



DICA TÉCNICA 02

Colunas Calculadas

Sumário

1. INTRODUÇÃO	3
2. CRIANDO UMA COLUNA CALCULADA.....	3
3. SINTAXE	4
4. APLICAÇÕES COMUNS DE COLUNAS CALCULADAS.....	5
4.1 UTILIZANDO OPERADORES ARITMÉTICOS	5
4.2 CONCATENANDO TEXTO	5
4.3 SUBSTRING	6
4.4 CONVERTENDO ENTRE TIPOS DE DADOS	7
4.5 UTILIZANDO FUNÇÕES MATEMÁTICAS PRONTAS.....	7
4.6 UTILIZANDO IF-ELSE	7
4.7 COMPODO FÓRMULAS COM E / OU.....	8
4.8 INSERÇÃO DE CONSTANTES COM CASAS DECIMAIS	9
4.9 CALCULAR DIFERENÇA DE DIAS ENTRE DATAS.....	9
4.10 SOMAR/SUBTRAIR N DIAS DE UMA DATA	9
4.11 IDENTIFICAR MÊS ANTERIOR.....	9
4.12 TRANSFORMAR HORA, MINUTO E SEGUNDO (FORMATO HH:MM:SS) EM MINUTOS (E DECIMAL)	9
5. FAQs	10
APÊNDICE 1 – MEMBROS DA CLASSE MATH.....	10

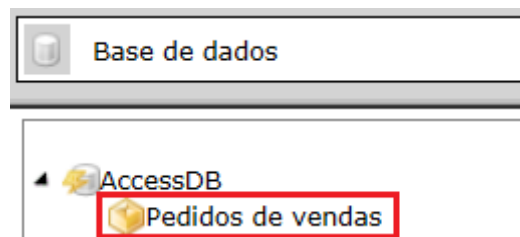
1. Introdução

No BXBwebsite, as fórmulas ou colunas calculadas são utilizadas para gerar dimensões ou métricas com base em cálculos onde as variáveis são as informações de outras dimensões ou métricas do cubo.

Ao ser criada, cada coluna calculada passa a fazer parte do cubo que lhe deu origem na condição de dimensão ou métrica, como qualquer outra coluna, o que permite seu uso em relatórios e até mesmo como variáveis em novas colunas calculadas.

2. Criando uma coluna calculada

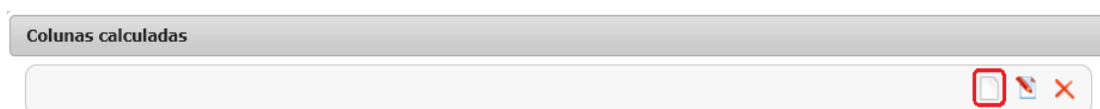
Para criar uma coluna calculada, basta selecionar um cubo no módulo de base de dados



e clicar no ícone *Colunas calculadas*, no canto superior direito da tela.



Com isso, será aberta a tela de manutenção de colunas calculadas para o cubo selecionado. Clicando no botão “Novo”, abre-se a tela “Manutenção de coluna calculada”.



Na primeira parte da tela “Manutenção de coluna calculada” encontram-se os campos para gerenciar o nome e a respectiva expressão. Também é exibida uma lista com as métricas/dimensões que podem ser utilizadas na expressão da coluna calculada. Para adicioná-las à expressão, basta selecionar o item escolhido e clicar no botão *Adicionar metadado* (<). Para verificar se a sintaxe está correta, basta clicar no botão *Testar sintaxe*.

Manutenção de coluna calculada

Nome:*

result =

Colunas do Cubo

- Ano (text)
- Ano-Mes (text)
- Categ produto (text)
- Cidade (text)
- Cliente (text)
- Cod vendedor (text)
- Data pedido (date)
- Mes (text)
- Num pedido (decimal number)
- País (text)
- Produto (text)
- Qtd faturada (decimal number)
- Qtd solicitada (decimal number)
- Valor custo (decimal number)
- Valor da venda (decimal number)

Testar sintaxe

Avançar >> Cancelar

Na segunda parte, encontram-se as outras propriedades da coluna calculada, como *Rótulo* e *Formato*.

Manutenção de coluna calculada

Nome: NovaFormula

Rótulo:* NovaFormula

Descrição:

Unidade:

Formato: #.##0,00

Sumarização default: Soma

<< Voltar Aplicar Cancelar

3. Sintaxe

Para criar uma coluna calculada a sintaxe utilizada é ***formula = expressão***; O valor da coluna calculada será o valor atribuído a variável *formula*, na expressão introduzida. A *expressão* da coluna calculada pode ser composta por operações entre métricas, dimensões e valores fixos. As operações envolvem operadores aritméticos, relacionais e lógicos.

