



# Instrumentação Virtual

Rodrigo dos Anjos Souza, Elisabete Galeazzo

## Tutorial 1B

---

### Estruturas de Dados com o LabVIEW

---

#### 1. Teoria II:

##### 1.1 Objetivos:

- i. Compreensão das estruturas de dados, sua criação e manipulação;
- ii. Familiarização com o conceito de Laços aplicados ao ambiente LabVIEW;
- iii. Familiarização com a criação de gráficos.

##### 1.2 Estrutura de Dados Unidimensionais:

###### 1.2.1 Código de Cores:

No LabVIEW tipos de dados diferentes possuem códigos de cores diferentes.

Dentre os mais utilizados, encontram-se:



Inteiros [**Numeric**]: Esta classe engloba números de representação inteira, sejam 8 bits ou 64 bits.



Pontos flutuantes [**Double**]: Cobre números de representação flutuante, isto é, que possuem mantissa, uma base e um expoente.



Vetor de Caracteres [**String**]: Vetor de caracteres.

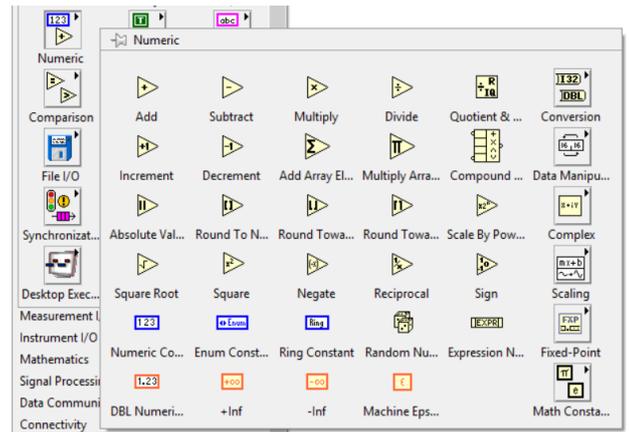


Boolean [**Bool**]: Representações binárias de verdadeiro e falso.

### 1.2.2 Manipulação Algébrica:

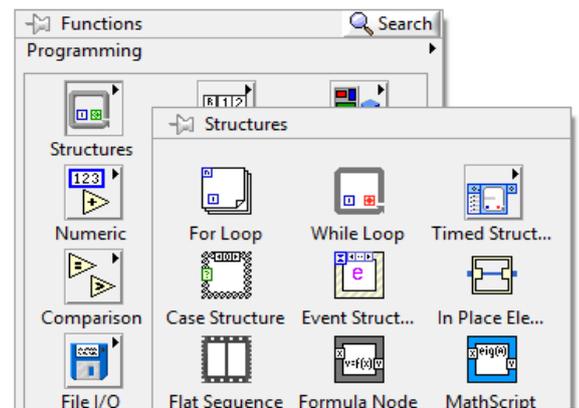
Assim como em outros programas, o LabVIEW possui um conjunto de **funções** para que o programador possa manipular as estruturas de dados mais simples.

Tais funções podem ser acessadas por meio da **Paleta de Funções** do **Diagrama de Blocos**.



#### i. Laços:

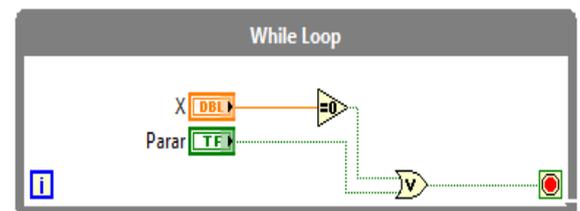
Dentro do ambiente do LabVIEW, os laços são muito utilizados para determinação do **Fluxo de Dados**, bem como para determinação de vetores. Tais estruturas, dentre outras importantes, podem ser encontradas dentro do **menu de Estruturas (Structures)** da **Paleta de Funções** do Diagrama de Blocos.



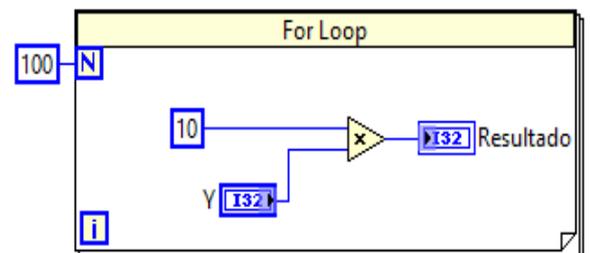
#### ii. Laços While:

Laços While executam o conjunto de tarefas dentro de sua estrutura até que uma **condição de parada** seja alcançada. O programa ao lado, mostra uma **estrutura While** que é executada caso o número **X** seja 0 ou o botão **Parar** seja verdadeiro.

