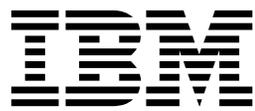


IBM Cognos Analytics  
Versão 11.0

*Guia de modelagem de dados*



©

## **Informações do produto**

Este documento se aplica ao IBM Cognos Analytics versão 11.0.0 e também pode se aplicar a liberações subsequentes.

## **Copyright**

Licensed Materials - Property of IBM

© Copyright IBM Corp. 2015, 2018.

Direitos restritos aos usuários do governo dos EUA - Uso, duplicação ou divulgação restritos pelo GSA ADP Schedule Contract com a IBM Corp.

IBM, o logotipo IBM e [ibm.com](http://ibm.com) são marcas ou marcas registradas da International Business Machines Corp., registradas em várias jurisdições em todo o mundo. Outros nomes de produtos e serviços podem ser marcas registradas da IBM ou de outras empresas. Uma lista atual de marcas registradas da IBM está disponível na Web em "Copyright and trademark information" em [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

---

# Índice

<b>Capítulo 1. Modelagem de dados no Cognos Analytics</b>	<b>1</b>
<b>Capítulo 2. Criando um módulo de dados</b>	<b>3</b>
Usando uma origem de módulo de dados	3
Usando uma origem do servidor de dados	4
Usando uma origem de arquivo transferido por upload	5
Usando uma origem de conjunto de dados	5
Usando uma origem de pacote	6
Criando um módulo de dados simples	6
Vinculando origens novamente	7
<b>Capítulo 3. Refinando um módulo de dados.</b>	<b>11</b>
Relacionamento	12
Criando um relacionamento desde o início	13
Cálculos	14
Criando cálculos básicos	14
Agrupando Dados	15
Limpando dados	16
Criando cálculos customizados	18
Criando caminhos de navegação	18
Filtrando Dados	20
Ocultando tabelas e colunas	21
Validando módulos de dados	22
Propriedades de tabela e coluna	22
<b>Apêndice A. Usando o Editor de Expressão.</b>	<b>27</b>
Operadores	27
(	27
)	27
*	27
/	27
	27
+	27
-	28
<	28
<=	28
<>	28
=	28
>	28
>=	28
e	29
meio.	29
tipo de letra	29
contém	29
distinct	29
else	30
termina.	30
terminar por	30
se.	30
in.	30
está faltando	30
like	30
procurar	31
não	31
ou	31

começa com . . . . .	31
then . . . . .	31
quando . . . . .	32
Resumos . . . . .	32
Funções estatísticas . . . . .	32
médio . . . . .	33
contar . . . . .	33
máximo . . . . .	33
Mediano . . . . .	34
minimum . . . . .	34
percentage . . . . .	34
percentil . . . . .	34
quantil . . . . .	35
quartil . . . . .	36
rank . . . . .	36
tertile . . . . .	37
total . . . . .	37
Funções de data/hora comerciais . . . . .	38
_add_seconds . . . . .	38
_add_minutes . . . . .	38
_add_hours . . . . .	39
_add_days . . . . .	40
_add_months . . . . .	40
_add_years . . . . .	41
_age . . . . .	42
current_date . . . . .	42
current_time . . . . .	42
current_timestamp . . . . .	42
_day_of_week . . . . .	43
_day_of_year . . . . .	43
_days_between . . . . .	43
_days_to_end_of_month . . . . .	44
_end_of_day . . . . .	44
_first_of_month . . . . .	44
_from_unixtime . . . . .	44
_hour . . . . .	45
_last_of_month . . . . .	45
_make_timestamp . . . . .	45
_minute . . . . .	46
_month . . . . .	46
_months_between . . . . .	46
_second . . . . .	46
_shift_timezone . . . . .	47
_start_of_day . . . . .	48
_week_of_year . . . . .	48
_timezone_hour . . . . .	49
_timezone_minute . . . . .	49
_unix_timestamp . . . . .	49
_year . . . . .	49
_years_between . . . . .	50
_ymdint_between . . . . .	50
Funções comuns . . . . .	50
abs . . . . .	50
conversão . . . . .	51
limitando . . . . .	51
char_length . . . . .	52
unir . . . . .	52
exp . . . . .	52
floor . . . . .	52
ln . . . . .	53
inferior . . . . .	53

mod . . . . .	53
nullif . . . . .	54
posição . . . . .	54
position_regex . . . . .	54
potência . . . . .	55
_round . . . . .	55
raiz quadrada . . . . .	56
substring . . . . .	56
substring_regex . . . . .	56
trim . . . . .	57
maiúscula . . . . .	57
Funções trigonométricas . . . . .	57
<b>Apêndice B. Sobre este Guia . . . . .</b>	<b>61</b>
<b>Índice Remissivo . . . . .</b>	<b>63</b>



---

## Capítulo 1. Modelagem de dados no Cognos Analytics

É possível usar a modelagem de dados no IBM® Cognos Analytics para juntar muitas origens de dados, incluindo bancos de dados relacionais, tecnologias baseadas em Hadoop, planilhas do Microsoft Excel, arquivos de texto, etc. Usando essas origens, é criado um módulo de dados que pode então ser usado em relatórios e painéis.

Esquemas em estrela são a estrutura do banco de dados ideal para módulos de dados, mas os esquemas transacionais também são suportados.

É possível refinar um módulo de dados criando cálculos, definindo filtros, referenciando tabelas adicionais, atualizando metadados e mais.

Depois de salvar módulos de dados, outros usuários possam acessá-los. Salve o módulo de dados em uma pasta que usuários, grupos e funções tenham permissões apropriadas para acessar. Esse procedimento é a mesma ideia que salvar um relatório ou painel em uma pasta que controle quem pode acessá-lo.

Os módulos de dados podem ser usados em painéis e relatórios. Um painel pode ser montado a partir de vários módulos de dados.

**Dica:** A modelagem de dados no Cognos Analytics não substitui o IBM Cognos Framework Manager, o IBM Cognos Cube Designer ou o IBM Cognos Transformer, que permanecem disponíveis para modelagens mais complexas.

### Modelagem orientada por intento

É possível usar a modelagem orientada por tentativa para incluir tabelas em seu módulo de dados. A modelagem orientada por tentativa propõe tabelas para incluir no módulo, com base em correspondências entre os termos que você fornece e os metadados nas origens subjacentes.

Enquanto você estiver digitando em palavras-chave para a modelagem orientada por intento, o texto dos nomes de coluna e de tabela nas origens de dados subjacentes serão recuperados pelo software Cognos Analytics. O campo de intento tem uma lista de digitação antecipada de sugere os termos que são localizados nos metadados de origem.

A modelagem orientada por tentativa reconhece a diferença entre as tabelas de fato e as tabelas de dimensão, pelo número de linhas, os tipos de dados e a distribuição de valores entre as colunas. Quando possível, a proposta a intenção de modelagem orientada por tentativa é uma estrela ou um floco de neve de tabelas. Se uma estrela ou um floco de neve apropriado não puder ser determinado, a modelagem orientada por tentativa proporá uma única tabela ou uma coleção de tabelas.



---

## Capítulo 2. Criando um módulo de dados

É possível criar módulos de dados combinando entradas de outros módulos de dados, servidores de dados, arquivos transferidos por upload, conjuntos de dados e pacotes.

Ao criar um novo módulo de dados a partir da tela inicial do IBM Cognos Analytics, são apresentadas cinco origens de entrada possíveis em **Origens**. Essas origens são descritas aqui.

### Módulos de dados

Módulos de dados são objetos de origem que contêm dados de servidores de dados, arquivos transferidos por upload ou outros módulos de dados e são salvos em **Meu conteúdo** ou **Conteúdo da equipe**.

### Servidores de dados

Os servidores de dados são bancos de dados com os quais existem conexões. Para obter mais informações, consulte *Gerenciando o IBM Cognos Analytics*.

### Arquivos transferidos por upload

Os arquivos transferidos por upload são dados armazenados com o recurso **Upload de arquivos**.

### Conjuntos de Dados

Os conjuntos de dados contêm dados extraídos de um pacote ou de um módulo de dados e que são salvos em **Meu conteúdo** ou em **Conteúdo da equipe**.

### Pacotes

Pacotes são criados no IBM Cognos Framework Manager e contêm dimensões, assuntos de consulta e outros dados contidos em projetos do Cognos Framework Manager. Também é possível usar pacotes como origens para um módulo de dados.

É possível combinar diversas origens em um módulo de dados. Após incluir uma

origem, clique em **Incluir origens** (  ) em **Origens selecionadas** para incluir outra origem. É possível usar uma combinação de tipos de origem de dados em um módulo de dados.

Cada tipo de origem de dados é descrito nos tópicos a seguir.

---

## Usando uma origem de módulo de dados

### 11.0.5 11.0.4

Os módulos de dados salvos podem ser usados como origens de dados para outros módulos de dados. Quando um módulo de dados for usado como uma origem para outro módulo de dados, partes desse módulo serão copiadas no novo módulo de dados.

## Procedimento

1. Ao selecionar **Módulos de dados** no painel deslizável **Origens**, será apresentada uma lista de módulos de dados que serão usados como entrada. Marque um ou mais módulos de dados a serem usados como origens.
2. Clique em **Iniciar** ou em **Concluído** nas **Origens selecionadas** para expandir o módulo de dados em suas tabelas de componentes.
3. Arraste as tabelas para o novo módulo de dados.
4. Continue a trabalhar com o novo módulo de dados e, em seguida, salve-o.
5. Se o módulo de dados de origem ou qualquer uma de suas tabelas forem excluídas, então, na próxima vez em que abrir o novo módulo de dados, as tabelas que não estiverem mais disponíveis terão um contorno vermelho no diagrama e **ausente** será listado nos campos **Origem** da área de janela **Propriedades** da tabela.
6. Uma tabela em seu novo módulo de dados que está vinculado é somente leitura. Não é possível modificá-la no novo módulo de dados de modo algum. É possível quebrar o link para o módulo de dados de origem e modificar a tabela clicando em **Quebrar link** nas ações para a tabela.

---

## Usando uma origem do servidor de dados

Servidores de dados são bancos de dados para os quais as conexões existem e podem ser usados como origens para módulos de dados.

É possível usar várias origens de dados do servidor para seu módulo de dados.

### Antes de Iniciar

Conexões do servidor de dados já deverão ter sido criadas em **Gerenciar > Conexões do servidor de dados** ou em **Gerenciar > Console de administração**. Para obter mais informações, consulte *Gerenciando o IBM Cognos Analytics*.

## Procedimento

1. Ao selecionar **Módulos de dados** no painel deslizável **Origens**, será apresentada uma lista de servidores de dados que serão usados como entrada. Selecione o servidor de dados para usar como uma origem.
2. Os esquemas disponíveis no servidor de dados são listados. Escolha o esquema que deseja usar.  
Somente os esquemas para os quais os metadados estiverem pré-carregados são exibidos. Se desejar usar outros esquemas, clique em **Gerenciar esquemas...** para carregar metadados de outros esquemas.
3. Clique em **Iniciar** ou em **Concluído** nas **Origens selecionadas** para expandir o módulo de dados em suas tabelas de componentes.
4. Para começar a preencher o módulo de dados, digite alguns termos no painel deslizante **Intento** e, em seguida, clique em **Acessar**.
5. Um modelo proposto é apresentado. Clique em **Incluir esta proposta** para criar um módulo de dados.
6. Também é possível arrastar tabelas do esquema de servidor de dados para o módulo de dados.

### Exemplo

Para obter um exemplo de módulo de dados criado a partir de um servidor de dados, consulte “Criando um módulo de dados simples” na página 6

## O que Fazer Depois

Se os metadados dos esquemas de servidor de dados forem mudados após criar o módulo de dados, você poderá atualizar os metadados do esquema. Para obter mais informações, consulte o tópico sobre pré-carregamento de metadados de uma conexão do servidor de dados em *Gerenciando o IBM Cognos Analytics*.

---

## Usando uma origem de arquivo transferido por upload

Os arquivos transferidos por upload são dados que são armazenados com o recurso **Upload de arquivos**. É possível usar esses arquivos como origens para um módulo de dados.

### Antes de Iniciar

Os formatos suportados para arquivos transferidos por upload são planilhas do Microsoft Excel (.xlsx e .xls) e arquivos de texto contendo valores separados por vírgula, separados por tabulação, separados por ponto e vírgula ou separados por barra vertical. Somente a primeira folha da planilha do Microsoft Excel é transferida por upload. Se desejar fazer upload dos dados de várias folhas em uma planilha, salve as folhas como planilhas separadas. Os arquivos transferidos por upload são armazenados em um formato de coluna.

Para fazer upload de um arquivo, clique em **Upload de arquivos** na barra de navegação na tela inicial do IBM Cognos Analytics.

### Procedimento

1. Ao selecionar **Arquivos transferidos por upload** no painel deslizável **Origens**, será apresentada uma lista de arquivos transferidos por upload que serão usados como entrada. Marque um ou mais arquivos transferidos por upload a serem usados como origens.
2. Clique em **Iniciar** ou em **Concluído** nas **Origens selecionadas** para expandir o módulo de dados em suas tabelas de componentes.
3. Arraste o arquivo de origem transferido por upload para o módulo de dados para iniciar a modelagem.

