

## QLIK SENSE ENTERPRISE

### 1. TRASNSFORMANDO DADOS PARA CRIAR UM B.I:

Ao acessar a o QLIK SENSE : <https://qliksense.g10.com.br/hub/my/work>, ao lado direito terá a opção '**Criar novo aplicativo**', de um nome para seu B.I e click em 'criar'.

2. Ao criar o aplicativo, será pedido para ser inserido dados. Você pode inserir via script (**Os melhores profissionais que atuam com o Qlik Sense, trabalham apenas com script de dados**): na opção '**Editor de carregamento de dados** ' ou pode ser inserido via planilhas, arquivos ou outras fontes, na opção: 'Adicionar dados de arquivos e outras fontes'.**(NÃO RECOMENDÁVEL)**
3. Para carregar os dados via script de dados, você terá que criar uma conexão(banco de dados, link web, pastas com arquivos e planilhas, etc..), basta saber qual você ira utilizar.
4. Ao lado esquerdo estarão as abas que serão utilizadas para o carregamento e tratamento do script utilizado para fazer o B.I , na aba **Main** estará todas as configurações que estão sendo usadas dentro do qlik, no nosso caso as configurações são da nossa língua PT-BR. Para adicionar uma nova aba para tratar o script que será usado no B.I, basta apenas clicar no sinal de mais ( + ) e escolher um nome, por padrão utilizo o nome '**DADOS**' .
5. Ao adicionar a aba de dados, agora você terá que criar uma conexão para trazer os dados e tratar ele via script. Para isso basta somente clicar em "**CRIAR NOVA CONEXÃO** " e escolher alguma das opções de conexões.
6. Após a conexão ser feita basta importar e tratar os dados. Aqui está um exemplo de como os dados de uma tabela foi tratada dentro do editor da carga de dados:

```
1 NullAsValue*;  
2 Set NullValue = '';  
3  
4  
5 TEMP:  
6 LOAD  
7   RowNo() AS ID,  
8   "Responsável" AS RESPONSAVEL,  
9   "Data de Início Ocorrência" AS DATA_INICIO_OCORRENCIA_TEMP,  
10  "Nº Romaneio" AS ROMANEIO,  
11  "Cliente Origem" AS CLIENTE,  
12  "Data Chegada Destino" AS DATA_ENTRADA,  
13  YEAR(YEARSTART("Data Chegada Destino")) AS DATA_ANO,  
14  MONTH(MONTHSTART("Data Chegada Destino")) AS DATA_MES,  
15  "Data Fim Descarga Destino" AS DATA_FINALIZADA,  
16  DATE("Data Chegada Destino") AS DATA_CHEGADA_TEMP,  
17  Time(FRAC("Data Chegada Destino")) AS HORA_CHEGADA_TEMP,  
18  DATE("Data Fim Descarga Destino") AS DATA_DESCARGA_TEMP,  
19  Time(FRAC("Data Fim Descarga Destino")) AS HORA_DESCARGA_TEMP,  
20  Coalesce("Quantidade Horas", (Now("Data Fim Descarga Destino") - "Data Chegada Destino") * 24) AS quantidade_horas,  
21  Interval(Today() - If(IsNull("Data Baixa Operador"), Today(), "Data Baixa Operador"), 'DD') AS DIA_PARADO,  
22  Replace(Replace(Replace(("Descrição Motivo"), 'IND', ''), 'IND', ''), 'TP', ',') AS MOTIVOS,  
23  "Modalidade" AS MODALIDADE,  
24  "Placa Cavallo" AS PLACA,  
25  "VALOR TOTAL (R$)" AS VALOR_TOTAL,  
26  "Data de Início Ocorrência" AS DATA_ABERTURA,  
27  "Nº OC" AS NRO_OC,  
28  Agenda AS AGENDA,  
29  "Quantidade de diarias Estadia" AS QUANTIDADE_DIARIA,  
30  "Outros Custos (R$)" AS VALOR_EXTRA,  
31  Traker AS TRK,  
32  "Status Ocorrência" AS STATUS_OCORRENCIA,  
33  "Data Encerramento Ocorrência" AS DATA_FIM_OCORRENCIA_TEMP,  
34  "Status Final" AS STATUS,  
35  "Local/Operador" AS OPERADOR,  
36  IF(IsNull("Grid"), 'TERCEIRO', "Grid") AS GRID,  
37  "Ação1" AS ACAO,  
38  "Data Baixa Operador" AS BAIXA_OPERADOR,  
39  "Meio de Abertura" AS MEIO_DE_ABERTURA,  
40  "Cidade Origem" AS CIDADE,  
41  "Cidade Destino" AS DESTINO,  
42  "Descrição do Produto" AS PRODUTO  
43  FROM [lib://UNIFICADAS (g10 tp.dev.3)]  
44  (ooxml, embedded labels, table is [Ocorrencias - Auto])  
45  where "Data de Início Ocorrência" >= Date#('01/06/2023', DD/MM/YYYY);  
46  
47
```