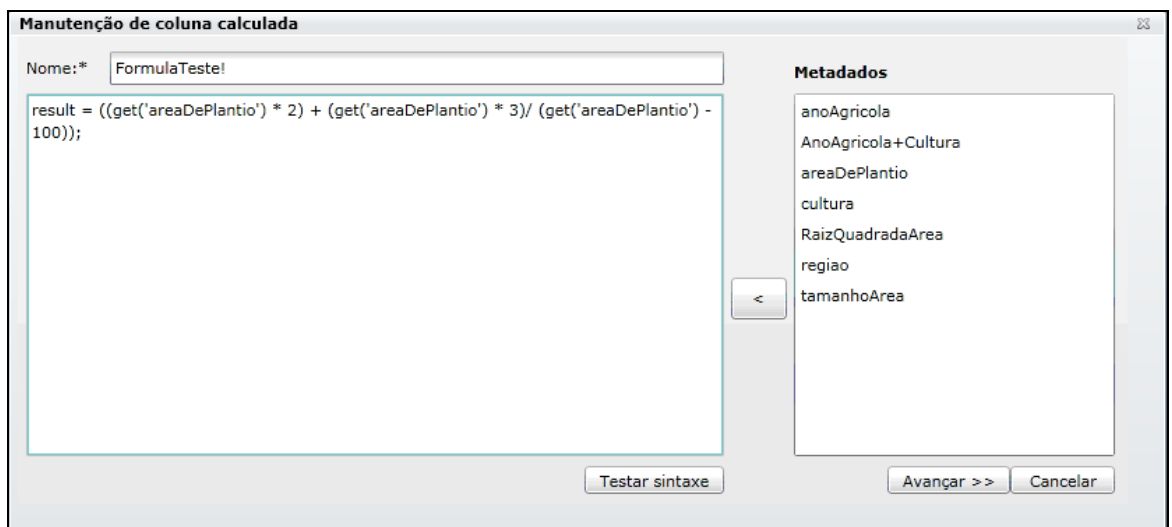
Exemplo:

*formula = ((get('areaDePlantio') * 2) + (get('areaDePlantio') * 3) / (get('areaDePlantio') - 100));*

4. Aplicações comuns de colunas calculadas

4.1 Utilizando operadores aritméticos

Exemplifica o uso de operadores aritméticos, multiplicação, divisão, adição, subtração, aplicando precedência de operadores com o uso dos parêntesis.

$$formula = ((get('areaDePlantio') * 2) + (get('areaDePlantio') * 3) / (get('areaDePlantio') - 100));$$


Resultado:

regiao	areaDePlantio	FormulaTeste!
CARAMBEÍ	6.783,25	13.569,54
CENTRO OESTE-MS	2.492,65	4.988,43
GOIÁS	23.579,42	47.161,85
LDB-LONDRINA	4.728,68	9.460,42
LDB-NORTE	6.443,07	12.889,19

4.2 Concatenando texto

Exemplifica a concatenação das dimensões *anoAgricola* e *cultura*, separadas por um hífen.

$$formula = get('anoAgricola') + " - " + get('cultura');$$

Manutenção de coluna calculada

Nome:*

result = get('anoAgricola') + " - " + get('cultura')

Metadados

- anoAgricola
- areaDePlantio
- cultura
- FormulaTeste!
- RaizQuadradaArea
- regiao
- tamanhoArea

Testar sintaxe Avançar >> Cancelar

Resultado:

anoAgricola	cultura	AnoAgricola+Cultura	areaDePlantio
2006/2007	ALGODAO	2006/2007 - ALGODAO	0,86
2006/2007	ARROZ	2006/2007 - ARROZ	2.389,20
2006/2007	FEIJÃO	2006/2007 - FEIJÃO	100,70
2006/2007	MILHO	2006/2007 - MILHO	4.296,70
2006/2007	SOJA	2006/2007 - SOJA	114.147,19

ATENÇÃO: Utilizar aspas como delimitador de texto fixo a ser concatenado com os valores, como no exemplo acima, onde o hífen está entre aspas. Se os valores a serem concatenados forem métricas, e utilizar apóstrofos como delimitador, o resultado da expressão será os valores somados com o valor ASCII do caracter (no caso o hífen, que tem valor 45).