---

## Usando uma origem de conjunto de dados

### 11.0.4

Os conjuntos de dados contêm dados que foram extraídos de uma pacote ou de um módulo de dados e que são salvos em **Meu conteúdo** ou em **Conteúdo da equipe**.

### Sobre Esta Tarefa

#### Procedimento

1. Ao selecionar **Conjuntos de dados** no painel deslizável **Origens**, será apresentada uma lista de conjuntos de dados que serão usados como entrada. Marque um ou mais conjuntos de dados a serem usados como origens.
2. Clique em **Iniciar** ou em **Concluído** nas **Origens selecionadas** para expandir o conjunto de dados em suas tabelas e consultas de componentes.
3. Arraste as tabelas ou consultas para o novo módulo de dados.

4. Se os dados nos conjuntos de dados forem mudados, essa mudança será refletida no módulo de dados.

---

## Usando uma origem de pacote

Os pacotes são criados no IBM Cognos Framework Manager. É possível usar os pacotes de modo de consulta dinâmica como origens para módulos de dados.

### Procedimento

1. Ao selecionar **Pacotes** no painel deslizante **Origens**, será apresentada uma lista de pacotes a serem usados como entrada. Marque um ou mais pacotes a serem usados como origens.
2. Clique em **Iniciar** ou em **Concluído** em **Origens selecionadas** para selecionar os pacotes.
3. Arraste os pacotes de origem para seu módulo de dados para iniciar a modelagem.

### O que Fazer Depois

Ao usar um pacote como sua origem de dados, não é possível selecionar tabelas individuais. Deve-se arrastar o pacote inteiro para seu módulo de dados. As únicas ações que são possíveis de serem executadas são de criação de relacionamentos entre assuntos de consulta no pacote e assuntos de consulta no módulo de dados.

---

## Criando um módulo de dados simples

É possível criar um módulo de dados simples com base no banco de dados de vendas Great Outdoors Warehouse que está incluído nas amostras estendidas do IBM Cognos Analytics.

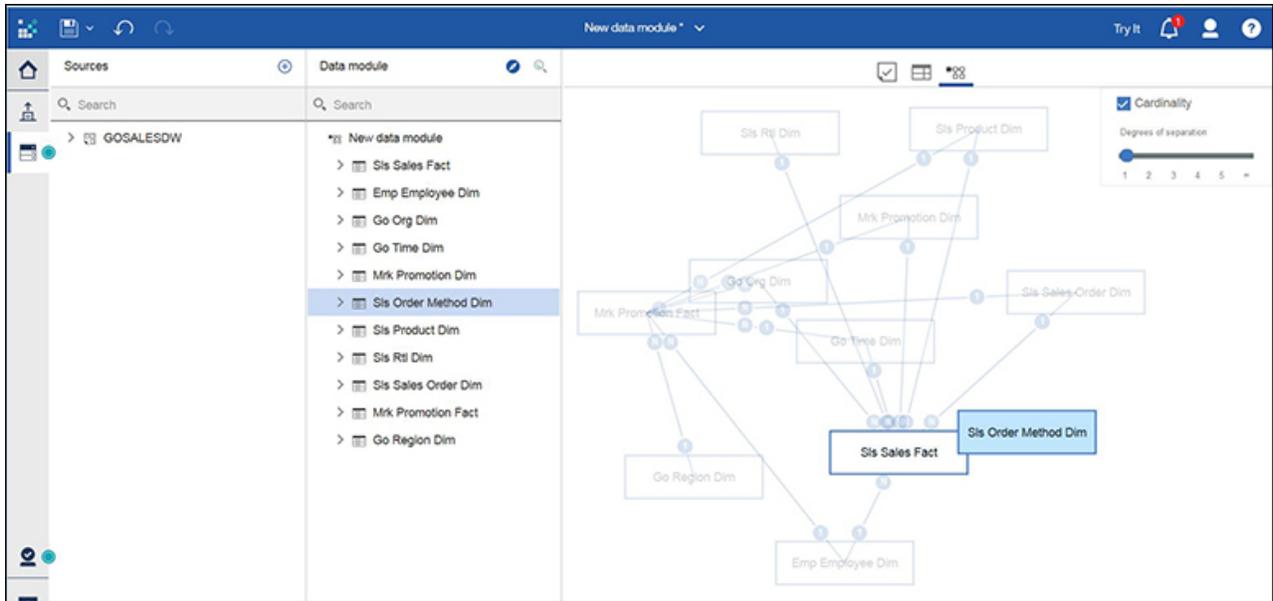
### Antes de Iniciar

Instale o banco de dados do data warehouse da Grandes Aventuras e crie uma conexão com o banco de dados. Para obter mais informações, consulte *Amostras do IBM Cognos Analytics*.

### Procedimento

1. Na tela de boas-vindas do IBM Cognos Analytics, clique em **Novo Módulo de dados**.
2. Em **Origens**, selecione **Servidores de dados**.
3. Em **Servidores de dados**, selecione **great\_outdoors\_warehouse**.
4. Em **great\_outdoors\_warehouse**, selecione o esquema **GOSALESDW**.
5. Em **Origens selecionadas**, clique em **Pronto**.
6. No painel **Módulo de dados**, clique no ícone de modelagem de intenção .
7. No painel **Intenção**, digite renda de vendas e clique em **Ir**. Um modelo proposto é exibido no painel **Intenção**.
8. Clique em **Incluir proposta**. É criado um módulo de dados básico.

No painel seguinte, clique no ícone de diagrama de módulo  para ver o diagrama de módulo de dados que será gerado automaticamente.



9. Agora é possível explorar o módulo de dados. Por exemplo, clique em um item no **Módulo de dados** e, em seguida, clique em suas propriedades  para visualizar e modificar as propriedades do item. Na visualização do diagrama, tente mudar as configurações de **Cardinalidade** para visualizar relacionamentos entre tabelas.
10. Para salvar o módulo de dados, existem as opções **Salvar** ou **Salvar como** .
11. Para criar um relatório a partir de seu módulo de dados, clique em **Tentar**. Uma nova guia é aberta no navegador com o IBM Cognos Analytics - Reporting aberto. Os dados do módulo são mostrados em **Itens de dados de origem**.
12. Arraste **Código de Linha do Produto** de **Dim do Produto SIs** e **Quantidade de Fatos de Vendas SIs** para o relatório.
13. Clique em **Opções de Execução**  para selecionar um formato de saída e, em seguida, clique em **Executar HTML** para executar o relatório e visualizar a saída como uma página da web.

## Vinculando origens novamente

É possível vincular novamente uma origem de módulo de dados a uma origem diferente. Após uma nova vinculação com sucesso, os cálculos globais e os relacionamentos no módulo de dados permanecem válidos.

Aqui estão alguns cenários nos quais uma nova vinculação de uma origem pode ser útil:

- Você constrói e testa um módulo de dados com relação a uma origem de teste. Quando o módulo de dados está pronto, você vincula novamente a origem à origem de produção desejada.
- A origem atual em seu módulo de dados é inválida e uma nova origem válida deve ser usada.

- Você deseja vincular novamente seu módulo de dados de um servidor de dados para outro servidor de dados ou de um esquema para outro esquema.  
A nova vinculação entre diferentes tipos de servidores de dados é suportada, bem como entre esquemas e catálogos em servidores de dados.

**Dica:** Origens do servidor de dados podem ser organizadas em esquemas, catálogos, ambos ou nenhum.

## Sobre Esta Tarefa

A origem vinculada novamente (destino) deve ser do mesmo tipo que a origem original. Um servidor de dados pode ser vinculado novamente apenas a um servidor de dados, um arquivo transferido por upload a um arquivo transferido por upload e assim por diante.

Além dos tipos de origem correspondentes, as condições a seguir devem ser atendidas:

- Todas as colunas da origem original devem existir na origem de destino e as propriedades **Identificador** de colunas (com distinção entre maiúsculas e minúsculas) e os tipos de dados devem corresponder.

Por exemplo, o arquivo A com as colunas ColA e ColB pode ser vinculado novamente ao arquivo B com as colunas ColA e ColB. A nova vinculação ao arquivo B com as colunas colA e colB não funciona.

Os tipos de dados das colunas correspondentes devem ser compatíveis para que os cálculos e os relacionamentos do módulo de dados permaneçam válidos. Por exemplo, quando o tipo de dados da coluna na origem original é date, o tipo de dados da coluna na origem de destino também deve ser date, não string ou timestamp.

- Para servidores de dados, pacotes e módulos de dados, todas as tabelas da origem original devem existir na origem de destino e as propriedades **Identificador** de tabelas (com distinção entre maiúsculas e minúsculas) devem corresponder.

Se uma correspondência duplicada for localizada na origem de destino, a última tabela na lista será usada para a correspondência.

- Colunas e tabelas extras podem existir na origem de destino.

Ao vincular novamente para uma origem que contenha uma tabela com colunas extras, é possível incluir as colunas extras na tabela no módulo de dados arrastando a tabela da área de janela **Origens** para a área de janela **Módulo de dados**.

- Os nomes de origem, como nomes de arquivos e de pacotes ou nomes de conexão do servidor de dados, não precisam corresponder.

**Dica:** A correspondência de colunas e tabelas é feita comparando suas propriedades de **Identificador**. O valor de **Identificador** da coluna ou tabela pode ser, mas nem sempre, igual ao nome da coluna ou tabela (**Rótulo**). É possível visualizar o valor de **Identificador** na área de janela **Propriedades**, seção **Avançado** da coluna ou tabela.

## Procedimento

1. Em **Conteúdo da equipe** ou **Meu conteúdo**, abra seu módulo de dados.
2. Na área de janela **Origens**, localize a origem que você deseja vincular novamente.
3. No menu de contexto de origem, selecione **Vincular novamente**.

4. Selecione o tipo de origem que corresponda ao tipo de origem original. Se a origem original for servidor de dados, selecione o servidor de dados, se for um arquivo transferido por upload, selecione um arquivo e assim por diante.
5. Clique em **Concluído**.

Se a nova vinculação for bem-sucedida, uma mensagem de confirmação será exibida.

Se a nova vinculação foi concluída com erros, uma mensagem será exibida sugerindo que você abra a visualização de validação na qual os problemas da nova vinculação são listados. Resolva os problemas e salve o módulo de dados. Também é possível salvar o módulo de dados com problemas não resolvidos.

**Importante:** O processo de validação não detecta tipos de dados incompatíveis nas colunas. Se houver colunas com tipos de dados incompatíveis em suas origens e se outras condições da nova vinculação forem atendidas, a mensagem de nova vinculação com sucesso será exibida. Esse tipo de problemas de dados deve ser reconciliado nas origens.

## Resultados

Após vincular novamente com sucesso uma origem em um módulo de dados, relatórios e painéis que são baseados nesse módulo de dados poderão começar a usar a nova origem sem qualquer envolvimento dos autores de relatório.



---

## Capítulo 3. Refinando um módulo de dados

O módulo de dados iniciais que você cria manualmente ou usando modelagem de intenção pode conter dados que não são necessários para seus propósitos de relatório. Seu objetivo é criar um módulo de dados que contenha somente os dados que atendem a seus requisitos de relatório e que sejam formatados e apresentados corretamente.

Por exemplo, é possível excluir algumas tabelas do módulo de dados iniciais ou incluir tabelas diferentes. Também é possível aplicar formatação de dados diferente, filtrar e agrupar dados e mudar as propriedades de metadados.

É possível refinar seu módulo de dados aplicando as modificações a seguir:

- Incluir ou excluir tabelas.
- Editar ou criar novos relacionamentos entre tabelas.
- Mudar propriedades da coluna.
- Criar cálculos básicos e customizados.
- Criar caminhos de navegação.
- Definir filtros.
- Agrupar dados.
- Limpar os dados de texto.
- Oculte tabelas e colunas.

É possível iniciar essas ações a partir do painel **Módulo de Dados** ou do diagrama.

Ao trabalhar em um módulo de dados, é possível usar as ações de desfazer  e de refazer  na barra de aplicativos para reverter ou restaurar mudanças no módulo de dados na sessão de edição atual. É possível desfazer ou refazer até 20 vezes.

### Painel de origem

O painel de origem mostra as origens de dados que foram selecionadas quando o módulo de dados foi criado. Os tipos de origens podem incluir outros módulos de dados, servidores de dados, arquivos transferidos por upload, conjuntos de dados e pacotes.

Exceto para pacotes, é possível expandir a origem específica para visualizar suas tabelas e colunas. Arraste as tabelas sobre o painel do módulo de dados ou diagrama para incluí-las no módulo de dados.

### Painel do módulo de dados

A árvore do módulo de dados mostra as tabelas e colunas de dados que estão incluídas no módulo de dados. Esse é o espaço primário para a edição do módulo de dados.

Clique no ícone do menu de contexto  para o módulo, tabela ou coluna para visualizar suas opções de menu de modelagem e edição de contexto. Aqui é possível começar a associar tabela, criar filtros e cálculos ou renomear e excluir itens.