Feito isso clique em **carregar dados**. Caso alguma tratativa do script esteja errada, o próprio Qlik informa o erro. Após carregados os dados, uma pasta vazia será criada, nesta pasta será criado os gráficos, KPI, tabelas, etc..

**OBS: todo B.I que precisar ser modificado seu script estará dentro do Editor da carga de dados.**

Onde encontrar o KPI para utilizar:

**Objetos personalizados → Climber Self-Service Extensions → Climber KPI**

KPI→ Usados para índice de medidas, quantidades.

## A importância dos Itens Mestres

- **Itens Mestres de Dimensões** : Geralmente usados com o tipo de dimensão: **Hierarquia**. Itens mestres do tipo hierarquia não podem ser alterados ou modificados quando estão dentro de um KPI, caso queira mudar ou modificar pelo KPI. Para modificar ou mudar: **Itens Mestres → Dimensões**. Para criar um, utiliza do mesmo passo.
- **Itens Mestres de Medidas**: Usado para criar as Set Analysis que iram calcular e trazer as medidas (tempo médio, tempo, contagem,etc). Quando criado basta arrastar e jogar dentro do KPI, gráficos. Itens Mestres do tipo de medidas, podem ser alterados e modificados dentro do KPI.
- **Itens Mestres de Visualizações**: São KPI que se tornam Itens Mestres, e uma vez que se tornam Itens mestres de visualizações, podem ser usados dentro da extensão **GROUPED CONTAINER → Extensão usada para incluir mais de uma visualização e agrupar em um mesmo espaço**.
- **OBS**: Quando criados todos os itens mestre que serão utilizados, você não precisara mudar KPI por KPI quando houver alguma alteração, basta alterar diretamente dentro do item mestre a Set Analysis utilizada, que automaticamente todos os KPI serão mudados. Por isso os itens mestres têm sua importância dentro do Qlik, poupando tempo.
- Esses pontos são os pontos básicos para a criação de um B.I

## TIPOS DE OPERADORES

- **Operadores numéricos:** Soma (+); Menos(-) ; Divisão (/); Multiplicação(\*)

- **Operadores Relacionais:** São operadores utilizados para retornar o resultado de uma expressão como verdadeiro ou falso.

- **Comparador igual (=):**

Quando a expressão retornar '-1' a expressão é verdadeira.

Quando a expressão retornar '0' a expressão é falsa.

Exemplo:  $1 = 1$  retorna -1 |  $1 = 3$  retorna 0.

Pode ser usado a função '**True ()**' ou '**False ()**'.

Para converter um texto em um número deve se usar "" para a conversão.

Exemplo:  $1 = '1' = -1$ , automaticamente o qlik converte o texto que pode ser convertido para número retornando -1(true).

Agora se tiver uma letra junto ao número, o qlik retorna 0 (false), pois esse texto não pode ser convertido para número.

Exemplo:  $1 = '1a' = 0$ .

- **Comparador menor (<):**

$1 < 2 = -1$  (true).

$2 < 2 = 0$  (false).

- **Comparador menor ou igual (<=):**

$2 <= 2 = -1$  (true).

$2 <= 1 = 0$  (false).

- **Comparador maior (>):**

$3 > 2 = -1(\text{true})$ .

$2 > 5 = 0(\text{false})$ .

- **Comparador maior ou igual (>=)**

$3 >= 2 = -1 (\text{true})$ .

$2 >= 3 = 0 (\text{false})$ .

- **Comparador diferente (< >):**

$2 < > 2 = 0 (\text{false})$ .

$5 < > 10 = -1 (\text{true})$ .