*desafio 1: você consegue elaborar o programa indicado ao lado no LabVIEW e entender como ele funciona?*

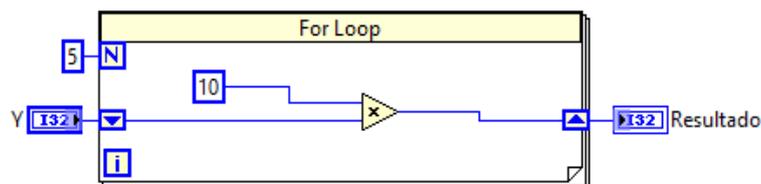
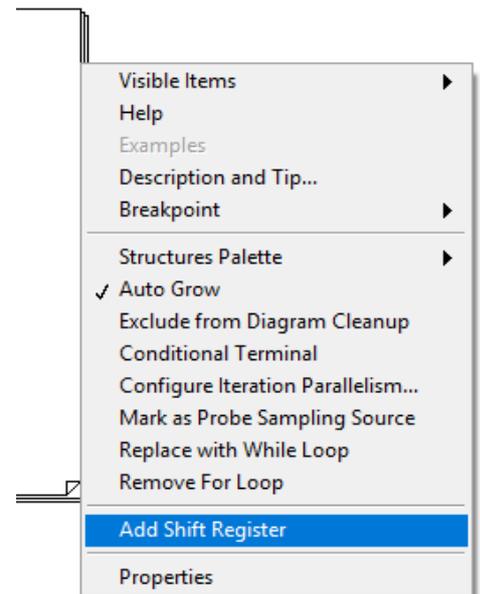


iii. **Laços For:** Semelhante à estrutura **While**, os Laços **For** executam um conjunto de instruções um **número N de vezes**, determinado pelo usuário. No programa ao lado, o laço é executado 100 vezes e nele o número **Y**, definido pelo usuário, é multiplicado por 10.



**Desafio 2:** Monte o programa acima, ative a função de **Realce de Execução** e o execute. Veja o que ocorre!

iv. **Shift Registers:** Caso tenha montado e executado o programa do desafio 2, terá percebido algo meio sem graça: a **mesma conta** é realizada a cada laço. Porém, na maioria das aplicações de laços, não queremos que uma mesma conta seja realizada **N** vezes. Ao invés disso, gostaríamos de poder **utilizar contas previamente realizadas** em um loop para uma próxima iteração, por exemplo. Aplicações como **Controle Digital** e **Filtros Digitais** fazem uso de tais técnicas. Para solucionar tal problema, foram concebidos os **SHIFT REGISTERS** em LabVIEW. Com eles é possível transmitir dados **do fim de um loop para o início do próximo**. Para criar um par de Shift Registers basta clicar com o **botão direito** na **borda do Laço For ou Laço While** e selecionar **Add Shift Registers**, como na figura ao lado.



Assim, podemos criar programas como o da figura acima, onde inicialmente o é utilizado o valor Y para ser multiplicado por 10, porém, nos loops subsequentes é utilizado o **resultado do loop anterior** para efetuar-se a próxima operação. Depois de todas as iterações, o resultado final é transmitido para o **Indicador Resultado**. Tente montar este programa e testá-lo com **Realce de Execução**.

### 1.2.3 Estruturas de Dados Multidimensionais:

v. **Código de Espessuras:** Assim como o código de cores, que diferencia dados de naturezas distintas, o código de espessuras distingue estruturas de dados mais complexas por meio da espessura dos fios. Dentre elas temos:

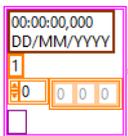
- Vetores [**Array**]: São estruturas complexas que são compostas de apenas um tipo de estrutura simples, como um inteiro ou uma string. São estruturas geralmente utilizadas para aquisição de dados. No ambiente do LabVIEW, os vetores são estruturas que



podem ter mais de uma dimensão, formando Matrizes e até outras estruturas multidimensionais. Note que **os fios de vetores são mais espessos do que os fios das estruturas de que é composto.**



- Aglomerado [**Cluster**]: Pode incorporar todos os outros tipos de estruturas de dados mais simples. Apesar de não ter um espaço devido neste documento, os *Aglomerados* são estruturas muito importantes para organização e separação de vias de dados com propósitos distintos. **Possuem fios espessos como os de vetores, porém sua cor é marrom.**



