


**Variáveis de Fluxo** permitem a parametrização de um workflow. Uma variável de fluxo é um parâmetro que pode assumir valores diferentes em diferentes pontos de execução no workflow e substituir as definições de configuração nos próximos nodes.

**Portas Ocultas de Variáveis de Fluxo**


Cada node possui 2 portas de variáveis de fluxo ocultas para aceitá-las de entrada e propagá-las aos próximos nodes. Para tornar visíveis, clique com o botão direito do mouse no node e selecione "Show Flow Variable Ports". Somente portas do mesmo tipo podem ser conectadas

**Criando uma Flow variable**


- Clique com o botão direito no workflow no KNIME Explorer e selecione **Workflow Variables**.
- Use um node de Quickform para criar uma flow variable em qualquer ponto do seu workflow.
- Use qualquer node que converta dados em Flow variables.
- Na configuração do node, na guia Flow Variables, preencha uma caixa em branco com o nome da flow variable.




**String Manipulation**  
(Variable)  
Este node é a versão Flow Variables do node String Manipulation. Da mesma forma, outros nodes têm sua própria versão para variáveis de fluxo, como o node Rule Engine Variable e o node Math Formula (Variable).



**Merge Variables**  
Combina variáveis de fluxo de ramificações separadas. Se as variáveis de fluxo com o mesmo nome forem coletadas, a variável de fluxo na conexão mais alta será mantida.




**Inject Variables**  
(Data)  
Adiciona (injeta) as Variáveis de Fluxo contida na porta de entrada de Variável na tabela de dados da porta de entrada superior. A tabela de dados de entrada é encaminhada (inalterada) na porta de saída do node.




**Extract Variables**  
(Data)  
Extraí as Variáveis de Fluxo que entram pela porta de dados de entrada e as produz como variáveis de fluxo independentes na porta de saída.


**Nodes de Widget e Configuração** criam uma ou mais novas Variáveis de Fluxo e as disponibiliza na porta de saída. Os nodes do widget criam um componente de UI para a visualização composta ou KNIME WebPortal (caixa de texto, botão de opção etc.) para ter a opção de alterar o valor padrão da variável de fluxo criada. Os nodes de configuração criam um componente de UI na caixa de diálogo de um componente e não são visíveis na visualização composta ou KNIME WebPortal. Se um ou mais nodes de configuração forem incluídos em um componente, o componente adquirirá uma janela de configuração composta pelos components da UI dos nodes de configuração contidos.




**Column Selection Widget**  
Cria uma lista de colunas selecionáveis da tabela de dados de entrada na forma de um menu ou botões de opção. O node produz o nome da coluna selecionada em uma variável de fluxo em sua porta de saída




**Interactive Range Slider Filter Widget**  
Cria um controle deslizante para filtrar dados que incluem apenas linhas com valores na coluna selecionada dentro do intervalo especificado. O controle deslizante pode interagir com visualizações de outros nodes baseados em JavaScript na mesma visualização composta




**Text Output Widget**  
Cria um parágrafo de texto livre, pré-formatado ou HTML




**Credentials Selection Widget**  
Cria campos para inserir credenciais (nome de usuário e senha) na forma de caixas de texto. O texto na caixa da senha está mascarado. O node produz essas credenciais em uma variável de fluxo em sua porta de saída



**Single Selection Configuration**  
Cria uma lista de opções do tipo String na forma de um menu ou botões de opção. Essas podem ser definidas na caixa de configuração junto com o valor padrão selecionado. O node produz o valor da opção selecionada em uma variável de fluxo em sua porta de saída



**Boolean Configuration**  
Cria uma seleção booleana para um sinalizador ativado/desativado (1/0) na forma de uma caixa de seleção. O node produz o valor da opção selecionada em uma variável de fluxo em sua porta de saída (1 se a caixa de seleção estiver ativada, 0 se desativada)



**Metanodes** apenas coletam nodes e são uma maneira eficiente de limpar seu workflow.

**Um Metanode ou Component** é um node que contém outros nodes. **Criando um Metanode ou Component** Selecione todos os nodes relevantes, clique com o botão direito do mouse e selecione "Collapse into Metanode" para um metanode ou "Encapsulate into Component" para um component. Clicar com o botão direito do mouse em um Metanode ou Component abre o menu com várias opções, como abrir, expandir, configurar ou reconfigurar e salvar como modelo.

**Um loop** é uma sequência de operações repetidas até que uma condição seja atendida. Tem começo, fim e corpo de operações. Um loop é implementado via um node Loop Start, um Loop End e vários nodes no meio para o corpo das operações. Diversos nodes de Loop Start fornecem maneiras alternativas de iterar nos dados de entrada. Diferentes nodes Loop End dão alternativas para coletar resultados. A condição final pode ser definida no node Loop Start ou node End Loop, dependendo do tipo de loop. Alguns nodes funcionam com dados e outros com variáveis de fluxo para iniciar e finalizar um loop. Esses nodes podem ser emparelhados livremente - os loops podem começar com dados e terminar com variáveis de fluxo e vice-versa.