- **Comparador de texto:**

1. **Precedes:** função usada para verificar a ordem alfabética (ordem crescente).

**Exemplo:** 'Amanda' precedes 'Bia' = -1(true).

2. **Follows:** função usada para verificar a ordem alfabética (ordem decrescente).

**Exemplo:** 'Zulu' follows 'Bia' = -1 (true).

**OBS:** Lembre-se para o QLIK reconhecer que se trata de um texto, a palavra tem que estar dentro das aspas: 'Bia'.

- **Operadores lógicos:** Pode ser usado mais de uma vez dentro da operação.

**AND:** Operador lógico usado para retornar True ou False, dependendo da sua operação. Pode ser usado mais de uma vez dentro da operação.

**Exemplo:**  $1=1 \text{ and } 2=2$ , vai retornar true (- 1).

$1=1 \text{ and } 2=2 \text{ and } 3=5$ , vai retornar false (0).

**OR:** Operador logico 'ou', se qualquer operação retornar True(-1), o resultado final retorna True (-1), só retorna False(0) caso todas as operações forem falsas.

**Exemplo:** 1=2 OR 5=3 OR 3=3, vai retornar True (-1).

**XOR:** Conhecido como exclusivo 'ou', somente uma das condições 'ou' pode retornar verdadeiro, caso mais alguma condição retorne verdadeiro, o sinal final irá retornar falso (0).

**NOT :** Operador logístico que inverte o resultado final.

Exemplo : 1 = 1, retorna -1 (True) se utilizar o operador not : not 1=1, vai retornar 0 (false).

### **Operadores de Caracteres :**

- **Ampersand ( & ) :** Junta os textos.  
**Exemplo:** 'Bia' & 'Ana' = Bia Ana
- **Like : Comparação de texto para retornar false ou true.**  
**Exemplo :** 'Amanda' like 'Amanda' retorna -1 (true).

#### **Dentro do like podemos utilizar caracteres coringas, exemplos:**

'Amanda' like \*a\*, retorna -1(true), já que o \*a\* quer dizer que contenha qualquer coisa antes da letra 'a' e depois da letra 'a'.

Podemos usar especificações junto com o LIKE, por exemplo:

'Sergio' like \*g?o\*, vai retornar -1 (true). → Começa com qualquer letra e termina com qualquer letra, mas que entre letra G e a O tenha apenas um caractere.

'\*' → Significa qualquer coisa e qualquer caractere.

'?' → significa qualquer palavra ou caractere, porém com quantidade 1.

Essa parte de operadores lógicos e caracteres parece ser uma parte boba, mas é muito usada dentro do QLIK para construir as funções, dentro do editor de Carga de Dados e nas Set Analysis.

## FUNÇÕES DE AGREGAÇÕES

- **SUM** → Função de soma, vai somar todos os valores.

Pode incluir operadores junto a função, exemplo:

Sum(valor\_total \*2), Sum(valor\_total / 2), etc...

- **MAX** → TRÁS O VALOR MAXIMO DA COLUNA  
EXEMPLO: MAX(VALOR).
- **MIN** → TRÁS O VALOR MINIMO DA COLUNA.  
EXEMPLO: MIN(VALOR).
- **COUNT** → Conta todas as linhas que possuem dados(valores), onde se tem campos nulos ele não irá contar.  
EXEMPLO: COUNT(VALOR).
- **ONLY** → Trás o resultado quando existe apenas um valor dessa dimensão, se tiver mais de um valor vai retornar como nulo.  
EXEMPLO: ONLY(ANO).  
  
Se existir mais de um ano dentro da tabela, retorna nulo (0), se você filtrar apenas por 1 ano, exemplo: 2018, vai retornar 2,018 como valor.
- **FirstSortedValue** → Trás a classificação do ranking de acordo que você colocar, desde que exista apenas 1.

EXEMPLO: FirstSortedValue (cliente, valor,1) → Traga o cliente que tenha o menor valor.

Para trazer o cliente que tem o maior valor basta pôr o sinal de menos antes do valor → FirstSortedValue (cliente, -valor,1).

Quando dois ou mais clientes ocupam a mesma posição, o resultado vai retornar nulo.

- **Mode** → Trás o valor que mais se repetiu dentro do banco de dados ou do arquivo que está trazendo as informações.  
EXEMPLO → mode(cliente)

- **Distinct** → prefixo que faz contagem excluindo as colunas que se repetem.  
EXEMPLO → Count (distinct cliente).