Clique no ícone de modelagem de intenção  na barra de ferramentas do painel para incluir tabelas em seu módulo de dados. A modelagem orientada a intenções propõe tabelas para incluir no módulo que são baseadas em correspondências entre os termos fornecidos e os metadados nas origens subjacentes.

## Diagrama

O diagrama é uma representação gráfica de relacionamentos de tabelas em um módulo de dados. É possível usar o diagrama para examinar os relacionamentos, editar o módulo de dados e visualizar as informações de cardinalidade para os relacionamentos.

Clique com o botão direito em uma tabela no diagrama para visualizar o menu de contexto da tabela, que pode ser seu ponto de início para criar junções ou filtros, renomear a tabela, visualizar propriedades da tabela ou removê-la do módulo.

Clique em qualquer junção de tabela para ver informações de resumo da junção que incluam chaves de correspondência. Quando você clica com o botão direito na linha da junção, o menu de contexto aparece com opções para editar ou excluir a junção.

Selecione a caixa de seleção **Cardinalidade** para mostrar a cardinalidade dos relacionamentos entre as diferentes tabelas em seu módulo de dados. Mova a régua de controle **Graus de separação**. Dependendo da posição da régua de controle, o diagrama mostra diferentes distâncias de relacionamentos entre tabelas.

## Visualização de Dados

É possível usar a visualização de dados para examinar os dados reais nas colunas e linhas da tabela.

Selecione uma tabela na árvore de módulo de dados ou no diagrama, e clique no ícone de grade  para abrir a visualização de dados.

## Visualização de validação

É possível usar a visualização de validação para examinar erros que são identificados pelo processo de validação.

As mensagens são exibidas após iniciar a operação **Validar** em qualquer lugar na interface com o usuário de modelagem, e o ícone de validação com falha  é exibido para tabelas, colunas, expressões ou junções onde os erros são descobertos.

---

## Relacionamento

Um relacionamento associa objetos logicamente relacionados que os usuários desejam combinar em uma única consulta. Existem relacionamentos entre duas tabelas.

É possível modificar ou excluir relacionamentos ou criar relacionamentos novos para que o módulo de dados represente corretamente a estrutura lógica de seus negócios. Verifique se os relacionamentos requeridos existem no módulo de dados, se a cardinalidade está configurada corretamente e se a integridade referencial é forçada.

O diagrama fornece uma visualização gráfica dos relacionamentos de tabelas em um módulo de dados. É possível usar o diagrama para criar, examinar e editar os relacionamentos.

## Criando um relacionamento desde o início

É necessário criar relacionamentos sempre que os relacionamentos necessários não são detectados pelo software IBM Cognos.

### Sobre Esta Tarefa

Relacionamentos podem ser criados entre tabelas da mesma origem e de origens diferentes.

O diagrama é o local mais conveniente para visualizar todos os relacionamentos do módulo de dados e descobrir rapidamente as tabelas desconectadas.

**Importante:** A lista de chaves possíveis no editor de relacionamento exclui as medidas. Isso significa que se uma linha em uma coluna for identificada erroneamente como uma medida, mas você deseja usá-la como um identificador, a linha não aparecerá na lista suspensa de chaves. É necessário examinar o módulo de dados para confirmar se a propriedade de uso está correta em cada coluna na tabela.

### Procedimento

1. Na árvore de módulo de dados ou no diagrama, clique na tabela para a qual você deseja criar um relacionamento, e no menu de contexto, clique em **Criar relacionamento**.

**Dica:** Também é possível começar criando um relacionamento usando os métodos a seguir:

- Na árvore de módulo de dados ou no diagrama, com a tecla Control pressionada, clique nas duas tabelas que você deseja associar em um relacionamento e clique em **Criar relacionamento**.
- Na guia **Relacionamentos** nas propriedades da tabela, clique em **Criar um relacionamento**.

Se o módulo de dados não incluir a tabela da qual você precisa, é possível arrastar essa tabela de **Origens Selecionadas** diretamente sobre o diagrama.

2. No editor de relacionamento, especifique a segunda tabela para incluir no relacionamento e, em seguida, selecione as colunas correspondentes em ambas as tabelas.

Dependendo do método usado para iniciar o relacionamento, a segunda tabela já pode estar incluída, e você só precisará fazer a correspondência das colunas. É possível incluir mais de um conjunto de linhas correspondentes em ambas as tabelas.

3. Localize as colunas correspondentes em ambas as tabelas e selecione **Corresponder colunas selecionadas**.

4. Especifique as opções **Tipo de relacionamento**, **Cardinalidade** e **Otimização** para o relacionamento.
5. Clique em **OK**.

## Resultados

O novo relacionamento aparece na guia **Relacionamentos**, na página de propriedades das tabelas unidas, e na visualização do diagrama.

Para visualizar ou editar todos os relacionamentos definidos para uma tabela, acesse a guia **Relacionamentos** nas propriedades da tabela. Clique no link de relacionamento e faça as modificações. Para visualizar um relacionamento a partir do diagrama, clique na linha da junção para abrir uma pequena visualização gráfica do relacionamento. Para editar um relacionamento a partir do diagrama, clique com o botão direito na linha de junção e clique em **Editar relacionamento**.

Para excluir um relacionamento para uma tabela, acesse a guia **Relacionamentos** nas propriedades da tabela e clique no ícone Remover  para o relacionamento necessário. Para excluir o relacionamento do diagrama, clique com o botão direito na linha associando as duas tabelas e clique em **Remover**.

---

## Cálculos

Os cálculos permitem responder perguntas que não possam ser respondidas pelas colunas de origem.

Os seguintes recursos do produto são baseados em cálculos subjacentes:

- Cálculos aritméticos básicos e concatenações de campo.
- Grupos customizados.
- Limpeza de dados de texto.
- Cálculos customizados.

### Criando cálculos básicos

É possível criar cálculos aritméticos básicos para colunas com tipos de dados numéricos e concatenar valores de texto para colunas com tipo de dado de texto.

#### Sobre Esta Tarefa

A expressão para esses cálculos é predefinida e basta apenas selecioná-las. Por exemplo, é possível criar uma coluna Renda multiplicando os valores de Quantidade e Preço unitário. É possível criar uma coluna Nome combinando duas colunas: Nome e Sobrenome.

#### Procedimento

1. Para criar um cálculo aritmético simples para colunas com tipos de dados numéricos, use as seguintes etapas:
  - a. Na árvore de módulo de dados, clique com o botão direito na coluna para a qual deseja criar um cálculo. Para cálculos baseados em duas colunas, clique com Ctrl pressionado para selecionar as colunas.
  - b. Na caixa **Criar cálculo**, digite um nome para o cálculo.
  - c. Se o cálculo for baseado em uma coluna, digite o número a ser usado no cálculo.

**Dica:** O link **Usar editor de cálculo** abre o editor de expressão.

- d. Clique em **OK**.
2. Para criar um cálculo que concatene valores para colunas com o tipo de dado de texto, use as etapas a seguir:
  - a. Na árvore de módulo de dados, clique com Ctrl pressionado nas duas colunas que deseja combinar em uma única coluna. O valor da coluna selecionada em primeiro aparecerá no início da sequência combinada.
  - b. Clique em **Criar um cálculo** e selecione a opção sugerida.
  - c. Digite o nome do cálculo.
  - d. Clique em **OK**.

## Resultados

Na tabela em que foi incluído o cálculo agora é possível ver uma nova coluna calculada no final da lista de colunas.

## Agrupando Dados

É possível organizar os dados da coluna em grupos customizados para que seja mais fácil ler e organizar os dados.

### Sobre Esta Tarefa

É possível criar dois tipos de grupos customizados, dependendo do tipo de dados da coluna: um tipo de grupo para colunas com dados numéricos e um segundo tipo de grupo para colunas com dados de texto. Por exemplo, na coluna Código do funcionário, é possível agrupar os funcionários em intervalos, como 0 a 100, 101 a 200, mais de 200. Na coluna Gerente, é possível agrupar os gerentes de acordo com suas posições, como Gerente de primeira linha, Gerente sênior e assim por diante.

### Procedimento

1. Na árvore de módulo de dados, clique com o botão direito na coluna pela qual deseja agrupar e clique em **Grupos customizados**.
2. Se você selecionar uma coluna numérica, especifique o agrupamento da seguinte maneira:
  - a. Especifique quantos grupos que deseja criar.
  - b. Especifique a distribuição dos valores como **Distribuição igual** ou **Customizada**.
  - c. Se você escolher **Distribuição igual**, especifique os valores a serem contidos em cada grupo, digitando os números ou clicando nas barras de rolagem.
  - d. Se você escolher **Customizado**, será possível digitar seu próprio intervalo de valores para o grupo.
  - e. Opcional: Altere o nome do grupo.
  - f. Clique em **Criar**.
3. Se você selecionar uma coluna de texto, especifique o agrupamento da seguinte maneira:
  - a. Selecione com Ctrl pressionado os valores a serem incluídos no primeiro grupo.
  - b. Na coluna **Grupos**, clique no sinal de mais.
  - c. Especifique o nome para o grupo e clique em **OK**. Os valores serão incluídos na coluna **Membros do grupo** e o nome do grupo aparecerá na

coluna **Grupos**. É possível incluir valores adicionais em um grupo após sua criação, e também remover valores de um grupo. Também é possível remover um grupo.

- d. Opcional: Para incluir outro grupo, repita as etapas do primeiro grupo.
- e. Opcional: Para criar um grupo que contenha todos os valores que ainda não estejam incluídos em um grupo, marque a caixa de seleção **Agrupar valores restantes e futuros** e especifique um nome para o grupo.
- f. Clique em **Criar**.

## Resultados

A coluna de grupo customizado baseada em suas seleções aparece no final da lista de colunas na tabela. Uma expressão de grupo é criada automaticamente no editor de expressão. Para visualizar a expressão, acesse a página de propriedades da coluna, e clique na expressão que é mostrada para a propriedade **Expressão**.

**Dica:** Para concluir a ação de criação do grupo customizado, é possível clicar em **Substituir** em vez de em **Criar**. Essa opção substituirá o nome da coluna na tabela pelo nome do grupo.

## Limpendo dados

Os dados são sempre confusos e inconsistentes. Você pode desejar impor alguma ordem de formatação nos dados, para que fiquem mais claros e fáceis de ler.

### Sobre Esta Tarefa

As opções **Limpar** que estão disponíveis para uma coluna dependem do tipo de dados da coluna. Algumas opções podem ser especificadas para múltiplas colunas com o mesmo tipo de dados e somente para algumas colunas exclusivas.

As opções a seguir estão disponíveis para limpar os dados:

#### Espaço em Branco

##### Aparar espaços em branco do início e do final

Marque esta caixa de seleção para remover das sequências o espaço em branco à esquerda e à direita.

#### Converter casos para

##### MAIÚSCULA, minúscula, Não mude

Use esta opção para mudar da diferença entre maiúsculas e minúsculas de todos os caracteres na sequência para maiúscula ou minúscula, ou assegure-se de que a diferença entre maiúsculas e minúsculas de cada caractere individual seja inalterada.

#### Retornar uma subsequência de caracteres

Retorne uma sequência que inclua somente parte da sequência original em cada valor. Por exemplo, um código do funcionário pode ser armazenado como CA096670, mas é necessário somente o número 096670 para que essa opção seja usada para remover a parte CA. É possível especificar essa opção somente para colunas exclusivas.

Para o valor **Iniciar**, digite um número que represente a posição de um caractere na sequência que iniciará a subsequência. O número 1 representa o primeiro caractere na sequência. Para o valor **Comprimento**, especifique o número de caracteres que serão incluídos na subsequência.

## Valores NULL

### 11.0.4

Especifique as opções de manipulação de NULL para colunas com texto, numéricas, de data e de tipos de data e hora que permitem valores NULL. Quando o Cognos Analytics detectar que uma coluna não permite valores NULL, essas opções não estarão disponíveis para essa coluna.

O valor padrão para cada opção depende do tipo de dados da coluna. Para dados de texto, o padrão é uma sequência vazia. Para números, o padrão é 0. Para datas, o padrão é 2000-01-01. Para horário, o padrão é 12:00:00.

Para data e hora (registro de data e hora), o padrão é 2000-01-01T12:00:00.

O campo de entrada para cada opção também depende do tipo de dados da coluna. Para texto, o campo de entrada aceita caracteres alfanuméricos, para números, o campo de entrada aceita somente entrada numérica. Para datas, o selecionador de data é fornecido para selecionar a data, e para horário, um selecionador de horário é fornecido para selecionar a hora.

As opções de manipulação de NULL a seguir estão disponíveis:

#### **Substituir esse valor por NULL**

Substitui por NULL os valores do texto, números, data e hora, conforme você especifica no campo de entrada.

Por exemplo, se você desejar usar uma sequência vazia em vez de NULL em uma determinada coluna, mas o arquivo transferido por upload às vezes usa a sequência n/a para indicar que o valor é desconhecido, será possível substituir n/a por NULL e, em seguida, escolher substituir NULL pela sequência vazia.

