

Expressões

- [Expressões - Função Include em FK](#)
- [Expressões - Visão Geral Sobre Expressões BPM](#)

Expressões - Função Include em FK

Esse tópico serve para ensinar como realizar o uso da nova função (Expressão) em campos de FK, usando especialmente a função Includes. Como exemplo, foi feito um processo simples que contém as seguintes configurações:

- Um campo de FK normal.
- Um campo que deve se preencher automaticamente conforme seja selecionada uma opção da FK.
- Um botão que fica visível caso a FK esteja preenchida com determinada informação.

Esse procedimento foi feito tanto comparando pelo ID, quanto pela descrição.

Configuração dos Campos por ID

Campo de FK por ID

A configuração do campo é normal (marque utilizar FK na aba Propriedades), e se atente ao campo destacado em vermelho, pois você precisará fazer uso dele para configurar as expressões.

Campo de preenchimento automático

Para o preenchimento automático, você deve marcar a expressão no campo valor, e especificar sua condição nele. Nesse caso, o processo irá preencher com a descrição caso o valor selecionado esteja preenchido, independente do preenchimento escolhido.

Código:

```
CAMPODEFKID.valor ? CAMPODEFKID.valor.descricao : ''
```

Botão que deve ficar visível caso determinada opção seja selecionada

Esse botão está configurado para caso o ID selecionado seja igual a três, o botão ficará visível. O símbolo "?" é como se fosse uma estrutura de IF, uma questão. Nesse caso, faz a seguinte análise: Caso o campo esteja preenchido, ele ficará visível.

Código:

```
[3].includes(CAMPODEFKID.valor ? CAMPODEFKID.valor.id : 0)
```

Configuração dos Campos por Descrição

Campo de FK por Descrição

A configuração do campo é normal (marque utilizar FK na aba Propriedades), e se atente ao campo destacado em vermelho, pois você precisará fazer uso dele para configurar as expressões. A única diferença de fazer com ID, é que o identificador está diferente por ser outro campo. O identificador nunca pode se repetir em uma atividade.

Campo de preenchimento automático

Para o preenchimento automático, você deve marcar a expressão no campo valor, e especificar sua condição nele. Nesse caso, o processo irá preencher com a descrição caso o valor selecionado esteja preenchido, independente do preenchimento escolhido.

Código:

```
CAMPODEFKDESCRICA0.valor ? CAMPODEFKDESCRICA0.valor.descricao : ''
```

Botão que deve ficar visível caso determinada opção seja selecionada

Esse botão está configurado para caso a descrição selecionada seja igual a "Consumidor final", o botão ficará visível. O símbolo "?" é como se fosse uma estrutura de IF, uma questão. Nesse caso, faz a seguinte análise: Caso o campo esteja preenchido, ele ficará visível.

Código:

```
['Consumidor final'].includes(CAMPODEFKDESCRICA0.valor ? CAMPODEFKDESCRICA0.valor.descricao : '')
```

Campos de Modelos diferentes

Caso queira fazer uma situação, por exemplo, com um campo FK e um campo dissertativo, você pode fazer uso da seguinte sintaxe:

Código:

```
[3].includes(CAMP0DEFKID.valor ? CAMP0DEFKID.valor.id : 0) &&  
['0la'].includes(CAMP0DEDESCRICA0.valor ? CAMP0DEDESCRICA0.valor : '')
```

Dessa forma vai estar comparando um campo de FK, e um outro campo de texto incluído em uma Dissertativa.

Expressões - Visão Geral Sobre Expressões BPM

A partir da versão 12.24, os processos de negócio (BPM) contam com um novo conceito: Propriedades definidas por expressão. Isso traz dinamismo para a colaboração de processos sem depender do PEX.

1 - O que é uma expressão?

Expressão pode ser entendida como uma fórmula matemática que vai retornar um valor, para representar o estado de uma propriedade. A expressão pode ser uma média de valores ou uma verificação se algum campo foi informado, por exemplo.

2 - Motivações

Os principais objetivos de implementar as expressões foram:

- Dar uma alternativa ao atual evento ao sair de campos (PEX), que fosse mais performática e mais simples.
- Avançar no conceito de Low-Code dentro do DOX.

3 - Identificador

Para possibilitar as expressões, foi criado o campo “Identificador”, cujo propósito é auto-explicativo: ser um identificador único para um campo de uma atividade. Atualmente, nos eventos ao sair, é comum encontrar linhas de código parecidas com essa:

```
LValorAtual := aoFormularios.GetJSON('1').GetStr('TEXT0');
```

Nessa abordagem fica difícil entender o que está acontecendo analisando apenas essa linha de código: Afinal, ao que corresponde o aoFormulario “1” e ao que corresponde o “TEXT0”? Partindo para uma abordagem em que o campo pode ser identificado por um nome, ao invés de um número, o código fica muito mais compreensível.

3.1 - Gerando o identificador

Ao atualizar o sistema para a versão 12.24 e abrir uma atividade de um processo já cadastrado, uma nova coluna vai aparecer na listagem de campos: a coluna “identificador”, já preenchida, de acordo com o “Código” de cada campo, da seguinte forma: _1, _2, _3, etc. Isso porque esse campo não pode ficar vazio, nem pode se repetir na atividade e para usuários que não vão utilizar expressões, esse campo vai ser indiferente.