- **Total** → Faz a contagem total de colunas ignorando as dimensões, usado para fazer tabelas com porcentual ou gráficos.

EXEMPLO → Count (total valor).

Caso você deseje fazer uma contagem total, mas não queira ignorar uma dimensão dessa contagem, utilize: < >

Exemplo → Count (total <cliente> valor).

- **ALL** → Ignora dimensões e filtros, ele é um totalizador total, usado em gráficos de pareto.
- **NullCount** → Conta valores nulos (usado para fazer teste de consistência do seu banco de dados).
- **TextCount** → Quando tiver um texto e um número dentro de uma coluna, essa função conta somente os valores de texto. Exemplo seria uma coluna com o nome dos clientes, essa função vai contar quantos clientes existem.
- **NumericCount** → Essa função conta somente o que ela interpreta como número dentro da coluna.
- **AGV** → Função usada para se obter a média: número, tempo...
- **Median** → Função usada para trazer a mediana dentre os valores que se tem dentro do banco de dados.
- **MaxString** → Função de texto que traz em ordem alfabética, trás sempre o último nome.
- **MinString** → Função de texto que traz em ordem alfabética, traz sempre o primeiro nome.
- **Concat** → Função que faz agrupamento de texto, por exemplo trazer em quantas vezes um cliente parcelou um valor.



- **Funções de dimensões sintéticas:** funções sintéticas, são funções que precisam ser criadas, pois ainda não existem dentro do seu banco de dados. Exemplo:
- **ValueList** → Aplica uma subdimensão dentro dos valores sintéticos. Pode retornar textos.
- **ValueLoop** → Aplica um início, um meio e um fim, basicamente como o valuelist.

### **Tabela Virtual e Funções:**

- **AGGR:** Essa função cria uma tabela virtual antes de jogar os valores nas funções de agregações.
- **IF:** Função usada para comparar valores, para determinar cores.

Exemplo: IF(AVG(Valor) < 499, 'Media OK', 'Media NG') → Medida.

IF(AVG(Valor) < 499, '#00F00', '#FF0000') → Para cores.

Essas condicionais feitas com a função IF, podem serem usadas em KPI, gráficos, tabelas...

### **Funções de cores :**

- **RGB** : Que são as combinações das cores: vermelho, verde e azul que vai de 0 a 255.

Exemplo: RGB(255,200,170).

- **ARGB** : Uma função alfa, que seria a transparência da cor, também vai de 0 a 255.

Exemplo:(255,0,0).

- **ColorMix** : Essa função faz um gradiente da cor 0 até a cor especificada, e na formula você precisa por um resultado que retorne de 0 a 1(que seria em percentual). Dentro da função utilize a função **RAND ()**, pois essa função já retorna o valor entre 0 e 1.

Exemplo: ColorMix1(Rand(),RED(),Blue())

Podemos fazer as cores navegarem entre um intervalo mínimo e máximo dos valores dentro de uma tabela usando como medida. (Como se fosse um mapa de calor ).

Exemplo:

```
ColorMix1(
    Sum([VIAGEM.ALERTA_GRID]
    -
    Min(total
        Aggr(
            sum([VIAGEM.ALERTA_GRID]),
            [VIAGEM.ALERTA_DATA],
            [VIAGEM.ALERTA_OPERACAO]
        )
    )
)
/
(
    max(total
        Aggr(
            sum([VIAGEM.ALERTA_GRID]),
            [VIAGEM.ALERTA_DATA],
            [VIAGEM.ALERTA_OPERACAO]
        )
    )
    -
    Min(total
        Aggr(
            sum([VIAGEM.ALERTA_GRID]),
            [VIAGEM.ALERTA_DATA],
            [VIAGEM.ALERTA_OPERACAO]
        )
    )
)
```

)

)

Rand(),RED(),Blue())

- **ALT** → Função de logica que retorna o primeiro resultado numérico valido. Pode-se usar várias expressões dentro da função alt.

**Exemplo:** ALT(Sum(Valor),Count(valor),'Nenhuma expressão numérica valida').

Nesse exemplo se a expressão Sum(Valor) e a expressão Count(Valor) retornar algum valor valido, vai retornar o resultado numérico, caso contrário vai retornar '**Nenhuma expressão numérica válida**'.