#### **Substituir valores NULL por**

Substitui valores NULL pelos valores de texto, números, data e hora, conforme você especifica no campo de entrada.

Por exemplo, para a coluna Nome do Meio, é possível especificar os seguintes valores a serem usados para células em que o nome do meio não existe: n/a, none, ou a sequência vazia padrão. Para a coluna Quantia de Desconto, é possível especificar 0.00 para células em que a quantia é desconhecida.

## Procedimento

1. Na árvore de módulo de dados, clique no ícone de menu de contexto  para uma coluna, e clique em **Limpar**.

**Dica:** Para limpar dados em múltiplas colunas de uma vez, controle e selecione as colunas que deseja limpar. A opção **Limpar** estará disponível somente se o tipo de dados de cada coluna selecionada for o mesmo.

2. Especifique as opções aplicáveis para a coluna ou as colunas selecionadas.
3. Clique em **Limpar**.

## Resultados

Após concluir a operação **Limpar**, o editor de expressão criará automaticamente uma expressão para a coluna ou colunas modificadas. Para visualizar a expressão, abra o painel de propriedades da coluna e clique na expressão que é mostrada para a propriedade **Expressão**.

## Criando cálculos customizados

Para criar um cálculo customizado, deve-se definir sua própria expressão usando o editor de expressão.

### Sobre Esta Tarefa

Os cálculos customizados podem ser criados no nível do módulo de dados ou no nível da tabela. Os cálculos em nível de módulo podem fazer referência a colunas de várias tabelas.

Para obter informações sobre as funções que podem ser usadas para definir suas expressões, consulte Apêndice A, "Usando o Editor de Expressão", na página 27.

### Procedimento

1. Na árvore de módulo de dados, clique com o botão direito no nome do módulo de dados ou em um nome de tabela específico e clique em **Criar cálculo customizado**.
2. No painel **Editor de expressão**, defina a expressão para o cálculo e especifique um nome para ela.
  - Para inserir uma função para sua expressão, digite o primeiro caractere do nome da função e selecione a função da lista suspensa de funções sugeridas.
  - Para incluir colunas de tabela em sua expressão, arraste e solte uma ou mais colunas da árvore do módulo de dados para o painel do editor de expressão. O nome da coluna é incluído no local onde você coloca o cursor no editor de expressão.

**Dica:** Também é possível clicar duas vezes na coluna na árvore do módulo de dados e o nome da coluna aparece no editor de expressão.

3. Clique em **Validar** para verificar se a expressão é válida.
4. Após a validação bem-sucedida, clique em **OK**.

### Resultados

Se você criou o cálculo no nível do módulo de dados, o cálculo será incluído após a última tabela na árvore de módulo de dados. Se você criou o cálculo no nível de tabela, o cálculo será incluído no final da lista de colunas na tabela. Para visualizar a expressão para o cálculo, abra o painel de propriedades do cálculo e clique na expressão que é mostrada para a propriedade **Expressão**.

---

## Criando caminhos de navegação

Um caminho de navegação é uma coleção de colunas sem medida que os usuários de negócios podem associar para exploração de dados.

Quando um módulo de dados contém caminhos de navegação, os usuários do painel podem realizar drill down e voltar para mudar o foco da sua análise, percorrendo os níveis de informações. Os usuários podem realizar drill down de coluna para coluna no caminho de navegação seguindo a ordem das colunas no caminho de navegação, ou escolhendo a coluna na qual eles desejam continuar.

## Sobre Esta Tarefa

É possível criar um caminho de navegação com colunas que estão logicamente relacionadas, tais como ano, mês, trimestre, semana. Também é possível criar um caminho de navegação com colunas que não estão relacionadas logicamente, como produto, cliente, estado, cidade.

Colunas de tabelas diferentes podem ser incluídas em um caminho de navegação. A mesma coluna pode ser incluída em vários caminhos de navegação.

Um módulo de dados pode ter vários caminhos de navegação.

## Procedimento

1. No painel do módulo de dados, comece criando um caminho de navegação usando um dos seguintes métodos:

- No menu de contexto do módulo de dados , clique em **Propriedades** e, em seguida, clique na guia **Caminhos de navegação**. Clique em **Incluir um caminho de navegação**. Na caixa de diálogo **Criar caminho de navegação**, arraste colunas do painel do módulo de dados para o painel do caminho de navegação. Altere a ordem das colunas conforme necessário. Clique em **OK**.
- Na árvore de módulo de dados, selecione uma ou mais colunas, e no menu de contexto  de qualquer uma das colunas selecionadas, clique em **Criar caminho de navegação**. As colunas selecionadas são listadas na caixa de diálogo **Criar caminho de navegação**. Clique em **OK**.

**Dica:** O nome padrão do caminho de navegação inclui nomes da primeira e da última coluna no caminho. É possível mudar esse nome.

2. Salve o módulo de dados para preservar o caminho de navegação.

3. Para modificar um caminho de navegação, no menu de contexto do módulo de dados , clique em **Propriedades** e, em seguida, clique na guia **Caminhos de navegação**. Clique no link **Editar** para o caminho que você deseja modificar. Na caixa de diálogo **Editar caminho de navegação**, é possível fazer as seguintes modificações:

- Para incluir colunas diferentes, arraste as colunas do módulo de dados para o caminho de navegação. É possível selecionar múltiplas colunas e arrastá-las todas elas de uma vez.
- Para remover colunas, clique no ícone de remoção  para a coluna.
- Para mudar a ordem das colunas, arraste-as para cima ou para baixo.
- Para mudar o nome do caminho de navegação, sobrescreva o nome existente. O nome padrão reage à ordem mudada das colunas. Se você sobrescrever o nome padrão, ele não mudará mais quando a definição do grupo for modificada. O nome não pode estar em branco.

## Resultados

O caminho de navegação é incluído no módulo de dados e está disponível para os usuários em painéis e histórias. Se você selecionar a opção **Identificar membros do caminho de navegação**  na barra de ferramentas do módulo de dados, as colunas que são membros do caminho de navegação serão sublinhadas.

## O que Fazer Depois

O modelador pode modificar o caminho de navegação a qualquer momento e salvar novamente o módulo de dados.

Para visualizar o caminho de navegação ao qual uma coluna pertence, no menu de contexto da coluna , clique em **Propriedades > Caminhos de navegação**. Clique no nome do caminho de navegação para visualizar ou modificar sua definição.

Para visualizar todos os caminhos de navegação em um módulo de dados, no menu de contexto do módulo de dados , clique em **Propriedades > Caminhos de navegação**. Clique no nome do caminho de navegação para visualizar ou modificar sua definição. Para excluir um caminho de navegação, clique no ícone Remover  para o caminho.

---

## Filtrando Dados

Um filtro especifica as condições que as linhas devem atender para serem recuperadas de uma tabela.

### Sobre Esta Tarefa

O filtro é baseado em uma coluna específica em uma tabela, mas afeta a tabela inteira. Além disso, somente linhas que atendem aos critérios de filtro são recuperadas de outras tabelas.

É possível criar filtros em nível de tabela, o que permite incluir vários filtros de uma vez, ou em nível de coluna.

### Procedimento

1. Na árvore do módulo de dados ou no diagrama, localize a tabela para a qual deseja criar filtros.
2. Expanda a tabela no painel do módulo de dados, e no menu de contexto da coluna, clique em **Filtrar**.

**Dica:** Também é possível clicar com o botão direito na tabela no diagrama e clicar em **Gerenciar filtros** de lá.

3. Selecione os valores de filtro da seguinte forma:
  - a. Se o tipo de dados da coluna for um número inteiro, haverá duas opções para especificar os valores: **Intervalo** e **Itens individuais**. Ao escolher **Intervalo**, use a régua de controle para especificar os intervalos de valores. Quando escolher **Itens individuais**, selecione as caixas de seleção associadas aos valores.
  - b. Para colunas com tipos de dados numéricos que não sejam números inteiros, use a régua de controle para especificar os valores do intervalo.
  - c. Para colunas com tipos de dados de data e hora (registro de data e hora), especifique um intervalo de valores antes, depois ou entre a data e hora selecionadas ou selecione valores individuais.
  - d. Para colunas com tipos de dados de texto, marque as caixas de opções associadas aos valores.

4. Opcional: Para selecionar valores que estão fora do intervalo especificado, clique em **Inverter**.
5. Clique em **OK**.

## Resultados

Após a criação de um filtro, o ícone de filtro  é incluído para a tabela e a coluna no painel do módulo de dados e no diagrama.

## O que Fazer Depois

Para visualizar, editar ou remover os filtros definidos para uma tabela, selecione a opção do menu de contexto **Gerenciar filtros** para a tabela e clique na guia **Filtros** nas propriedades da tabela.

Para editar o filtro, clique em sua expressão, faça as modificações e clique em **OK**.

Para remover um filtro da tabela, selecione o ícone Remover  para o filtro.

**Dica:** Para editar um filtro em uma única coluna, no menu de contexto da coluna no painel do módulo de dados, clique em **Filtrar** para abrir a definição de filtro.

---

## Ocultando tabelas e colunas

### 11.0.4

É possível ocultar uma tabela ou coluna em um módulo de dados. As tabelas ou colunas ocultas permanecem visíveis na interface de modelagem, mas não ficam visíveis nas interfaces de relatórios e painéis. Os itens ocultos são totalmente funcionais no produto.

## Sobre Esta Tarefa

Use este recurso para fornecer uma visualização clara de metadados para os usuários de relatórios e painéis. Por exemplo, ao ocultar colunas referenciadas em um cálculo, a árvore de metadados nas interfaces de relatórios e painéis mostra somente a coluna de cálculo, mas não as colunas referenciadas. Ao ocultar as colunas de identificador usadas como chaves para junções, as chaves não são expostas nas interfaces de painéis e relatórios, mas as junções são funcionais em todas as interfaces.

## Procedimento

1. Na árvore de módulo de dados, clique no ícone de menu de contexto  para uma tabela ou coluna e clique em **Ocultar**.

Também é possível selecionar diversas tabelas ou colunas para ocultá-las de uma só vez.

**Dica:** Para reexibir os itens, clique no ícone de menu de contexto para a tabela ou coluna oculta e clique em **Mostrar**.

2. Salve o módulo de dados.

## Resultados

Os rótulos nas tabelas ou colunas ocultas são esmaecidos na árvore de módulo de dados e no diagrama. Além disso, na guia **Geral** das propriedades da tabela ou coluna, a caixa de seleção **Este item está oculto dos usuários** é marcada.

As tabelas ou colunas ocultas não são visíveis nas interfaces de relatórios e painéis.

---

## Validando módulos de dados

Use o recurso de validação para verificar referências de objeto inválido dentro de um módulo de dados.

### Sobre Esta Tarefa

A validação identifica os erros a seguir:

- Uma tabela ou coluna na qual um módulo de dados se baseia não existe mais na origem.
- Uma expressão de cálculo é inválida.
- Um filtro faz referência a uma coluna que não existe mais no módulo de dados.
- Uma tabela ou coluna que é referenciada em uma junção não existe mais no módulo de dados.

Os erros no módulo de dados são identificados pelo ícone de validação com falha



### Procedimento

1. Na árvore de módulo de dados, clique no ícone de menu de contexto do módulo de dados  e clique em **Validar**

Se os erros forem identificados, o ícone de validação com falha  será exibido na árvore de módulo de dados, no diagrama e no painel de propriedades, ao lado da coluna ou expressão onde o erro existe. As descrições de erros são exibidas na visualização de validação.

**Dica:** Para abrir a visualização de validação, clique em seu ícone .

2. Clique no ícone de validação com falha  para um módulo, coluna, expressão ou junção para visualizar uma caixa pop-up que informa sobre o número de erros para o item selecionado. Dê um clique duplo no ícone de validação com falha  na caixa pop-up para visualizar os detalhes do erro.

### Resultados

Usando as mensagens de validação, tente resolver os erros. É possível salvar um módulo de dados com erros de validação nele.

---

## Propriedades de tabela e coluna

É possível visualizar e modificar as propriedades de tabela e coluna em um módulo de dados.

As propriedades podem ser acessadas a partir do menu de contexto da tabela ou coluna  na guia **Geral** do painel **Propriedades**.

## Rótulo

Especifica o nome da tabela ou coluna. É possível mudar o nome, conforme necessário.

## Este item está oculto dos usuários

Use esta propriedade para ocultar uma tabela ou coluna em um módulo de dados. As tabelas ou colunas ocultas permanecem visíveis na interface de modelagem, mas não são visíveis nas interfaces de relatórios e de painéis. Para obter mais informações, consulte “Ocultando tabelas e colunas” na página 21.

## Expressão

Mostra a expressão subjacente para uma coluna. Clicar na expressão abre o editor de expressão, no qual é possível modificar a expressão.

## Comentários

Utilize esta propriedade para especificar informações opcionais sobre a tabela ou coluna. O comentário não está disponível fora do ambiente de modelagem.

## Dica de tela

Use esta propriedade para especificar uma descrição opcional simples da tabela ou coluna. A dica de tela aparece quando se coloca o ponteiro sobre o nome da tabela ou coluna no ambiente de modelagem, de relatório ou de painéis.