4.3 Substring

Para pegar parte de um texto, basta utilizar a função *Substring*, com a seguinte sintaxe:

formula = (get('ano-mes')).Substring(startIndex,length);

startIndex: posição inicial, sendo que a primeira posição é 0

length: quantidade de caracteres a ser extraída (tamanho do campo resultante)

Por exemplo, se você tem uma coluna texto "ano-mes" com o conteúdo no formato "2018-08", para extrair o mês você deve fazer:

formula = (get('ano-mes')).Substring(5,2);

4.4 Convertendo entre tipos de dados

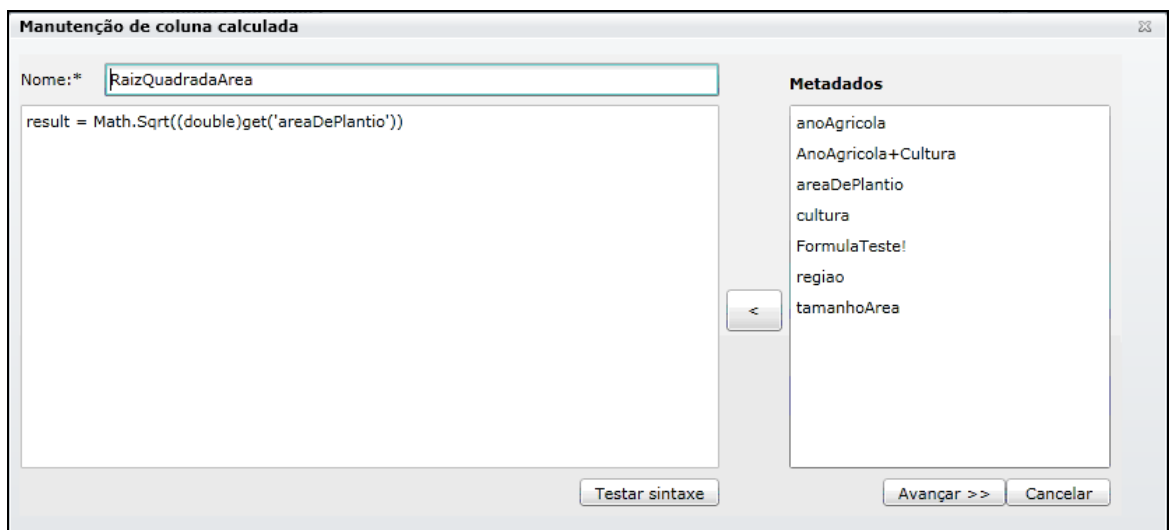
Para converter um dado numérico para texto, basta utilizar a expressão `Convert.ToString(valor)`. Para converter de texto para numérico, podemos utilizar `Convert.ToInt32(valor)`, `Convert.ToDouble(valor)` e `Convert.ToDecimal(valor)`.

```
formula = get('cultura') + Convert.ToString(get('areaDePlantio'));
```

4.5 Utilizando funções matemáticas prontas

Ao criar colunas calculadas no BXBwebsitesuite, é possível utilizar constantes e funções que facilitam a realização de muitas operações matemáticas. No exemplo abaixo, é utilizada a função `Sqrt` para calcular a raiz quadrada da métrica `areaDePlantio`. Note que nessa coluna calculada é feita uma conversão entre tipos de dados. Para mais informações a respeito, veja item 5 da seção FAQs. Para ver a lista completa de constantes e funções da classe `Math`, vide item **Apêndice 1 - Membros da classe Math**.

```
formula = Math.Sqrt((double) get('areaDePlantio'));
```



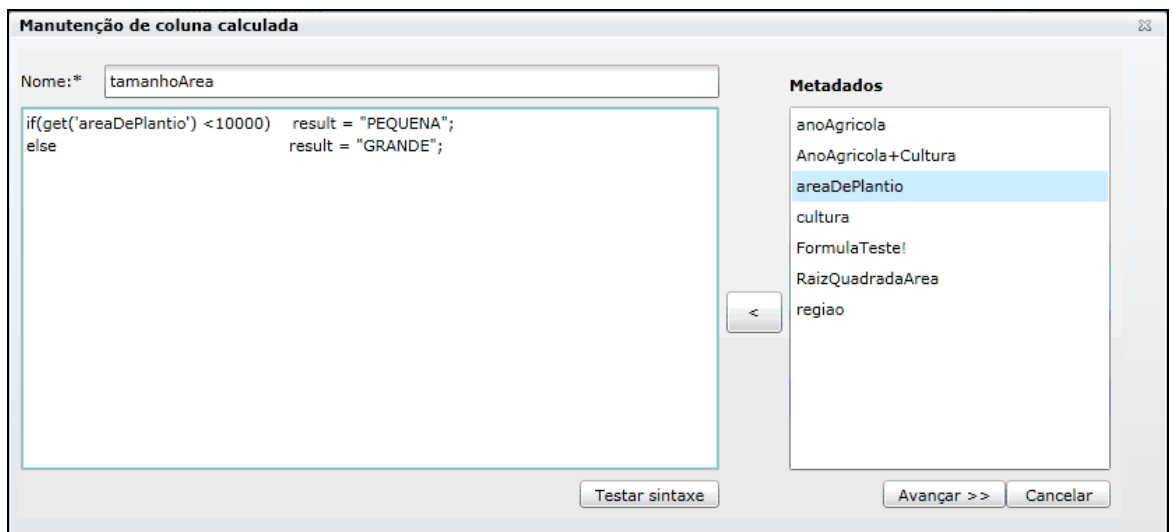
Resultado:

