

Por Cleuton Sampaio

Se você entrar em uma empresa, seja qual for, pública ou privada, encontrará algumas centenas de planilhas Microsoft Excel e, muitas delas, contendo sistemas inteiros, feitos como macros. Muitas vezes, as pessoas utilizam versão “pirata” do Excel, só para continuar a utilizar as macros.

Bem, a grande novidade é que o LibreOffice suporta macros! E em todos os produtos!

Nesta série, vamos ver como criar macros na planilha do LibreOffice, o que lhe permitirá substituir seu software de planilha, migrando totalmente para a suíte Open Source.

## ***O que é uma macro?***

Boa pergunta! Vou tentar responder de maneira mais simples possível: É um conjunto de comandos que você quer executar. Em vez de digitar sempre os mesmos comandos, você pode agrupá-los e criar uma “macro”, que pode executar sempre que necessitar.

Porém, uma macro é mais do que isso! É um verdadeiro programa de computador, escrito em uma linguagem de programação de alto nível, que permite a você automatizar as mais diversas tarefas em uma planilha.

# COMO FAZER | *tutorial*

## ***Então eu preciso saber programar?***

Infelizmente, sim. Para criar macros é necessário ter experiência em lógica de programação. Embora o LibreOffice ofereça várias linguagens, entre elas o Basic, que é muito fácil, mesmo assim é necessário conhecer um pouco de programação.

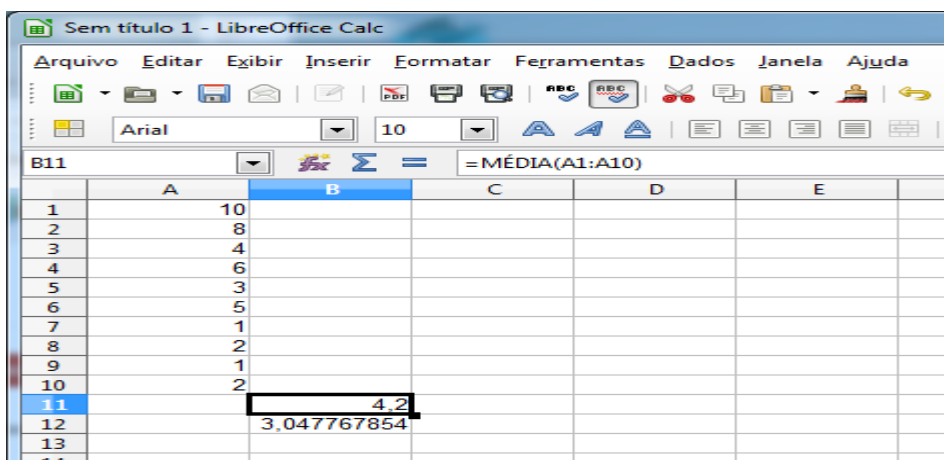
Porém, muitos usuários de Excel aprenderam de forma simples, utilizando livros, e seguiram adiante, criando macros que são verdadeiros sistemas aplicativos.

Eu mesmo escrevi um livro: “Criando Macros no BrOffice Calc”, editado pela Brasport. Apesar do livro ter sido escrito utilizando o

BrOffice, seus conceitos funcionam perfeitamente no LibreOffice. O LibreOffice oferece várias linguagens de programação para criar macros: LibreOffice Basic, BeanShell, JavaScript e Python. Eu optei pelo LibreOffice Basic, por que é muito parecido com o VBA (Visual Basic for Applications), utilizado no Microsoft Excel.

## ***Vamos começar?***

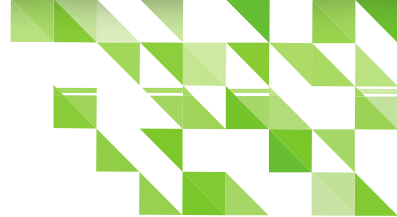
Neste primeiro artigo, vamos criar uma macro bem simples, que calcula a média e o desvio padrão de um intervalo de células. Para começar, crie uma planilha como a da figura 1.



The screenshot shows the LibreOffice Calc interface. The title bar reads "Sem título 1 - LibreOffice Calc". The menu bar includes "Arquivo", "Editar", "Exibir", "Inserir", "Formatar", "Ferramentas", "Dados", "Janela", and "Ajuda". The toolbar contains various icons for file operations and editing. The formula bar shows "=MÉDIA(A1:A10)". The spreadsheet has columns A through E and rows 1 through 14. Column A contains the values 10, 8, 4, 6, 3, 5, 1, 2, 1, 2. Cell B11 contains the value 4,2. Cell B12 contains the formula result 3,047767854.

	A	B	C	D	E
1	10				
2	8				
3	4				
4	6				
5	3				
6	5				
7	1				
8	2				
9	1				
10	2				
11		4,2			
12		3,047767854			
13					
14					

Figura 1: Planilha de exemplo



- No **intervalo A1:A10**, informei dez números aleatórios.
- na **célula B11**, inseri a fórmula da média destes números, e
- na **célula B12**, o desvio padrão.