## Uso

Essa propriedade identifica o uso desejado para os dados na coluna.

O valor da propriedade inicial é baseado no tipo de dados que a coluna representa na origem. É necessário verificar se a propriedade está configurada corretamente. Por exemplo, se você importar uma coluna numérica que participa de um relacionamento, a propriedade **Uso** será configurada como **Identificador**. É possível mudar esta propriedade.

Os tipos de **Uso** a seguir são suportados:

- **Identificador**

Representa uma coluna que é usada para agrupar ou resumir dados em uma coluna **Medida** com a qual eles têm um relacionamento. Ele também pode representar um tipo de coluna, como índice, data ou hora. Por exemplo, Número da fatura ou Data da fatura.

- **Medida**

Representa uma coluna que contém dados numéricos que podem ser agrupados ou resumidos, como Custo do produto.

- **Atributo**

Representa uma coluna que não é um **Identificador** ou uma **Medida**, como Descrição.

## Agregação

A propriedade **Agregado** define o tipo de agregação que é aplicado a uma coluna que resume os dados em um relatório ou painel. Por exemplo, quando o valor da propriedade **Agregação** da coluna Quantidade é **Total** e é agrupado pelo Nome do produto em um relatório, a coluna Quantidade no relatório mostra a quantidade total de cada produto. Os dados agregados melhoram o desempenho da consulta e ajuda a recuperar dados mais rapidamente.

O tipo de agregação nova vinculação é herdado da origem. Ao modificar essa propriedade, é possível selecionar valores que a origem não fornece, como média ou máxima. Para saber qual valor agregado é necessário, deve-se entender o que os dados representam. Por exemplo, se você agrega Número da peça, os valores agregados que se aplicam são contagem, contagem distinta, máximo e mínimo.

Os tipos de agregação a seguir são suportados:

- Nenhum (nenhuma agregação está configurada para uma coluna)
- Médio
- Contar
- Contagem distinta
- Máximo
- Mínimo
- Total

## Tipo de dado

O tipo de dados da coluna é herdado da origem e não pode ser modificado no módulo de dados.

## Representa

Use esta propriedade para especificar se uma coluna inclui a data/hora ou um tipo de local geográfico de dados. Essas informações são usadas nos ambientes de relatórios e de painéis que sugerem visualizações padrão mais apropriadas, entre outras possibilidades.

- Localização geográfica

Os valores incluem **Continente, Subcontinente, País, Região, Estado ou município, Condado, Cidade, Código postal, Endereço da rua, Posição, Latitude e Longitude.**

- Tempo

Os valores incluem **Data, Ano, Trimestre, Período, Mês, Semana, Dia, Hora, Minuto e Segundo.**

## Classificando

Utilize esta propriedade para ativar ou desativar a classificação de uma coluna e para especificar a linha pela qual classificar a ordem de classificação e a colocação de valores NULL na coluna.

## **Identificador**

Para tabelas e colunas, o valor da propriedade é herdado da origem e não pode ser mudado no módulo de dados. O valor de **Identificador** da coluna ou tabela pode ser, mas nem sempre, igual ao nome da coluna ou tabela (**Rótulo**).

É possível visualizar a propriedade **Identificador** na seção **Avançado** do painel **Propriedades**.

## **Fonte**

Mostra o nome da origem e o caminho para uma tabela ou coluna. É possível visualizar a propriedade **Origem** na seção **Avançado** do painel **Propriedades**



---

## Apêndice A. Usando o Editor de Expressão

Uma expressão é qualquer combinação de operadores, constantes, funções e outros componentes que são avaliados para um único valor. Você constrói expressões para criar definições de cálculos e filtros. Um cálculo é uma expressão usada para criar um novo valor a partir dos valores existentes que estão contidos em um item de dados. Um filtro é uma expressão que você usa para recuperar um subconjunto específico de registros.

---

### Operadores

Os operadores especificam o que acontece com os valores em ambos os lados do operador. Os operadores são semelhantes às funções, porque manipulam itens de dados e retornam um resultado.

(

Identifica o começo de uma expressão.

**Sintaxe**

( expression )

)

Identifica o final de uma expressão.

**Sintaxe**

( expression )

\*

Multiplica dois valores numéricos.

**Sintaxe**

valor1 \* valor2

/

Divide dois valores numéricos.

**Sintaxe**

valor1 / valor2

||

Concatena, ou associa, strings.

**Sintaxe**

string1 || string2

+

Adiciona dois valores numéricos.

### Sintaxe

valor1 + valor2

-

Subtrai dois valores numéricos ou torna negativo um valor numérico.

### Sintaxe

value1 - value2

ou

- value

<

Compara os valores representados por "valor1" e "valor2" e recupera os valores menores que "valor2".

### Sintaxe

valor1 < valor2

<=

Compara os valores representados por "valor1" e "valor2" e recupera os valores menores ou iguais a "valor2".

### Sintaxe

valor1 <= valor2

<>

Compara os valores representados por "valor1" e "valor2" e recupera os valores diferentes de "valor2".

### Sintaxe

valor1 <> valor2

=

Compara os valores representados por "valor1" e "valor2" e recupera os valores iguais a "valor2".

### Sintaxe

valor1 = valor2

>

Compara os valores representados por "valor1" e "valor2" e recupera os valores maiores que "valor2".

### Sintaxe

valor1 > valor2

>=

Compara os valores representados por "valor1" e "valor2" e recupera os valores maiores ou iguais a "valor2".

### Sintaxe

valor1 >= valor2

## e

Retorna "verdadeiro" se as condições nos dois lados da expressão forem verdadeiras.

### Sintaxe

argumento1 and argumento2

## meio

Determina se um valor enquadra-se em um determinado intervalo.

### Sintaxe

expressão between valor1 and valor2

### Exemplo

[Revenue] between 200 and 300

Resultado

Retorna o número de resultados com receitas entre 200 e 300.

Resultado de dados

Receita	Entre
\$332.06	false
\$230.55	true
\$107.94	false

## tipo de letra

Trabalha com when, then, else e end. Case identifica o começo de uma situação específica, na qual as ações when, then e else estão definidas.

### Sintaxe

```
case expression { when expression then expression } [ else expression ] end
```

## contém

Determina se a "string1" contém a "string2".

### Sintaxe

string1 contains string2

## distinct

Palavra-chave utilizada em uma expressão agregada para incluir somente ocorrências distintas de valores. Consulte também a função unique.

### Sintaxe

distinct dataItem

## Exemplo

```
count ( distinct [OrderDetailQuantity] )
```

Resultado

1704

## else

Trabalha com construções if ou case. Se a condição if ou a expressão case não forem verdadeiras, a expressão else será utilizada.

### Sintaxe

```
if ( condition ) then .... else ( expression ) , or case .... else ( expression ) end
```

## termina

Indica o final de uma construção case ou when.

### Sintaxe

```
case .... termina
```

## terminar por

Determina se a "string1" termina com "string2".

### Sintaxe

```
string1 ends with string2
```

## se

Trabalha com construções then e else. If define uma condição. Se a condição if for verdadeira, a expressão then será utilizada. Se a condição if não for verdadeira, a expressão else será utilizada.

### Sintaxe

```
if ( condition ) then ( expression ) else ( expression )
```

## in

Determina se a "expressão1" existe em uma lista de expressões.

### Sintaxe

```
expression1 in ( expression_list )
```

## está faltando

Determina se um "valor" está indefinido nos dados.

### Sintaxe

```
valor is missing
```

## like

Determina se a "string1" corresponde ao padrão da "string2", com o caractere "char" usado opcionalmente para os caracteres de escape na sequência padrão.

## Sintaxe

string1 LIKE string2 [ ESCAPE char ]

### Exemplo 1

```
[PRODUCT_LINE] like 'G%'
```

Resultado

Todas as linhas de produto que iniciam com 'G'.

### Exemplo 2

```
[PRODUCT_LINE] like '%G%' escape 'a'
```

Resultado

Todas as linhas de produto que terminam com 'G%'.

## procurar

Localiza e substitui dados por um valor que você especifica. É preferível utilizar a construção case.

### Sintaxe

```
lookup ( name ) in ( value1 --> value2 ) default ( expression )
```

### Exemplo

```
lookup ( [Country] ) in ( 'Canada'--> ( [List Price] * 0.60),  
'Australia'--> ( [List Price] * 0.80 ) ) default ( [List Price] )
```

## não

Retorna TRUE se o "argumento" for falso ou retorna FALSE se o "argumento" for verdadeiro.

### Sintaxe

```
NOT argumento
```

## ou

Retorna TRUE caso o "argumento1" ou o "argumento2" sejam verdadeiros.

### Sintaxe

```
argumento1 or argumento2
```

## começa com

Determina se a "string1" inicia-se com "string2".

### Sintaxe

```
string1 starts with string2
```

## then

Trabalha com construções if ou case. Se a condição if ou a expressão case forem verdadeiras, a expressão else será utilizada.

## Sintaxe

```
if ( condition ) then ..., or case expression when expression  
then .... termina
```

## quando

Trabalha com a construção case. É possível definir quais condições ocorrerão se a expressão WHEN for verdadeira.

## Sintaxe

```
case [expressão] when ... end
```

---

## Resumos

Esta lista contém funções pré-definidas que retornam ou um único valor de resumo para um grupo de valores relacionados ou um valor de resumo diferente para cada instância de um grupo de valores relacionados.

## Funções estatísticas

Esta lista contém funções de resumo predefinidas de natureza estatística.

### Desvio Padrão

Retorna o desvio padrão dos itens de dados selecionados.

#### Sintaxe

```
standard-deviation ( expression [ auto ] )  
standard-deviation ( expression for [ all|any ] expression { ,  
expression } )  
standard-deviation ( expression for report )
```

#### Exemplo

```
standard-deviation ( ProductCost )
```

Resultado

Retorna um valor indicando o desvio entre os custos do produto e o custo médio do produto.

### variance

Retorna a variação dos itens de dados selecionados.

#### Sintaxe

```
variance ( expression [ auto ] )  
variance ( expression for [ all|any ] expression { , expression } )  
variance ( expression for report )
```

#### Exemplo

```
variance ( Product Cost )
```

Resultado

Retorna um valor indicando o grau de variação dos custos do produto em relação aos custos médios.

## médio

Retorna a média do valor dos itens de dados selecionados. Distinct é uma expressão alternativa que é compatível com versões anteriores do produto.

### Sintaxe

```
average ( [ distinct ] expression [ auto ] )  
average ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
average ( [ distinct ] expression for report )
```

### Exemplo

```
average ( Sales )
```

Resultado

Retorna a média de todos os valores de Vendas.

## contar

Retorna o número de itens de dados selecionados, com exceção de valores nulos. Distinct é uma expressão alternativa que é compatível com versões anteriores do produto. Todas as versões são suportadas apenas no modo DQM e isso evita a presunção de uma contagem dupla de itens de dados de uma tabela de dimensões.

### Sintaxe

```
count ( [ all | distinct ] expression [ auto ] )  
count ( [ all | distinct ] expression for [ all|any ] expression { ,  
    expression } )  
count ( [ all | distinct ] expression for report )
```

### Exemplo

```
count ( Sales )
```

Resultado

Retorna o número total de entradas em Vendas.

## máximo

Retorna o valor máximo dos itens de dados selecionados. Distinct é uma expressão alternativa que é compatível com versões anteriores do produto.

### Sintaxe

```
maximum ( [ distinct ] expression [ auto ] )  
maximum ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
maximum ( [ distinct ] expression for report )
```

### Exemplo

```
maximum ( Sales )
```

Resultado

Retorna o valor máximo de todos os valores de Vendas.

## Mediano

Retorna o valor mediano dos itens de dados selecionados.

### Sintaxe

```
median ( expression [ auto ] )  
median ( expression for [ all|any ] expression { , expression } )  
median ( expression for report )
```

## minimum

Retorna o valor mínimo dos itens de dados selecionados. Distinct é uma expressão alternativa que é compatível com versões anteriores do produto.

### Sintaxe

```
minimum ( [ distinct ] expression [ auto ] )  
minimum ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
minimum ( [ distinct ] expression for report )
```

### Exemplo

```
minimum ( Sales )
```

Resultado

Retorna o valor mínimo de todos os valores de Vendas.

## percentage

Retorna o percentual do valor total para os itens de dados selecionados. A "<opção for>" define o escopo da função. A opção "at" define o nível de agregação e só é possível utilizá-la no contexto de origens de dados relacionais.

### Sintaxe

```
percentage ( numeric_expression [ at expression { , expression } ]  
    [ <for-option> ] [ prefilter ] )  
percentage ( numeric_expression [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression } | for report | auto
```

### Exemplo

```
percentage ( Sales 98 )
```

Resultado

Retorna a porcentagem do total de vendas para 1998 atribuída a cada representante de vendas.