regiao	areaDePlantio	RaizQuadradaArea
CARAMBÉI	6.783,25	82,36
CENTRO OESTE-MS	2.492,65	49,93
GOIÁS	23.579,42	153,56
LDB-LONDRINA	4.728,68	68,77
LDB-NORTE	6.443,07	80,27

4.6 Utilizando if-else

Além dos operadores aritméticos relacionais e lógicos, é possível utilizar as dimensões/métricas para aplicar condições que vão influenciar no resultado, utilizando a sintaxe *if else*. No exemplo abaixo, se o valor da métrica *areaDePlantio* for menor que 10000, o resultado recebe o valor “PEQUENA”, senão, recebe o valor “GRANDE”.

```
if(get('areaDePlantio') <10000) result = "PEQUENA";
else result = "GRANDE";
```



Essa coluna calculada também poderia ser escrita utilizando operador ternário (maneira compacta de expressar *if-else*), da seguinte forma:

```
result = get('areaDePlantio') < 10000 ? "PEQUENA" : "GRANDE";
```

Resultado:

regiao	areaDePlantio	tamanhoArea
CARAMBÉI	6.783,25	PEQUENA
CENTRO OESTE-MS	2.492,65	PEQUENA
GOIÁS	23.579,42	GRANDE
LDB-LONDRINA	4.728,68	PEQUENA
LDB-NORTE	6.443,07	PEQUENA

Exemplo tendo uma dimensão como condição:

```
formula= get('status') == "Overdue" ? get('Invoice_amount') : 0
```

4.7 Compondo fórmulas com E / OU

“E” é representado por &&

“OU” é representado por ||

Exemplo de cálculo de um valor em atraso:

```
formula= get('Due_date') < DateTime.Today && (get('receipt_date') ==
Convert.ToDateTime("1900-01-01")) ? get('Invoice_amount') : 0
```



```
formula= get('Due_date') > DateTime.Today || (get('receipt_date') <>
Convert.ToDateTime("1900-01-01")) ? 0 : get('Invoice_amount')
```

4.8 Inserção de constantes com casas decimais

Exemplo, obter 10% do valor da venda :

```
result = get('Valor da venda') * (decimal)0.1;
```

4.9 Calcular diferença de dias entre datas

```
formula = (get('DataFinal') - get('DataInicial')).Days;
```

4.10 Somar/subtrair n dias de uma data

```
formula = DateTime.Today.AddDays(-30);
formula = (get('DataVenda')).AddDays(30);
```

4.11 Identificar mês anterior

```
formula = ((get('DataVenda')).Year == DateTime.Today.AddMonths(-1).Year &&
(get('DataVenda')).Month == DateTime.Today.AddMonths(-1).Month) ? 'Mês anterior' :
'Outro mês';
```

4.12 Transformar hora, minuto e segundo (formato HH:MM:SS) em minutos (e decimal)

```
formula = Convert.ToDecimal((Convert.ToDecimal((get('HHMMSS')).Split(new Char[]{' ':''})[0])
* 60) + Convert.ToDecimal((get('HHMMSS')).Split(new Char[]{' ':''})[1]) +
(Convert.ToDecimal((get('HHMMSS')).Split(new Char[]{' ':''})[2]) / 60))
```

5. FAQs

- a. Quando tento adicionar uma coluna calculada apresenta um erro do tipo “Melhor método sobrecarregado compatível com 'System.Math.Sqrt(double)' tem alguns argumentos inválidos. Argumento '1': não pode converter de 'decimal' em 'double'.". O que devo fazer?