Vamos criar uma macro que calcule a média e o desvio padrão, sem usar as funções do LibreOffice, e que coloque os **resultados**, respectivamente, nas **células A11 e A12**.

## **Desvio padrão?**

Segundo a Wikipédia ([https://pt.wikipedia.org/wiki/Desvio\\_pad%C3%A3o](https://pt.wikipedia.org/wiki/Desvio_pad%C3%A3o)):

*“O desvio padrão é a medida mais comum da dispersão estatística (representado pelo símbolo sigma,  $\sigma$ ). Ele mostra o quanto de variação ou “dispersão” existe em relação à média (ou valor esperado).”*

Ainda, segundo a Wikipédia, sua fórmula é como mostra a figura2.

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$$

Figura 2: Desvio padrão

Ok, é uma macro muito simples e, na verdade, desnecessária, já que temos duas fórmulas no LibreOffice para isso: “MÉDIA()” e “DESVPAD()”, porém, serve para demonstrar como criar macros de forma bem simples.

Antes de criar a macro, veja o que tem que ser feito para calcular a média:

1. Apontar para a primeira célula do Intervalo A1:A10;
2. Ler o valor da célula atual;
3. Acumular o valor;
4. Apontar para a próxima célula do Intervalo A1:A10;
5. Terminaram as células do Intervalo A1:A10? Senão: Ir para 2;
6. Fazer a média = valor\_acumulado: 10.

Este é um algoritmo simples para calcular a média. Agora, vamos ver como calcular o desvio padrão:

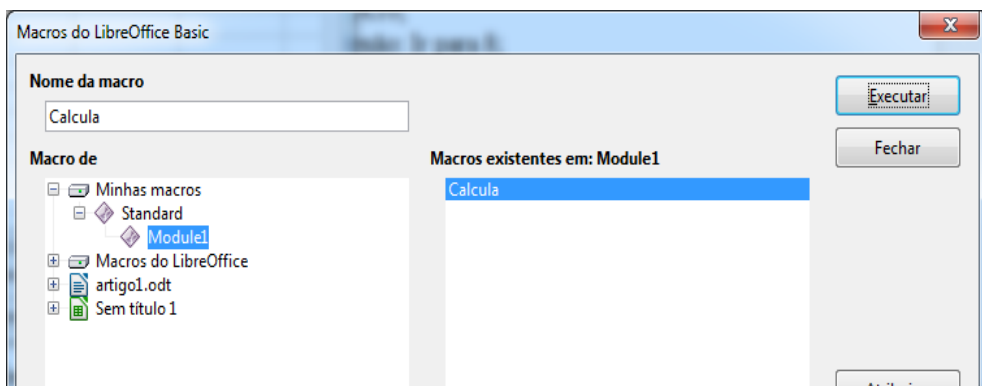
7. Apontar para a primeira célula do Intervalo A1:A10;
8. Ler o valor da célula atual;

9. Calcular a diferença = valor\_da\_célula – média;
10. Acumular o quadrado da diferença;
11. Apontar para a próxima célula do Intervalo A1:A10;
12. Terminaram as células do Intervalo A1:A10? Senão: Ir para 8;
13. Calcular divisão = valor\_acumulado / 9;
14. Calcular desvio = raiz quadrada da divisão.

Para concluir, é preciso colocar o valor da média na célula A11 e o valor do desvio na célula A12. Assim, podemos comparar com o valor retornado pelas fórmulas (B11 e B12), para ver se nossa macro funcionou.

### ***Editando a macro***

Selecione o menu **Ferramentas > Macros > Organizar macros > LibreOffice Basic...**. Você verá uma tela como a da figura 3.



*Figura 3: Editor de módulos*

Dentro da sua planilha, podem existir “módulos”, que contém código-fonte das macros. No seu caso, deverá aparecer uma macro padrão, chamada: “Main”. Ela é criada automaticamente e nada faz.

Vamos substituí-la pela nova macro. Clique no **botão Editar** para editar o **módulo Module1** e verá o editor de código, como o da figura 4.