Resultado de dados

Funcionário	Sales 98	Percentage
Gibbons	60646	7.11%
Flertjan	62523	7.35%
Corne1	22396	2.63%

## percentil

Retorna um valor, em uma escala de cem, que indica a porcentagem de uma distribuição que seja igual ou superior aos itens de dados selecionados. A "<opção

for>" define o escopo da função. A opção "at" define o nível de agregação e só é possível utilizá-la no contexto de origens de dados relacionais.

### Sintaxe

```
percentile ( numeric_expression [ at expression { , expression } ]  
[ <for-option> ] [ prefilter ] )  
percentile ( numeric_expression [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

### Exemplo

```
percentile ( Sales 98 )
```

Resultado

Para cada linha, retorna a porcentagem de linhas iguais ou menores do que o valor quantitativo da linha.

Resultado de dados

Qty	Percentile (Qty)
800	1
700	0.875
600	0,75
500	0,625
400	0.5
400	0.5
200	0,25
200	0,25

## quantil

Retorna a posição de um valor de um intervalo especificado. Retorna números inteiros para representar qualquer intervalo de classificações, como 1 (mais alto) a 100 (mais baixo). A "<opção for>" define o escopo da função. A opção "at" define o nível de agregação e só é possível utilizá-la no contexto de origens de dados relacionais.

### Sintaxe

```
quantile ( numeric_expression , numeric_expression [ at expression { ,  
expression } ] [ <for-option> ] [ prefilter ] )  
quantile ( numeric_expression , numeric_expression [ <for-option> ]  
[ prefilter ] )  
<for-option> ::= for expression { , expression }|for report|auto
```

### Exemplo

```
quantile ( Qty , 4 )
```

Resultado

Retorna a quantidade, a posição do valor da quantidade e os valores de quantidade divididos em 4 grupos quantis (quartis).

Resultado de dados

Qty	Rank	Quantile (Qty, 4)
800	1	1

Qty	Rank	Quantile (Qty, 4)
700	2	1
600	3	2
500	4	2
400	5	3
400	5	3
200	7	4
200	7	4

## quartil

Retorna a classificação de um valor, representado por números inteiros de 1 (máximo) a 4 (mínimo), com relação a um grupo de valores. A "<opção for>" define o escopo da função. A opção "at" define o nível de agregação e só é possível utilizá-la no contexto de origens de dados relacionais.

### Sintaxe

```
quartile ( numeric_expression [ at expression { , expression } ]
[ <for-option> ] [ prefilter ] )
quartile ( numeric_expression [ <for-option> ] [ prefilter ] )
<for-option> ::= for expression { , expression } | for report | auto
```

### Exemplo

```
quartile ( Qty )
```

#### Resultado

Retorna a quantidade e o quartil do valor de quantidade representados por números inteiros de 1 (mais alta) a 4 (mais baixa).

#### Resultado de dados

Qty	Quartil (Qtd)
450	1
400	1
350	2
300	2
250	3
200	3
150	4
100	4

## rank

Retorna o valor de classificação dos itens de dados selecionados. A ordem de classificação é opcional; a ordem decrescente (DESC) é assumida como padrão. Se uma ou mais linhas se unirem, então haverá um espaço entre a sequência dos valores classificados (também conhecido como classificação olímpica). A "<opção for>" define o escopo da função. A opção "at" define o nível de agregação e só é possível utilizá-la no contexto de origens de dados relacionais. Distinct é uma expressão alternativa que é compatível com versões anteriores do produto. Os valores nulos são classificados em último.

## Sintaxe

```
rank ( expression [ ASC|DESC ] { , expression [ ASC|DESC ] } [ at  
expression { , expression } ] [ <for-option> ] [ prefilter ] )  
rank ( [ distinct ] expression [ ASC|DESC ] { , expression  
[ ASC|DESC ] } [ <for-option> ] [ prefilter ] )  
<for-option> ::= for expression { , expression } | for report | auto
```

## Exemplo

```
rank ( Sales 98 )
```

### Resultado

Para cada linha, retorna o valor de posição das vendas para 1998 atribuído a cada representante de vendas. Alguns números são pulados quando ocorre uma ligação entre linhas.

### Resultado de dados

Funcionário	Sales 98	Rank
Gibbons	60000	1
Flertjan	50000	2
Cornel	50000	2
Smith	48000	4

## tertile

Retorna a posição de um valor como Alto, Intermediário ou Baixo em relação a um grupo de valores.

## Sintaxe

```
tertile ( expression [ auto ] )  
tertile ( expression for [ all|any ] expression { , expression } )  
tertile ( expression for report )
```

## Exemplo

```
tertile ( Qty )
```

### Resultado

Retorna a quantidade, o valor da posição de quantil da quantidade dividido em tertis e o rótulo da posição de quantil dividido em tertis.

### Resultado de dados

Qty	Quantile (Qty, 3)	Tertile (Qty)
800	1	H
700	1	H
500	2	M
400	2	M
200	3	C
200	3	C

## total

Retorna o valor total dos itens de dados selecionados. Distinct é uma expressão alternativa que é compatível com versões anteriores do produto.

## Sintaxe

```
total ( [ distinct ] expression [ auto ] )  
total ( [ distinct ] expression for [ all|any ] expression { ,  
    expression } )  
total ( [ distinct ] expression for report )
```

## Exemplo

```
total ( Sales )
```

Resultado

Retorna o valor total de todos os valores de Vendas.

---

## Funções de data/hora comerciais

Esta lista contém funções de negócios para execução de cálculos de data e hora.

### \_add\_seconds

Retorna o horário ou a data/hora, dependendo do formato de "time\_expression", resultante da inclusão de "integer\_expression" segundos em "time\_expression".

#### Sintaxe

```
_add_seconds ( time_expression, integer_expression )
```

#### Exemplo 1

```
_add_seconds ( 13:04:59 , 1 )
```

Resultado

13:05:00

#### Exemplo 2

```
_add_seconds ( 2002-04-30 12:10:10.000, 1 )
```

Resultado

2002-04-30 12:10:11.000

#### Exemplo 3

```
_add_seconds ( 2002-04-30 00:00:00.000, 1/100 )
```

Observe que o segundo argumento não é um número inteiro. Isso é suportado por algumas tecnologias de banco de dados e incrementa a parte do tempo.

Resultado

2002-04-30 00:00:00.010

### \_add\_minutes

Retorna o horário ou a data/hora, dependendo do formato de "time\_expression", resultante da inclusão de "integer\_expression" minutos em "time\_expression".

#### Sintaxe

```
_add_minutes ( time_expression, integer_expression )
```

### **Exemplo 1**

```
_add_minutes ( 13:59:00 , 1 )
```

Resultado

14:00:00

### **Exemplo 2**

```
_add_minutes ( 2002-04-30 12:59:10.000, 1 )
```

Resultado

2002-04-30 13:00:10.000

### **Exemplo 3**

```
_add_minutes ( 2002-04-30 00:00:00.000, 1/60 )
```

Observe que o segundo

argumento não é um número inteiro. Isso é suportado por algumas tecnologias de banco de dados e incrementa a parte do tempo.

Resultado

2002-04-30 00:00:01.000

## **\_add\_hours**

Retorna o horário ou a data/hora, dependendo do formato de "time\_expression", resultante da inclusão de "integer\_expression" horas em "time\_expression".

### **Sintaxe**

```
_add_hours ( time_expression, integer_expression )
```

### **Exemplo 1**

```
_add_hours ( 13:59:00 , 1 )
```

Resultado

14:59:00

### **Exemplo 2**

```
_add_hours ( 2002-04-30 12:10:10.000, 1 )
```

Resultado

2002-04-30 13:10:10.000,

### **Exemplo 3**

```
_add_hours ( 2002-04-30 00:00:00.000, 1/60 )
```

Observe que o segundo

argumento não é um número inteiro. Isso é suportado por algumas tecnologias de banco de dados e incrementa a parte do tempo.

Resultado

2002-04-30 00:01:00.000

## **\_add\_days**

Retorna a data ou data/hora, dependendo do formato da "date\_expression", que resulta da adição de "integer\_expression" dias a "date\_expression".

### **Sintaxe**

```
_add_days ( date_expression, integer_expression )
```

### **Exemplo 1**

```
_add_days ( 2002-04-30 , 1 )
```

Resultado

01-05-2002

### **Exemplo 2**

```
_add_days ( 2002-04-30 12:10:10.000, 1 )
```

Resultado

2002-05-01 12:10:10.000

### **Exemplo 3**

```
_add_days ( 2002-04-30 00:00:00.000, 1/24 )
```

Observe que o segundo argumento não é um número inteiro. Isso é suportado por algumas tecnologias de banco de dados e incrementa a parte do tempo.

Resultado

2002-04-30 01:00:00.000

## **\_add\_months**

Inclui meses de "integer\_expression" para "date\_expression". Se o mês resultante tiver menos dias do que o mês componente, então o último dia do mês resultante é retornado. Em outros casos, o valor retornado tem o mesmo dia do mês componente que o "date\_expression".

### **Sintaxe**

```
_add_months ( date_expression, integer_expression )
```

### **Exemplo 1**

```
_add_months ( 2012-04-15 , 3 )
```

Resultado

2012-07-15

### **Exemplo 2**

```
_add_months ( 2012-02-29 , 1 )
```

Resultado

2012-03-29

### **Exemplo 3**

```
_last_of_month ( _add_months ( 2012-02-29 , 1 ) )
```

Resultado

2012-03-31

### **Exemplo 4**

```
_add_months ( 2012-01-31 , 1 )
```

Resultado

29-02-2012

### **Exemplo 5**

```
_add_months ( 2002-04-30 12:10:10.000 , 1 )
```

Resultado

2002-05-30 12:10:10.000

## **\_add\_years**

Inclui anos de "integer\_expression" para "date\_expression". Se o "date\_expression" for 29 de fevereiro e o ano resultante não for um ano bissexto, então o dia resultante é configurado para 28 de fevereiro. Em outros casos, o valor retornado tem o mesmo dia e o mesmo mês que o "date\_expression".

### **Sintaxe**

```
_add_years ( date_expression, integer_expression )
```

### **Exemplo 1**

```
_add_years ( 2012-04-15 , 1 )
```

Resultado

15-04-2013

### **Exemplo 2**

```
_add_years ( 2012-02-29 , 1 )
```

Resultado

2013-02-28

### **Exemplo 3**

```
_add_years ( 2002-04-30 12:10:10.000 , 1 )
```

Resultado

2003-04-30 12:10:10.000

## **`_age`**

Retorna um número obtido por meio da subtração de "date\_expression" da data de hoje. O valor retornado tem a forma AAAAMMDD, em que AAAA representa o número de anos, MM representa o número de meses e DD representa o número de dias.

### **Sintaxe**

```
_age ( date_expression )
```

### **Exemplo**

```
_age ( 1990-04-30 ) (if today's date is 2003-02-05)
```

Resultado

120906, ou seja, 12 anos, 9 meses e 6 dias.

## **`current_date`**

Retorna um valor que representa a data atual do computador em que o software de banco de dados está em execução.

### **Sintaxe**

```
current_date
```

### **Exemplo**

```
current_date
```

Resultado

04-03-2003

## **`current_time`**

Retorna um horário com valor de fuso horário, representando o horário atual do computador que executa o software de banco de dados se o banco de dados suportar essa função. Caso contrário, ele representará a hora atual do servidor IBM Cognos Analytics.

### **Sintaxe**

```
current_time
```

### **Exemplo**

```
current_time
```

Resultado

16:33:11.354+05:00

## **`current_timestamp`**

Retorna uma data/hora com valor de fuso horário, representando o horário atual do computador que executa o software de banco de dados se o banco de dados suportar essa função. Caso contrário, ele representará a hora atual do servidor.

### **Sintaxe**

```
current_timestamp
```

## Exemplo

```
current_timestamp
```

Resultado

```
03-03-2003 16:40:15.535+05:00
```

## **\_day\_of\_week**

Retorna o dia da semana ( entre 1 e 7), em que 1 é o primeiro dia da semana, conforme indicado pelo segundo parâmetro (entre 1 e 7; 1 sendo que 1 é segunda e 7; domingo). Observe que no padrão ISO 8601, uma semana começa com a segunda como dia 1.

### Sintaxe

```
_day_of_week ( date_expression, integer )
```

### Exemplo

```
_day_of_week ( 2003-01-01 , 1 )
```

Resultado

```
3
```

## **\_day\_of\_year**

Retorna o dia do ano (1 a 366) na "date\_expression". Também conhecido como calendário juliano.

### Sintaxe

```
_day_of_year ( date_expression )
```

### Exemplo

```
_day_of_year ( 2003-03-01 )
```

Resultado

```
61
```

## **\_days\_between**

Retorna números positivos ou negativos que representam o número de dias entre "date\_expression1" e "date\_expression2". Se "date\_expression1" < "date\_expression2"; o resultado será um número negativo.

### Sintaxe

```
_days_between ( date_expression1 , date_expression2 )
```

### Exemplo

```
_days_between ( 2002-04-30 , 2002-06-21 )
```

Resultado

```
-52
```

## **\_days\_to\_end\_of\_month**

Retorna números que representam o número de dias restantes no mês representado pela expressão "date\_expression".

### **Sintaxe**

