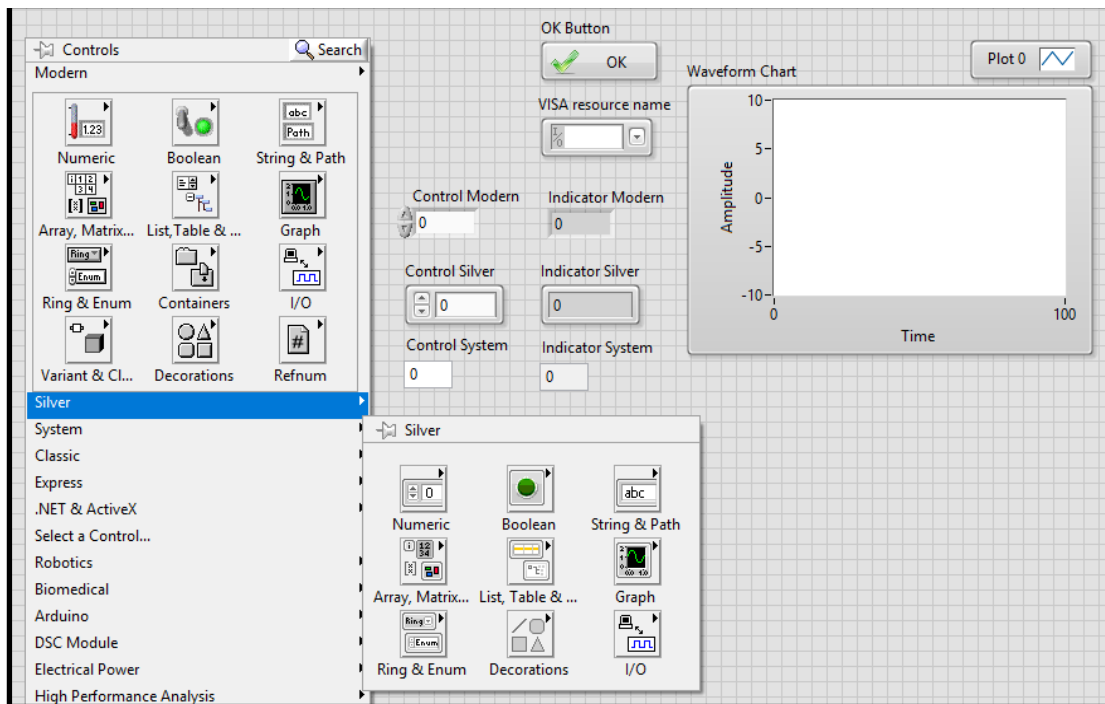


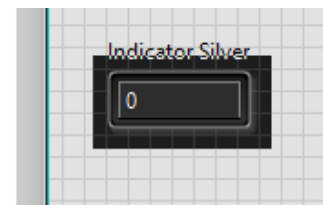
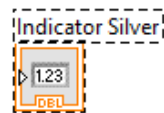
Figura 4: Blocos de destaque da paleta de controle “Silver”.

- **Numérico:** Controles e Indicadores numéricos.
- **Boolean:** Controles e Indicadores booleanos.
- **String & Path:** Controles e Indicadores de Strings e Paths (caminhos de arquivos)
- **Gráficos:** Gráficos (Waveform Graphs), Mapas (Waveform Charts), Gráficos XY
- **I/O:** Portas de Saída/Entrada Seriais (VISA Resource Name)

\*Para abrir a **Paleta de Controles**, basta clicar com o **botão direito** do mouse em qualquer lugar da janela do **Painel Frontal**.

### 1.3.3 Recursos Compartilhados entre as Duas Janelas:

- **Seleção de Objetos:** Muitas vezes, em programas de alta complexidade, perde-se tempo procurando, por exemplo, a referência do **Diagrama de Blocos** de um indicador ou controle criado no **Painel Frontal**. Para tal, basta dar um **click duplo** em um elemento em qualquer uma das janelas para que sua referência na outra janela seja indicada, como na figura ao lado.



- **Alternar entre janelas:** Para ir do **Diagrama de Blocos** para o **Painel Frontal** e vice-versa, é possível utilizar o atalho **Ctrl + E**.

- **Paleta de Ferramentas (Shift + Botão Direito)**

A paleta de ferramentas está disponível nas duas janelas do LabVIEW, com as seguintes funções:



**Seleção Automática:** Seleciona automaticamente qual a melhor ferramenta para utilização.



**Operar Valor:** Muda o valor de um botão, controle, gráfico ou qualquer outra opção de um elemento do Painel Frontal em tempo de programação.



**Posicionamento/Tamanho/Seleção:** Muda a posição e o tamanho de um objeto.



**Editar Texto:** Edita o texto de botões, gráficos e outros elementos, tanto do Painel Frontal, quanto do Diagrama de Blocos.