**OBS:** O valor numérico que retorna não é a soma das duas expressões e sim o valor da expressão que tenha o primeiro resultado numérico. Se a expressão Sum(valor) não retornar um resultado numérico e a expressão Count (Valor) retornar, vai ser o resultado da expressão Count(Valor) que ira aparecer, vice-versa.

- **CLASS** → Função usada para determinar qual intervalo certo o valor está.

Exemplo→ Class(15,10) → O resultado é 15 e o intervalo vai ser de 10 em 10. Essa função com esse resultado vai retornar:  
 $10 \leq x < 20$ .

### **Funções de comparações de textos:**

- **MATCH** → Retorna à posição do texto procurado

Exemplo → Match(Jan, Fev, Mar, Jan) → Nesse exemplo estamos procurando a posição que a palavra '**Jan**' está. → Retorna o número 3. Diferencia maiúscula de minúscula.

Pode ser usada para classificação de texto, impondo a ordem eu quiser.

Exemplo → Match(cliente, 'cargil', 'heineken')\*-1 → Multiplicar por -1(vem antes do 0) para trazer sempre quem você quer que apareça por primeiro, nesse exemplo, cargil vai aparecer por primeiro na tabela.

Se tiver muitas classificações, fica inviável usar o '**Match**' para classificar.

- **MIXMATCH** → Função que retorna à posição do texto procurado, porém essa função não diferencia maiúsculas de minúsculas.

Exemplo → MixMatch(Jan, Fev, Mar, jan) → Retorna o número 3.

- **WILDMATCH** → Função que retorna a posição do texto, mas pode ser usado caractere coringas junto a função.

Exemplo → WildMatch( 'Jun','Fev','Mar', 'Jan', 'Ju\*') → Retorna número 4.

WildMatch('Mari', 'Fev', 'Ma??','Jan','Ju') → Retorna número 2.

Lembrando que o ? significa um caractere, logo ?? seriam dois caracteres após a letra 'a' da palavra Ma.

E o \* **significa** qualquer coisa. No exemplo, seria qualquer coisa após a letra u.

- **PICK** → Função para pegar um determinado argumento que você digita após um número de argumento.

**Exemplo:** PICK(2,Sum(valor),Count(valor),30,50)

O primeiro argumento é o 2, ele retorna o Count(valor), se fosse o 1, retornaria o Sum(valor), se fosse o 3, retornaria 30 , e se fosse 4, retornaria 50.

**ROWNO** → Função usada para contar o número de linhas de uma tabela, banco de dados.

Serve para aplicar formatações dentro das linhas, usadas em parênteses, resultados acumulados.

## VALORES DE TEMPO – DATA E HORA

- Expressões com inteiro de hora .

**Now** → Traz a data e a hora.

**Second** → traz os segundos.

**Minute** → traz os minutos.

**Hour** → traz as horas.

**Day** → Traz o dia.

**Week** → Traz a semana do ano.

**Month** → Traz o mês.

**Year** → Traz o ano.

**WeekYear** → Retorna o ano ao qual a semana pertence.

**WeekDay** → Retorna o dia da semana.

**Today** → Traz a data.

**LocalTime** → Traz informações da hora local.

**MakeDate** → Criação do tipo date → MakeDate(2023,12,14).

**MakeWeekDate** → Calcula uma certa data da semana que você precisa fazer de acordo com a semana do ano →

**MakeWeekDate(2023,30,5)** → Nesse exemplo queremos saber qual a data da semana 30 do dia 5 de 2023.

- **Maketime** → Cria o campo de hora, minuto e segundo → **MakeTime(10,40,22)** → 10 horas, 40 min e 22seg.

Para transforma a data em número, basta multiplicar por 1, que o QLIK transforma em numero a data.

Exemplo → **MakeDate(2023,12,14)\*1** →retornar o número 43799.

### Adicionando meses e anos

- **AddMonths** → Função usada par quando você quiser adicionar ou remover meses de uma determinada data.

Exemplo: Para adicionar data → **AddMonths(data,2)** → vai adicionar dois meses a data atual. Para remover a data → **AddMonths(data,-2)** → vai remover dois meses da data atual.

- **AddYears** → Função usada para adicionar ou remover anos de uma determinada data.