Resposta: Esse erro acontece porque o método *Sqrt* da classe *Math* espera como parâmetro um valor numérico do tipo *double*, e o parâmetro utilizado representa um numérico do tipo *decimal*. Para executar a coluna calculada corretamente, basta fazer um *typecast* (conversão de tipo de dado), da seguinte maneira: “System.Math.Sqrt((double)parâmetro)”.

Apêndice 1 – Membros da classe Math

Funções:

Nome	Descrição	Sintaxe
Abs	Retorna o valor absoluto de um número específico.	<i>Math.Abs(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Acos	Retorna o ângulo cujo cosseno é o número especificado.	<i>Math.Acos(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Asin	Retorna o ângulo cujo seno é o número especificado.	<i>Math.Asin(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Atan	Retorna o ângulo cuja tangente é o número especificado.	<i>Math.Atan(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Atan2	Retorna o ângulo cuja tangente é o quociente de dois números especificados.	<i>Math.Atan2(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
BigMul	Produz o produto completo de dois números de 32 bits.	<i>Math.BigMul(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>int</i> .
Ceiling	Retorna o menor inteiro maior ou igual ao número especificado.	<i>Math.Ceiling(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> ou <i>decimal</i> .

Cos	Retorna o cosseno do ângulo especificado.	<i>Math.Cos(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Cosh	Retorna o cosseno hiperbólico do ângulo especificado.	<i>Math.Cosh(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
DivRem	Calcula o quociente de dois números e também retorna o resto em um parâmetro de saída.	<i>Math.DivRem(x, y, z)</i> Onde <i>x, y</i> e <i>z</i> são numéricos, do tipo <i>int</i> .
Exp	Retorna <i>e</i> elevado à potência especificada.	<i>Math.Exp(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Floor	Retorna o maior inteiro menor ou igual ao número especificado.	<i>Math.Floor(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> ou <i>decimal</i> .
IEEERemainder	Retorna o número especificado do restante resultantes da divisão de um número especificado por outro.	<i>Math.IEEERemainder(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
Log	Retorna o logaritmo de um número especificado em uma base especificada.	<i>Math.Log(x)</i> ou <i>Math.Log(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
Log10	Retorna o logaritmo base 10 de um número especificado.	<i>Math.Log10(x)</i> Onde <i>x</i> é numérico, do tipo <i>double</i> .
Max	Retorna o maior dos dois números especificados.	<i>Math.Max(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>int</i> , <i>double</i> ou <i>decimal</i> .
Min	Retorna o menor dos dois números.	<i>Math.Min(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>int</i> , <i>double</i> ou <i>decimal</i> .
Pow	Retorna um número especificado elevado à potência especificada.	<i>Math.Pow(x, y)</i> Onde <i>x</i> e <i>y</i> são numéricos, do tipo <i>double</i> .
Round	Arredonda um valor para o	<i>Math.Round(x)</i> ou <i>Math.Round(x, y)</i>

	inteiro mais próximo ou o número especificado de casas decimais.	Onde x e y são numéricos, x do tipo <i>double</i> ou <i>decimal</i> e y do tipo <i>int</i> .
Sign	Retorna um valor indicando o sinal de um número.	<i>Math.Sign(x)</i> Onde x é numérico, do tipo <i>int</i> , <i>double</i> ou <i>decimal</i> .
Sin	Retorna o seno do ângulo especificado.	<i>Math.Sin(x)</i> Onde x é numérico, do tipo <i>double</i> .
Sinh	Retorna o seno hiperbólico do ângulo especificado.	<i>Math.Sinh(x)</i> Onde x é numérico, do tipo <i>double</i> .
Sqrt	Retorna a raiz quadrada do número especificado.	<i>Math.Sqrt(x)</i> Onde x é numérico, do tipo <i>double</i> .
Tan	Retorna a tangente do ângulo especificado.	<i>Math.Tan(x)</i> Onde x é numérico, do tipo <i>double</i> .
Tanh	Retorna a tangente hiperbólica do ângulo especificado.	<i>Math.Tanh(x)</i> Onde x é numérico, do tipo <i>double</i> .
Truncate	Calcula a parte integral de um número.	<i>Math.Truncate(x)</i> Onde x é numérico, do tipo <i>double</i> ou <i>decimal</i> .

Constantes:

Nome	Descrição	Valor	Sintaxe
E	Representa a natural logarítmica base, especificado pela constante e .	2.7182818284590452354	Math.E
PI	Representa a proporção entre a circunferência de um círculo ao seu diâmetro, especificado pela constante π .	3.14159265358979323846	Math.PI