**Conectar Fios:** Cria os fios que transmitem os dados entre um bloco e outro do Diagrama de Blocos.



**Posicionamento da Janela:** Permite o reposicionamento da janela, seja o Painel Frontal, seja o Diagrama de Blocos, em relação a seus elementos.



**Criar/Limpar Breakpoint:** Cria ou limpa um breakpoint. Quando o fluxo de um programa passa por um breakpoint, a execução do programa é pausada.



**Sondagem de Dados:** Cria uma sonda no fio desejado. Quando o fluxo de dados passar pelo fio, o valor da estrutura sondada será atualizada na janela de Visualização de Sondas (Probe Watch Window).



**Amostrar Cor:** Amostra a cor de um elemento do Painel Frontal e a guarda na paleta de cores.

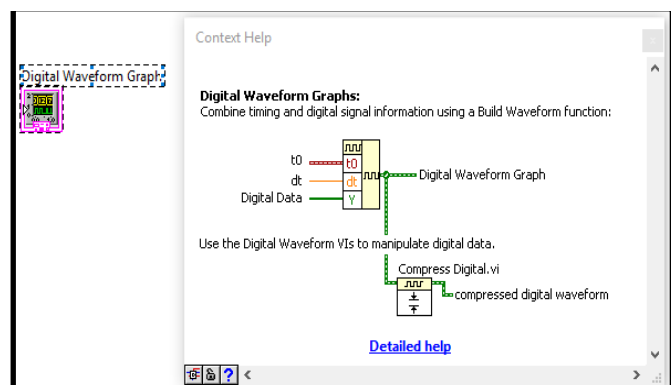


**Colorir:** Aplica as cores selecionadas ao elemento com o botão esquerdo, abre a paleta de cores com o botão direito.



- **Ajuda contextual (Ctrl + H)**

A ajuda contextual apresenta todas as informações básicas do elemento sobre o qual o mouse para, incluindo sua utilização e as entradas e saídas de blocos.

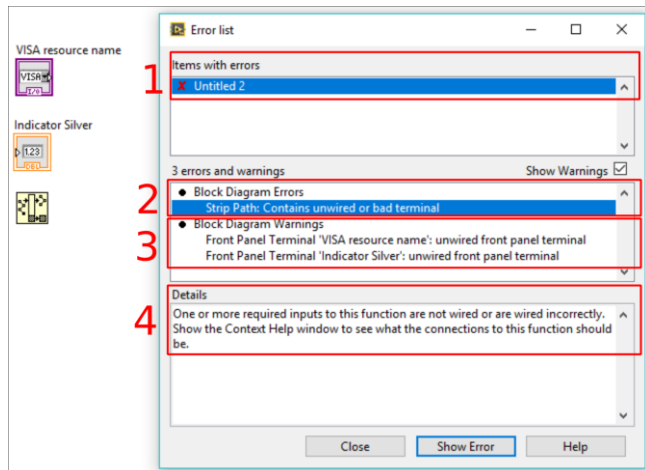


## 1.4 UTILIZANDO AS FERRAMENTAS DE DEPURAÇÃO

### 1.4.1 Para que depurar um programa?

Em muitas ocasiões, escreve-se um programa e percebe-se que existem erros ao executá-lo. Assim como nas linguagens compiladas, como C, existem dois tipos de erros possíveis em um programa:

**a) Erros associados ao tempo de Compilação:** ocorre quando há algum erro na escrita do programa. Um exemplo deste tipo de erro é quando se associa operações conjuntas entre estruturas de dados diferentes: multiplicação de uma string por um float. Nestes casos, o compilador dá um aviso ao programador antes da execução do programa. Para o caso do LabVIEW, o botão de execução fica “quebrado”, e quando clicamos nele aparece na tela uma lista de erros identificados pelo compilador, como o exemplificado ao lado.



**Sendo o significado de cada item da Error List:**

- (1) Itens com Erro:** Pode ser um VI ou outras estruturas utilizadas pelo LabVIEW.
- (2) Lista de Erros:** Listagem dos erros encontrados na estrutura selecionada.
- (3) Lista de Avisos:** Listagem dos avisos encontrados na estrutura selecionada.
- (4) Detalhes:** Detalhamento de onde está o erro, e porque ele ocorre.

**b) Erros em tempo de Execução:** Ocorre quando há algum erro na lógica do programa. Por exemplo, quando se realiza uma operação logicamente inválida: divisão por termo nulo. Nesses casos, não há aviso prévio do compilador, e cabe ao programador a depuração do programa.

### 1.4.2 Ferramentas de Depuração

**Visualização do Fluxo de Dados:** Dentre os métodos de depuração de um programa de LabVIEW, este se mostra ser o mais prático. Ao apertar o botão de **Realce da Execução**, cada passo tomado pelo computador para a execução do programa adquire uma representação visual. Os fios pelos quais o fluxo de dados passou tornam-se de cor sólida, e os blocos apresentam os dados que passaram por ele, enquanto que por onde o fluxo ainda não passou, os fios apresentam-se pontilhados.