**Table Row to Variable Loop Start**  
Inicia um loop que itera sobre as linhas de dados de entrada uma por vez, ou seja, cada iteração é dedicada a apenas uma linha. A cada iteração, os valores da linha são convertidos em variáveis de fluxo e nomeados após os cabeçalhos das colunas.



**Group Loop Start**  
Inicia um loop que itera sobre grupos de linhas de dados de entrada, ou seja, cada iteração funciona em um subgrupo diferente dos dados de entrada. Os grupos são extraídos dos valores nas colunas selecionadas, como na janela de configuração do node GroupBy.



**Column List Loop Start**  
Inicia um loop que itera sobre uma lista selecionada de colunas. A cada iteração, a coluna atual e as colunas restantes são passadas para o corpo do loop.



**Generic Loop Start**  
Inicia um loop. Ele deve ser emparelhado com um node Loop End que define a condição final.




**Parallel Chunk Start**  
Inicia um loop paralelo iterando sobre as linhas de dados de entrada em pedaços de tamanho igual, processados em paralelo no corpo do loop. O número de iterações paralelas (*chunks*) é definido automaticamente, baseado no tamanho dos dados de entrada, ou manualmente.




**Recursive Loop Start**  
Inicia uma iteração de loop recursivo na tabela de entrada. A tabela de entrada para cada iteração é a tabela de dados de saída da iteração anterior. A primeira iteração funciona na tabela de dados fornecida na porta de entrada do node. Esse é o único loop em que a tabela de dados da entrada alimenta a próxima iteração.




**Feature Selection Loop Start (1:1)**  
Inicia um loop de seleção de recurso que itera em um conjunto de colunas de entrada (recursos) para extrair o subconjunto que otimiza uma determinada variável de fluxo. Usado para seleção de recursos em um determinado modelo e em uma determinada métrica de desempenho. Juntamente com o node "Feature Selection Loop End", é usado no metanode




**Loop End**  
Termina um loop concatenando as linhas resultantes de cada iteração




**Loop End (Column Appended)**  
Finaliza um loop juntando as colunas resultantes de cada iteração na coluna que contém os IDs de linha




**Variable Loop End**  
Os nodes de Loop End não funcionam apenas com dados. No final de um loop, você pode passar os resultados como uma variável de fluxo, como no node Variable Loop End




**Variable Condition Loop End**  
Termina um loop quando uma condição é atendida, ou seja, um valor específico em uma variável de fluxo



**Parallel Chunk End**  
Termina um loop paralelo concatenando as linhas de dados resultantes de cada bloco



**Recursive Loop End**  
Termina um loop recursivo. A porta de entrada superior coleta as linhas de dados resultantes, já a inferior coleta a tabela de dados atualizada a ser transmitida de volta ao node Recursive Loop Start. Também define a condição final do loop, como máx. nº de iterações, nº min. de linhas e um valor específico para uma variável de fluxo



**Feature Selection Loop End**  
Finaliza um loop de seleção de recurso. Ele define a variável com uma métrica a ser maximizada/minimizada para a melhor seleção de recurso. A porta de saída superior do node produz as métricas de avaliação para todos os subconjuntos de colunas, já a inferior produz um resumo dos diferentes conjuntos de recursos com os scores associados.


**Switch:** construção que permite executar condicionalmente diferentes sequências de operações por meio de nodes localizados em diferentes ramificações do workflow. Todos iniciam com um node Switch Start e terminam com um node Switch End. Entre eles, várias ramificações paralelas implementam várias sequências de operações. Alguns nodes para iniciar e finalizar uma construção de switch trabalham com dados, outros com variáveis de fluxo, outros com modelos e outros com consultas ao banco de dados. Todos os nodes podem ser emparelhados livremente.



**CASE Switch (Data Start)**  
Ativa seletivamente apenas 1 de suas 3 portas de saída, permitindo 3 caminhos alternativos para os dados de entrada. A porta de saída ativa pode ser configurada manual ou automaticamente através do valor de uma variável de fluxo. O node IF Switch executa a mesma tarefa, mas com apenas 2 portas de saída alternativas (ambas podem estar ativas ao mesmo tempo).



**CASE Switch (Data End)**  
Coleta as linhas de dados resultantes da porta ativa entre as ramificações conectadas às suas portas de entrada. O node End IF funciona de maneira semelhante e é emparelhado com o node IF Switch.



**Try (Data Ports)**  
Inicia uma construção try-catch para habilitar um caminho alternativo para o fluxo de dados em caso de falha na ramificação principal. Um ramo é definido como o ramo principal, enquanto o outro é definido como o ramo secundário. Se a execução falhar na ramificação principal, a ramificação secundária será ativada. Ele deve ser fechado por um node Catch.