```
_days_to_end_of_month ( date_expression )
```

### **Exemplo**

```
_days_to_end_of_month ( 2002-04-20 14:30:22.123 )
```

Resultado

10

## **\_end\_of\_day**

Retorna o término de hoje como um registro de data e hora.

### **Sintaxe**

```
_end_of_day
```

### **Exemplo**

```
_end_of_day
```

Resultado

2014-11-23 23:59:59

## **\_first\_of\_month**

Retorna uma data ou data/hora, dependendo do argumento, pela conversão da "date\_expression" para uma data com o mesmo ano e mês, mas o dia configurado como 1.

### **Sintaxe**

```
_first_of_month ( date_expression )
```

### **Exemplo 1**

```
_first_of_month ( 2002-04-20 )
```

Resultado

01-04-2002

### **Exemplo 2**

```
_first_of_month ( 2002-04-20 12:10:10.000 )
```

Resultado

2002-04-01 12:10:10.000

## **\_from\_unixtime**

Retorna a hora do UNIX especificada por uma expressão de número inteiro como um registro de data e hora com fuso horário.

### **Sintaxe**

```
_from_unixtime ( integer_expression )
```

### **Exemplo**

```
_from_unixtime ( 1417807335 )
```

Resultado

```
05-12-2014 19:22:15+00:00
```

## **\_hour**

Retorna o valor do campo de hora em uma expressão de data.

### **Sintaxe**

```
_hour( date_expression )
```

### **Exemplo**

```
_hour ( 2002-01-31 12:10:10.254 )
```

Resultado

```
12
```

## **\_last\_of\_month**

Retorna a data ou data/hora, dependendo do argumento, que é o último dia do mês representado por "date\_expression".

### **Sintaxe**

```
_last_of_month ( date_expression )
```

### **Exemplo 1**

```
_last_of_month ( 2002-01-14 )
```

Resultado

```
31-01-2002
```

### **Exemplo 2**

```
_last_of_month ( 2002-01-14 12:10:10.000 )
```

Resultado

```
2002-01-31 12:10:10.000
```

## **\_make\_timestamp**

Retorna um registro de data e hora construído a partir de "integer\_expression1" (o ano), "integer\_expression2" (o mês) e "integer\_expression" (o dia). O padrão da fração de tempo é 00:00:00.000.

### **Sintaxe**

```
_make_timestamp ( integer_expression1, integer_expression2,  
integer_expression3 )
```

### **Exemplo**

```
_make_timestamp ( 2002 , 01 , 14 )
```

Resultado

```
2002-01-14 00:00:00.000
```

### **\_minute**

Retorna o valor do campo de minuto em uma expressão de data.

#### **Sintaxe**

```
_minute( date_expression )
```

#### **Exemplo**

```
_minute ( 2002-01-31 12:10:10.254 )
```

Resultado

```
10
```

### **\_month**

Retorna o valor do campo de mês em uma expressão de data.

#### **Sintaxe**

```
_month( date_expression )
```

#### **Exemplo**

```
_month ( 2003-03-01 )
```

Resultado

```
3
```

### **\_months\_between**

Retorna números inteiros positivos ou negativos que representam o número de meses entre "date\_expression1" e "date\_expression2". Se a "date\_expression1" for anterior à "date\_expression2", um número negativo será retornado.

#### **Sintaxe**

```
_months_between ( date_expression1, date_expression2 )
```

#### **Exemplo**

```
_months_between ( 2002-04-03 , 2002-01-30 )
```

Resultado

```
2
```

### **\_second**

Retorna o valor do campo de segundo em uma expressão de data.

#### **Sintaxe**

```
_second( date_expression )
```

## Exemplo

```
_second ( 2002-01-31 12:10:10.254 )
```

Resultado

10.254

## \_shift\_timezone

Desloca um registro de data e hora de um fuso horário para outro fuso horário. Essa função aceita o Horário de Verão quando aplicável. Se o primeiro argumento for do tipo "registro de data e hora", então, o segundo e o terceiro argumentos representam os fusos horários "a partir de" e "destino", respectivamente. Se o primeiro argumento for do tipo "registro de data e hora com fuso horário", então, o fuso horário "a partir de" já estará implícito no primeiro argumento; portanto, o segundo argumento representará o fuso horário "destino". O tipo de dados do primeiro argumento também determinará o tipo de dados do valor de retorno. O segundo e terceiro argumentos são do tipo "sequência" e representam identificadores de fuso horário. Uma lista desses identificadores pode ser localizada abaixo. Nota: usar essa função causará processamento local.

## Sintaxe

```
_shift_timezone ( timestamp_value , from_time_zone ,  
                  target_time_zone )  
_shift_timezone ( timestamp_with_time_zone_value , target_time_zone )
```

## Exemplo 1

```
_shift_timezone( 2013-06-30 12:00:00 , 'EST' , 'GMT' )
```

Resultado

2013-06-30 16:00:00

## Exemplo 2

```
_shift_timezone( 2013-11-30 12:00:00-05:00 , 'PST' )
```

Resultado

2013-11-30 09:00:00-08:00

## Exemplo 3

Abreviações de fuso horário:

Resultado de dados

GMT (GMT+00:00) Greenwich Mean Time  
UTC (GMT+00:00) Coordinated Universal Time  
WET (GMT+00:00) Western Europe Time: Lisbon, Faeroe Islands, Canary Ilhas  
ECT (GMT+01:00) European Central Time: Amsterdam, Brussels, Paris, Rome, Vienna  
MET (GMT+01:00) Middle European Time  
ART (GMT+02:00) Egypt Time: Cairo, Damascus, Beirut, Amman, Nicosia  
CAT (GMT+02:00) Central African Time: Johannesburg, Blantyre, Harare, Trípoli  
EET (GMT+02:00) Eastern Europe Time: Athens, Kiev, Sofia, Minsk, Bucharest, Vilnius, Tallinn  
EAT (GMT+03:00) East Africa Time: Addis Ababa, Asmara, Kampala, Nairobi, Mogadishu, Khartoum  
NET (GMT+04:00) Near East Time

PLT (GMT+05:00) Pakistan Lahore Time  
IST (GMT+05:30) Indian Time  
BST (GMT+06:00) Bangladesh Time  
VST (GMT+07:00) Vietnam Time  
CTT (GMT+08:00) Asia, Hong Kong S.A.R. China  
JST (GMT+09:00) Japan Time: Tokyo  
ACT (GMT+09:30) Australian Central Time: Darwin  
AET (GMT+10:00) Australian Eastern Time: Sydney, Melbourne, Canberra  
SST (GMT+11:00) Solomon Time  
AGT (GMT-03:00) Argentina Time  
BET (GMT-03:00) Brazil Eastern Time: Sao Paulo, Buenos Aires  
CNT (GMT-03:30) Newfoundland Time: St. Johns  
PRT (GMT-04:00) Puerto Rico and U.S. Virgin Islands Time  
EST (GMT-05:00) Eastern Time: Ottawa, New York, Toronto, Montreal, Jamaica, Porto Acre  
CST (GMT-06:00) Central Time: Chicago, Cambridge Bay, Mexico City  
MST (GMT-07:00) Mountain Time: Edmonton, Yellowknife, Chihuahua  
PST (GMT-08:00) Pacific Time: Los Angeles, Tijuana, Vancouver  
AST (GMT-09:00) Alaska Time: Anchorage, Juneau, Nome, Yakutat  
HST (GMT-10:00) Hawaii Time: Honolulu, Tahiti  
MIT (GMT-11:00) Midway Islands Time: Midway, Apia, Niue, Pago Pago

#### Exemplo 4

Um identificador de fuso horário customizado também pode ser usado por meio do formato GMT(+|-)HH:MM. Por exemplo, GMT-06:30 ou GMT+02:00.

Uma lista mais completa de identificadores de fuso horário (incluindo identificadores de formato mais longo, como "Europe/Amsterdam") podem ser localizados no arquivo "i18n\_res.xml" da pasta de configuração do produto.

### **\_start\_of\_day**

Retorna o início de hoje como um registro de data e hora.

#### Sintaxe

`_start_of_day`

#### Exemplo

`_start_of_day`

Resultado

2014-11-23 00:00:00

### **\_week\_of\_year**

Retorna o número da semana do ano de "date\_expression" de acordo com o padrão ISO 8601. Semana 1 do ano é a primeira semana do ano que contém uma quinta-feira, o que é equivalente à primeira semana que contém 4 de janeiro. Uma semana começa na segunda-feira (dia 1) e termina no domingo (dia 7).

#### Sintaxe

`_week_of_year ( date_expression )`

#### Exemplo

`_week_of_year ( 2003-01-01 )`

Resultado

1

## **\_timezone\_hour**

Retorna o valor do campo de hora de fuso horário em uma expressão de data.

### **Sintaxe**

```
_timezone_hour( date_expression )
```

### **Exemplo**

```
_timezone_hour ( 2002-01-31 12:10:10.254-05:30 )
```

Resultado

5

## **\_timezone\_minute**

Retorna o valor do campo de minuto de fuso horário em uma expressão de data.

### **Sintaxe**

```
_timezone_minute( date_expression )
```

### **Exemplo**

```
_timezone_minute ( 2002-01-31 12:10:10.254-05:30 )
```

Resultado

30

## **\_unix\_timestamp**

Retorna a hora do UNIX especificada por uma expressão de número inteiro como um registro de data e hora com fuso horário.

### **Sintaxe**

```
_unix_timestamp
```

### **Exemplo**

```
_unix_timestamp
```

Resultado

1416718800

## **\_year**

Retorna o valor do campo de ano em uma expressão de data.

### **Sintaxe**

```
_year( date_expression )
```

### **Exemplo**

```
_year ( 2003-03-01 )
```

Resultado

2003

## **`_years_between`**

Retorna números inteiros positivos ou negativos que representam o número de anos entre "date\_expression1" e "date\_expression2". Se "date\_expression1" < "date\_expression2", um valor negativo será retornado.

### **Sintaxe**

```
_years_between ( date_expression1, date_expression2 )
```

### **Exemplo**

```
_years_between ( 2003-01-30 , 2001-04-03 )
```

Resultado

1

## **`_ymdint_between`**

Retorna um número que representa a diferença entre "date\_expression1" e "date\_expression2". O valor retornado tem a forma AAAAMMDD, em que AAAA representa o número de anos, MM representa o número de meses e DD representa o número de dias.

### **Sintaxe**

```
_ymdint_between ( date_expression1 , date_expression2 )
```

### **Exemplo**

```
_ymdint_between ( 1990-04-30 , 2003-02-05 )
```

Resultado

120906, ou seja, 12 anos, 9 meses e 6 dias.

---

## **Funções comuns**

### **abs**

Retorna o valor absoluto de "numeric\_expression". Os valores negativos são retornados como valores positivos.

#### **Sintaxe**

```
abs ( numeric_expression )
```

#### **Exemplo 1**

```
abs ( 15 )
```

Resultado

15

#### **Exemplo 2**

```
abs ( -15 )
```

Resultado

15

## conversão

Converte "expressão" em um tipo de dado especificado. Alguns tipos de dados permitem que a extensão e a precisão sejam especificadas. Certifique-se de que o destino possui tipo e tamanho apropriados. Os exemplos a seguir podem ser usados para "datatype\_specification": character; varchar; char; numeric; decimal; integer; smallint; real; float; date; time; timestamp; time with time zone; timestamp with time zone e interval. Quando fizer uma conversão para um tipo interval, será necessário especificar um dos seguintes qualificadores de intervalo: year, month ou year to month para o tipo de dados do intervalo ano/mês; day, hour, minute, second, day to hour, day to minute, day to second, hour to minute, hour to second ou minute to second para o tipo de dados do intervalo dia/segundo. Observação: na conversão de um valor do tipo timestamp para date, a fração de tempo do registro de data e hora será ignorada. Na conversão de um valor do tipo timestamp para time, a fração de tempo do registro de data e hora será ignorada. Na conversão de um valor do tipo date para timestamp, os componentes de hora do registro de data e hora serão configurados como zero. Na conversão de um valor do tipo tempo para registro de data e hora, os componentes da data serão configurados de acordo com a data atual do sistema. A conversão de um tipo de dados de intervalo em outro é inválida (porque o número de dias em um mês é variável, por exemplo). Observe que é possível especificar o número de dígitos apenas para o qualificador inicial, por exemplo, YEAR(4) TO MONTH, DAY(5). Os erros serão reportados se o tamanho e o tipo do destino não forem compatíveis com o tamanho e o tipo da origem.

### Sintaxe

```
cast ( expression , datatype_specification )
```

### Exemplo 1

```
cast ( '123' , integer )
```

Resultado

123

### Exemplo 2

```
cast ( 12345 , varchar ( 10 ) )
```

Resultado

uma sequência que contém 12345

## limitando

Retorna o menor número inteiro maior que ou igual a "expressão\_numérica".

### Sintaxe

```
ceiling ( numeric_expression )
```

### Exemplo 1

```
ceiling ( 4.22 )
```

Resultado

5

## Exemplo 2

```
ceiling ( -1.23 )
```

Resultado

-1

## char\_length

Retorna o número de caracteres lógicos de "string\_expression". O número de caracteres lógicos poderá diferir do número de bytes em alguns códigos do idioma no leste asiático.

### Sintaxe