Exemplo: **AddYears(data,1)** → Vai adicionar um ano a data atual.  
**AddYears(-1)** → Vai remover 1 ano da data atual.

- **Yeartodate** → Retorna um valor booleano, para ver se a data que está sendo analisado está dentro da data do ano atual. Como padrão essa expressão vai pegar a data que foi carregada o arquivo.
- **YearStart(data)** → Essa função sempre vai retornar o início do ano.
- **YearEnd(data)** → retornas sempre o último dia do ano.
- **YearName(datat)** → retorna sempre o nome do ano.
- **QuarterStart(data)** → Retornar o início do trimestre.

**DayStart(data)** → Inicia as horas, transformando o campo em um timestamp.

**DayName(data)** → Trunca as horas, ou seja, se tiver um campo com data e horas juntos, vai retornar somente a data.

**MonthStart(data)** → Início do mês.

**MonthEnd(data)** → Final do mês.

**MonthName(data)** → Nome do mês.

**Age(Today(),Data)** → Age→ Função para calcular a idade de uma pessoa a partir da data.

**NetWorksDays(data,Today())** → Calcula a quantidade de dias uteis.

## FUNÇÕES DE CAMPO E NÚMERICAS

- **POW** → Função de potência  
EXEMPLO: pow(2,3) → 8.
- **SQRT** → Função de raiz quadrada  
EXEMPLO: sqrt(9) → 3.
- **GetAlternativeCount** → Função que conta as possibilidades de seleção alternativas a seleções que você fez.
- **GetExcludedCount** → Função que traz as alternativas que dentro da seleção estão excluídas.
- **GetPossibleCount** → Função que retorna as possíveis seleções que podem ser feitas.

- **GetSelectedCount** → Função que retorna à quantidade de seleções feitas.
- **GetCurrentSelections** → Função que retorna os campos que estão selecionados no momento.
- **GetFieldSelections** → Função que retorna o que foi selecionado no filtro.
- **GetObjectField** → Função que retorna a dimensão que está sendo utilizada. Essa função é utilizada para descobrir o nome da dimensão que está sendo usada sem ter que procurar no rotulo.
- **Date(43790,' DD/MM/YYYY ')** → Transformando o número para a data, lembrando que a expressão da data tem que ser em aspas simples, já que se trata de um texto.
- **Dual** → combina o valor de texto com o valor numérico.

Exemplo → Dual('Mês ' & month(data), month(data)\*1) → Trás o mês em texto, para trazem em número basta adicionar +0 no final da função.

- **Interval** → Trás um determinado número em um intervalo de tempo.

Exemplo → Interval(100.5,'D hh:mm') → Lembando que o ponto no Qlik é o separador decimal, nessa função pedimos para que o Qlik traga em dia, hora e minutos o número 100.5.

- **Money** → Retornar o valor numérico em moeda.

Exemplo → money(1000, 'R\$ ###0,00',',','.').

- **Num** → Retorna o valor numérico.

**Time** → Traz a formatação de um número em tempo.

Exemplo → time(0.845698, 'hh:mm:ss') → Vai retornar um tempo em horas.

**Timestamp** → Função que formata um número em data e hora.

Exemplo → Timestamp(50.6065,'DD:HH:MM:SS').



## Funções de Interpretações

São funções que pega a base de dados ao qual só vem como texto. Essas funções fazem com que o QLIK pegue o texto e interprete como um número:

Exemplo → Alguma data venha irregular 10-12/2023 →  
Data(**date#**('10-12/2023','MM/-DD YYYY'),'DD/MM/YYYY').

**Text** → Função que força o QLIK interpretar um número em texto.

Exemplo → Text('12/12/2023').

## Funções numéricas gerais

**DIV** → Função que retorna o valor inteiro de uma expressão de uma divisão, ela pede dois argumentos de números inteiros.

**Exemplo** → div(7/2) → retorna 3 e não 3,5 se retirar a função DIV retorna 3,5.

**FABS** → Função que retorna um número absoluto positivo independente do sinal. Seria como um número dentro de um módulo.

**Exemplo** → Fabs(-4) → retorna 4 .

**FACT** → Função que representa o fatorial.

**Exemplo** → Fact(5) → 120.

**FRAC** → Função que retorna a função fracionaria de x.

**Exemplo** → Frac(7/2) → retorna 0,5.