**Catch Errors (Data Ports)**  
Fecha uma construção try-catch iniciada com um node Try e coleta os resultados da ramificação ativa.




**Empty Table Switch**  
Fornece um caminho alternativo para o fluxo de dados, caso a ramificação principal não tenha linhas de dados. Ativa a porta de saída superior e desativa a porta de saída inferior se a tabela de entrada tiver pelo menos uma linha de dados. Desativa a porta de saída superior e ativa a porta de saída inferior se a tabela de entrada estiver vazia.




**Active Branch Inverter**  
Altera o status da atividade da ramificação. Se a porta de entrada estiver ativa, a porta de saída se tornará inativa e vice-versa. É frequentemente usado para forçar uma ramificação a produzir uma saída, mesmo que ela esteja inativa e vice-versa (para desativar uma ramificação, mesmo que esteja ativa).


ORQUESTRAÇÃO




**Container Input (Table)**  
Recebe uma tabela de dados do workflow chamador. Se nenhuma entrada for fornecida, a tabela de dados padrão do modelo será usada. Nodes semelhantes estão disponíveis para trocar Variáveis de Fluxo e credenciais. O node "Container Output (Table)" correspondente retorna os resultados como uma tabela de dados.




**Container Input (JSON)**  
Recebe uma estrutura de dados JSON do workflow chamador. Se nenhuma entrada for fornecida, a estrutura JSON padrão do modelo será usada. O node correspondente "Container Output (JSON)" retorna os resultados como uma estrutura JSON.




**GET Request**  
Chama um serviço REST no modo GET. O node pode enviar uma única solicitação de serviço definida na janela de configuração ou várias solicitações de serviço armazenadas em uma coluna da tabela de entrada. As respostas são salvas na tabela de dados de saída. Opções para definir autenticação, cabeçalho de solicitação e cabeçalho de resposta estão disponíveis.




**POST Request**  
Chama um serviço REST no modo POST. O node pode enviar uma única solicitação de serviço definida na janela de configuração ou várias solicitações de serviço armazenadas em uma coluna da tabela de entrada. As respostas são salvas na tabela de dados de saída. Opções para definir autenticação, cabeçalho de solicitação e cabeçalho de resposta estão disponíveis.




**KNIME Server Connection**  
Conecta-se a um KNIME Server usando a URL e as credenciais do servidor. Após a conexão ter sido criada, novos diretórios no servidor podem ser criados e os arquivos remotos podem ser acessados, criados e excluídos.




**Timer Info**  
Informa o número de execuções e tempos de execução para cada node em um workflow. São relatados o tempo de execução de node único e total do workflow. Os tempos de execução para nodes dentro de metanodes também podem ser relatados.




**Send Email**  
Envia e-mails em formato HTML ou de texto usando um SMTP externo para um destinatário - incluindo a mensagem e possíveis anexos.



**Save Workflow**  
Salva o workflow executado (também parcialmente até o ponto).



**Create Directory**  
Cria uma nova pasta na máquina local ou em um KNIME Server. A saída do node envia o URI para a nova pasta como uma variável de fluxo.



**Create File Name**  
Cria um caminho de arquivo usando o local, nome e extensão de arquivo especificados. O caminho do arquivo é produzido na porta de saída como uma variável de fluxo.

Portas de Nodes

Diferentes tipos de dados passam por diferentes portas de node. Somente portas do mesmo tipo podem ser conectadas. Aqui estão alguns exemplos de portas para tipos de dados usados com mais frequência.







Tabela de Dados




Modelo PMML




Flow Variable




Modelo Tree Ensemble




Conexão com Banco de Dados




Contexto Spark




Consulta ao Banco de Dados



Dados Spark



Imagem



HDFS

Recursos

E-Books: KNIME Advanced Luck cobre esses recursos avançados e muito mais. Practicing Data Science é uma coleção de estudos de casos de ciência de dados de projetos anteriores. Ambos disponíveis em [knime.com/knimepress](http://knime.com/knimepress)

KNIME Blog: Tópicos envolventes, desafios, notícias do setor e pitadas de conhecimento em [knime.com/blog](http://knime.com/blog)

Curso de E-Learning Course: faça nosso curso on-line gratuito para aprender sobre as diferentes etapas de um projeto de ciência de dados (com exercícios e soluções para testar seu conhecimento) em [knime.com/knime-introductory-course](http://knime.com/knime-introductory-course)

KNIME Hub: navegue e compartilhe workflows, nodes e components. Adicione classificações ou comentários a outros workflows em [hub.knime.com](http://hub.knime.com)

KNIME Forum: Participe da nossa comunidade global e participe de conversas em [forum.knime.com](http://forum.knime.com)

KNIME Server: Para colaboração, automação, gerenciamento e implantação em equipe, consulte o KNIME Server em [knime.com/server](http://knime.com/server)