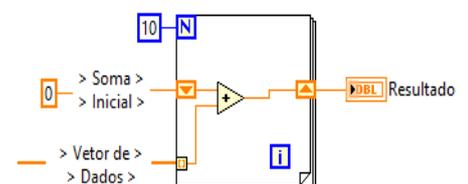
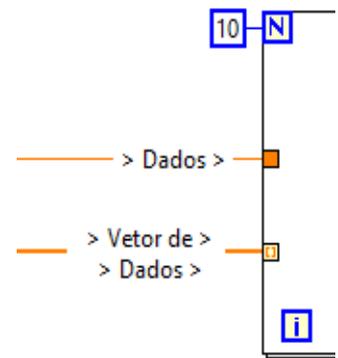
- Formas de Onda [**Waveform**]: As formas de onda são estruturas que possuem um **t(0)**, **dt**, **Y**, e uma classe abstrata chamada de atributos da onda. Esta estrutura é utilizada para organização de dados no tempo. Possuem **fios iguais aos de Clusters, porém de cor rosa.**

#### 1.2.4 Vetores Numéricos:

- **Inicialização:** Para se inicializar um *Vetor Numérico*, é necessário criar uma constante de vetor e uma constante numérica e arrastar a constante numérica para dentro da constante de vetor. Tal constante será o valor na posição indicada pela constante de vetor ("1" no caso da figura ao lado). A inicialização de um vetor se faz necessária no LabVIEW pois é obrigatório que se explicita para o compilador qual o tipo de vetor (Inteiros, Float, etc).

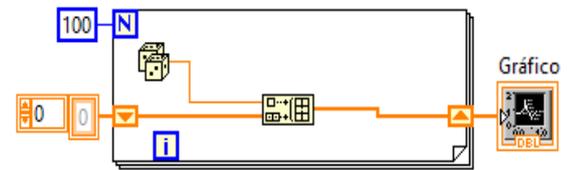


- **Túneis:** Antes de se ver como popular um vetor ao longo da execução de um programa, é necessário aprender o que são túneis e como é seu funcionamento básico. Túneis são caminhos pelos quais os **dados entram ou saem de laços**, e são criados automaticamente quando um fio entra em uma estrutura como um laço. Como na figura ao lado, a princípio existem dois tipos básicos de túneis. O primeiro trata **estruturas simples** e passa para dentro do laço os "**Dados**" como estão. Porém, quando se tem um "**Vetor de Dados**", o túnel pode ou não ser configurado para **indexação**. Ou seja, quando o túnel está em modo de **indexação**, repassa para dentro do Laço os **elementos do Vetor** um por um. No programa ao



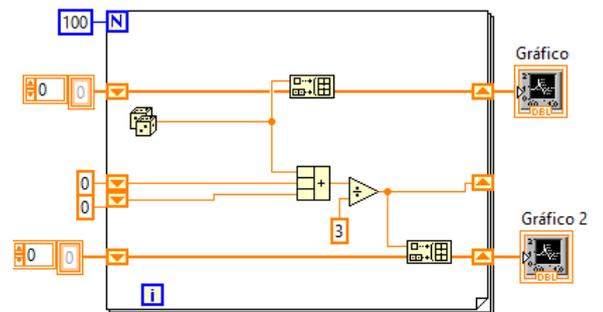
lado, por exemplo, faz-se o uso de **Shift Registers** e da **configuração de Túneis** para realizar-se a soma dos 10 primeiros elementos de um Vetor.

- **População:** Tomando-se o conceito acima, podemos utilizar Shift Registers e a função Build Array\* para se popular um vetor a partir de um vetor inicial (ou seja, dentro do laço adicionaremos elementos ao vetor). Veremos melhor tal processo na parte prática II, a seguir.



*\*dica: utilize o recurso ctrl-H sobre a função para compreender melhor o que ela realiza!*

- **Manipulação:** Por fim, utilizando-se de todos os conceitos acima, é possível se criar estruturas simples de manipulação de vetores. Na figura ao lado, tem-se um exemplo de tal aplicação. Na entrada do Laço temos dois vetores que serão populados: um com números gerados randomicamente, e outro com a média móvel desses valores. Note-se que ao **selecionar o Shift Register e expandi-lo criar-se múltiplos Shift Registers (ou empilhá-los)**, com isso é possível verificar, por exemplo, dados de iterações passadas do laço. Veja a figura ao lado e tente compreender como o programa funciona!



veja o vídeo

[http://www.ni.com/video/app/play\\_doc/p/id/2477/lang/pt/width/800/height/640/autostart/y/popup/y](http://www.ni.com/video/app/play_doc/p/id/2477/lang/pt/width/800/height/640/autostart/y/popup/y)

para entender como é possível expandir o shift register e qual é a sua utilidade para calcular a média móvel.

Abra o arquivo **media\_movel\_Pedro\_Ligabue.vi** para visualizar outro exemplo que efetua a média móvel.