```
char_length ( string_expression )
```

### Exemplo

```
char_length ( 'Canada' )
```

Resultado

6

## unir

Retorna o primeiro argumento não nulo (ou nulo se todos os argumentos forem nulos). Necessita de dois ou mais argumentos em "lista\_de\_expressão".

### Sintaxe

```
coalesce ( expression_list )
```

### Exemplo

```
coalesce ( [Unit price], [Unit sale price] )
```

Resultado

Retorna o preço da unidade ou o preço de vendas da unidade se o preço da unidade for nulo.

## exp

Retorna a constante "e" elevada à potência de "expressão\_numérica". A constante 'e' é a base do logaritmo natural.

### Sintaxe

```
exp ( numeric_expression )
```

### Exemplo

```
exp ( 2 )
```

Resultado

7,389056

## floor

Retorna o maior número inteiro menor que ou igual a "expressão\_numérica".

### **Sintaxe**

`floor ( numeric_expression )`

### **Exemplo 1**

`floor ( 3.22 )`

Resultado

3

### **Exemplo 2**

`floor ( -1.23 )`

Resultado

-2

## **ln**

Retorna o logaritmo natural de "expressão\_numérica".

### **Sintaxe**

`ln ( numeric_expression )`

### **Exemplo**

`ln ( 4 )`

Resultado

1,38629

## **inferior**

Retorna "string\_expression" com todos os caracteres maiúsculos substituídos por minúsculos.

### **Sintaxe**

`lower (string_expression)`

### **Exemplo**

`lower ( 'ABCDEF' )`

Resultado

abcdef

## **mod**

Retorna o resto (módulos) da divisão de "integer\_expression1" por "integer\_expression2". Não é possível que "integer\_expression2" seja igual a zero ou uma condição de exceção ocorrerá.

### **Sintaxe**

`mod ( integer_expression1; integer_expression2 )`

## Exemplo

```
mod ( 20 , 3 )
```

Resultado

2

## nullif

Retorna nulo se "expressão1" for igual a "expressão2"; do contrário, retorna "expressão1".

## Sintaxe

```
nullif ( expressão1; expressão2 )
```

## posição

Retorna o valor de número inteiro que representa a posição inicial "string\_expression1" em "string\_expression2" ou 0 se "string\_expression1" não for localizada.

## Sintaxe

```
position ( string_expression1 ; string_expression2 )
```

## Exemplo 1

```
position ( 'C' , 'ABCDEF' )
```

Resultado

3

## Exemplo 2

```
position ( 'H' , 'ABCDEF' )
```

Resultado

0

## position\_regex

Retorna o valor de número inteiro que representa a posição inicial ou final da subsequência em "string\_expression" que corresponde à expressão regular "regex\_expression". A procura inicia na posição "integer\_expression1", que tem um valor padrão de 1. A ocorrência do padrão para procura é especificada por "integer\_expression2", que tem um valor padrão de 1. A opção de retorno, especificada pelo primeiro argumento, especifica o que é retornado em relação à ocorrência. Se você especificar "start", a posição do primeiro caractere da ocorrência é retornada. Se você especificar "after", a posição do caractere seguinte à ocorrência é retornada. Se você não especificar uma opção de retorno, "start" está implícito. As sinalizações para configurar opções para a interpretação da expressão regular são especificadas por "flags\_expression". Letras individuais são usadas para definir as sinalizações, com valores válidos sendo 's', 'm', 'i' e 'x'.

## Sintaxe

```
position_regex ([ start|after ] regex_expression , string_expression  
[ , integer_expression1 [ , integer_expression2 [ , flags_expression ] ] ] )
```

### Exemplo 1

```
position_regex ( '.er' , 'Flicker Lantern' )
```

Resultado

5

### Exemplo 2

```
position_regex ( after '.er' , 'Flicker Lantern' )
```

Resultado

8

### Exemplo 3

```
position_regex ( '.er' , 'Flicker Lantern' , 1 , 2 )
```

Resultado

12

## potência

Retorna "numeric\_expression1" elevada à potência "numeric\_expression2". Se "numeric\_expression1" for negativa, "numeric\_expression2" deve resultar em um valor de número inteiro.

### Sintaxe

```
power ( numeric_expression1 ; numeric_expression2 )
```

### Exemplo

```
power ( 3 , 2 )
```

Resultado

9

## \_round

Retorna "numeric\_expression" arredondada para casas decimais "integer\_expression". Observação: "integer\_expression" deve ser um número inteiro não negativo. Arredondamentos são feitos antes que a formatação dos dados seja aplicada.

### Sintaxe

```
_round ( numeric_expression ; integer_expression )
```

### Exemplo

```
_round ( 1220.42369 , 2 )
```

Resultado

1220.42

## raiz quadrada

Retorna a raiz quadrada de "numeric\_expression". É necessário que "numeric\_expression" seja um número positivo.

### Sintaxe

```
sqrt ( numeric_expression )
```

### Exemplo

```
sqrt ( 9 )
```

Resultado

3

## substring

Retorna uma subsequência de "string\_expression", que começa na posição de "integer\_expression1", para os caracteres de "integer\_expression2", ou no final de "string\_expression", se a "integer\_expression2" for omitida. O primeiro caractere de "string\_expression" ocupa a posição 1.

### Sintaxe

```
substring ( string_expression , integer_expression1 [ ,  
integer_expression2 ] )
```

### Exemplo

```
substring ( 'abcdefg' , 3 , 2 )
```

Resultado

cd

## substring\_regex

Retorna uma subsequência de "string\_expression" que corresponde à expressão regular "regex\_expression". A procura inicia na posição "integer\_expression1", que tem um valor padrão de 1. A ocorrência do padrão para procura é especificada por "integer\_expression2", que tem um valor padrão de 1. As sinalizações para configurar opções para a interpretação da expressão regular são especificadas por "flags\_expression". Letras individuais são usadas para definir as sinalizações, com valores válidos sendo 's', 'm', 'i' e 'x'.

### Sintaxe

```
substring_regex ( regex_expression , string_expression [ , integer_expression1  
[ , integer_expression2 [ , flags_expression ] ] ] )
```

### Exemplo 1

```
substring_regex ( '.er' , 'Flicker Lantern')
```

Resultado

ker

## Exemplo 2

```
substring_regex ( '.er' , 'Flicker Lantern' , 1 , 2 )
```

Resultado

```
ter
```

## trim

Retorna uma "string\_expression" cujas lacunas iniciais e/ou finais foram removidas ou que não apresenta algum caractere especificado em "match\_character\_expression". "Both" está implícito quando o primeiro argumento não estiver declarado e a lacuna estiver implícita quando o segundo argumento não estiver declarado.

### Sintaxe

```
trim ( [ [ trailing|leading|both ] [ match_character_expression ] , ]  
string_expression )
```

## Exemplo 1

```
trim ( trailing 'A' , 'ABCDEF A' )
```

Resultado

```
ABCDEF
```

## Exemplo 2

```
trim ( both , ' ABCDEF ' )
```

Resultado

```
ABCDEF
```

## maiúscula

Retorna "expressão\_string" com todos os caracteres minúsculos substituídos por maiúsculos.

### Sintaxe

```
upper (string_expression)
```

## Exemplo

```
upper ( 'abcdef' )
```

Resultado

```
ABCDEF
```

## Funções trigonométricas

### arccos

Retorna o arco cosseno do argumento, em que o argumento está na faixa de -1 a 1 e o resultado é um valor expresso em radianos.

### Sintaxe

```
arccos ( numeric_expression )
```

**Exemplo**

`arccos ( -1 )`

Resultado

3.1415

**arcsin**

Retorna o arco seno do argumento, em que o argumento está na faixa de -1 a 1 e o resultado é um valor expresso em radianos.

**Sintaxe**

`arcsin ( numeric_expression )`

**Exemplo**

`arcsin ( 0 )`

Resultado

3.1415

**arctan**

Retorna arco tangente do argumento, em que o argumento está na faixa de -1 a 1 e o resultado é um valor expresso em radianos.

**Sintaxe**

`arctan ( numeric_expression )`

**Exemplo**

`arctan ( 0 )`

Resultado

3.1415

**cos**

Retorna o cosseno do argumento, em que o argumento é expresso em radianos.

**Sintaxe**

`cos ( numeric_expression )`

**Exemplo**

`cos ( 0.3333 * 3.1415 )`

Resultado

0.5

**coshyp**

Retorna o cosseno hiperbólico do argumento, em que o argumento é expresso em radianos.

**Sintaxe**

`coshyp ( numeric_expression )`

**Exemplo**

`coshyp ( 0 )`

Resultado

1

**sin**

Retorna o seno do argumento, em que o argumento é expresso em radianos.

**Sintaxe**

`sin ( numeric_expression )`

**Exemplo**

`sin ( 0.1667 * 3.1415 )`

Resultado

0.5

**sinhyp**

Retorna o seno hiperbólico do argumento, em que o argumento é expresso em radianos.

**Sintaxe**

`sinhyp ( numeric_expression )`

**Exemplo**

`sinhyp ( 0 )`

Resultado

0

**castanho**

Retorna a tangente do argumento, em que o argumento é expresso em radianos.

**Sintaxe**

`tan ( numeric_expression )`

**Exemplo**

`tan ( 0.25 * 3.1415 )`

Resultado

1

**tanhyp**

Retorna a tangente hiperbólica do argumento, em que o argumento é expresso em radianos.

**Sintaxe**

`tanhyp ( numeric_expression )`

**Exemplo**

`tanhyp ( 0 )`

Resultado

0

---

## Apêndice B. Sobre este Guia

Este documento foi criado para usar com o IBM Cognos Analytics. O Cognos Analytics integra relatório, modelagem, análise, painéis, métricas e gerenciamento de eventos, para que seja possível entender os dados de sua organização e tomar decisões de negócios efetivas.

Para localizar a documentação do produto na web, incluindo toda a documentação traduzida, acesse IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter>).

### Recursos de Acessibilidade

Os recursos de acessibilidade ajudam os usuários com alguma deficiência física, como restrição motora ou deficiências visuais, a utilizarem produtos de tecnologia da informação com êxito. Para obter informações sobre os recursos de acessibilidade no Cognos Analytics, consulte o *Guia de acessibilidade do Cognos Analytics*.

### Instruções prospectivas

Esta documentação descreve a funcionalidade atual do produto. Referências a itens que não estão disponíveis atualmente podem estar incluídas. Nenhuma implicação sobre qualquer disponibilidade futura deve ser inferida. Essas referências não representam qualquer compromisso, promessa ou obrigação legal para a entrega de nenhum material, código ou funcionalidade. O desenvolvimento, a liberação e a sincronização de recursos ou funcionalidade permanecem conforme critérios exclusivo da IBM.

### Termo de responsabilidade das amostras

A Companhia das Aventuras de Amostra, Companhia das Grandes Aventuras, Vendas GA, qualquer variação dos nomes Aventuras de Amostra ou Grandes Aventuras e Amostra de Planejamento representam operações de negócios fictícias com dados de amostra usados para desenvolver aplicativos de amostra para a IBM e para os clientes da IBM. Esses registros fictícios incluem dados de amostra para transações de vendas, distribuições de produtos, finanças e recursos humanos. Qualquer semelhança com nomes, endereços, números de contato ou valores de transações reais é coincidência. Outros arquivos de amostra podem conter dados fictícios gerados automática ou manualmente, dados factuais compilados de origens acadêmicas ou públicas ou dados usados com permissão do portador de copyright, para uso como dados de amostra para desenvolver aplicativos de amostra. Os nomes de produtos referidos podem ser marcas registradas de seus respectivos proprietários. É proibida a duplicação desautorizada.



---

# Índice Remissivo

## C

caminho de navegação  
  como criar 19  
  eliminando 19

## E

editando módulos  
  erros de validação 22  
Editando módulos de dados 11  
  ações desfazer e refazer 11  
  Interface do usuário 11  
editor de expressão  
  Funções comuns 50  
  Funções de data/hora comerciais 38  
  Funções estatísticas 32  
  Funções trigonométricas 57  
  Resumos 32

## F

filtros  
  incluindo 20  
  remover 20

## I

interface com o usuário de modelagem 11

## L

limpeza  
  colunas em módulos 16

## M

modelagem de dados 1  
módulos  
  editando 16  
  limpando dados 16

módulos (*continuação*)  
  ocultando tabelas e colunas 21  
  validando 22  
módulos de dados  
  editando 11  
  propriedades de tabela e coluna 23  
  vinculando novamente origens 8

## O

ocultando  
  tabelas e colunas 21  
origens  
  vinculando novamente em um módulo de dados 8

## P

propriedade agregada 23  
propriedade de uso 23  
propriedade do identificador 23  
propriedades  
  tabelas e colunas 23

## R

redo  
  Editando módulos de dados 11

## U

undo  
  Editando módulos de dados 11

## V

validando  
  módulos 22  
vinculando novamente  
  origens em um módulo de dados 8