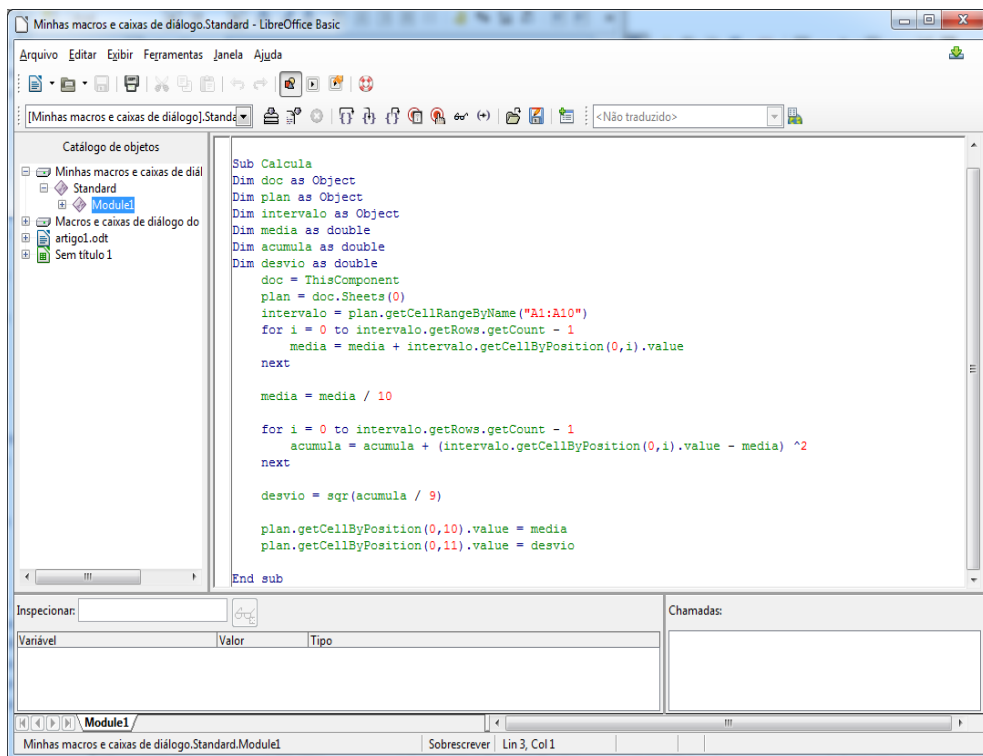


Figura 4: Editor de código

No seu caso, deve aparecer uma macro “Main” vazia. Algo assim: “Sub Main ... End Sub”. Selecione tudo, desde o “Sub Main”, até o “End sub” e apague. Depois, digite ou copie e cole o código-fonte a seguir.



```
Sub Calcula
Dim doc as Object
Dim plan as Object
Dim intervalo as Object
Dim media as double
Dim acumula as double
Dim desvio as double
    doc = ThisComponent
    plan = doc.Sheets(0)
    intervalo = plan.getCellRangeByName("A1:A10")
    for i = 0 to intervalo.getRows.getCount - 1
        media = media + intervalo.getCellByPosition(0,i).value
    Next

    media = media / 10

    for i = 0 to intervalo.getRows.getCount - 1
        acumula = acumula + (intervalo.getCellByPosition(0,i).value - media)
^2
    next

    desvio = sqr(acumula / 9)

    plan.getCellByPosition(0,10).value = media
    plan.getCellByPosition(0,11).value = desvio

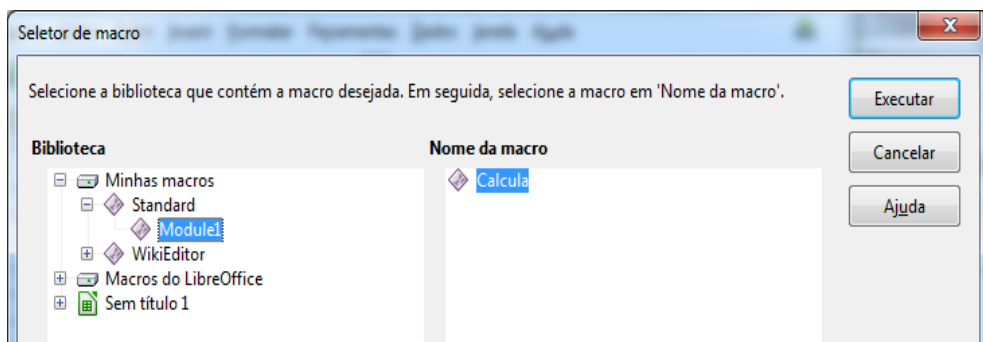
End sub
```



Calma! Muita calma nessa hora! Depois, eu explicarei um pouco sobre este código. Mas saiba que ele inclui os algoritmos que escrevi anteriormente.

Ok, agora clique em **A**rquivo > **S**alvar e volte à planilha.

Para executar a macro, selecione o menu **F**erramentas > **M**acros > **E**xecutar **m**acro... e selecione o **M**odule1 e a *macro* **C**alcular, conforme a figura 5.



*Figura 5: Selecionar a macro*

Ao clicar em **E**xecutar, verá que as células A11 e A12 aparecem preenchidas, com valores iguais aos das células B11 e B12.

## **Entendendo o Basic**

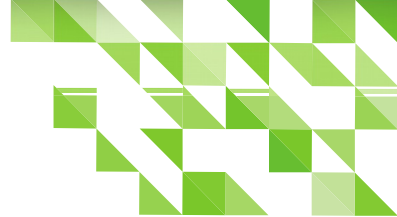
Dentro de um “Módulo”, podemos ter “procedimentos” ou “sub-rotinas”.

Nossa macro começa como “**S**ub **C**alcular” e termina com “**E**nd **S**ub”.

Estes comandos delimitam nossa macro.

Depois, eu declarei algumas variáveis locais, que precisei utilizar. Veja no quadro a seguir.

# COMO FAZER | *tutorial*



Dim doc as Object

Dim plan as Object

Dim intervalo as Object

Dim media as double

Dim acumula as double

Dim desvio as double

Algumas variáveis são do *tipo Object*, porque, na verdade, não conterão valores, mas apontarão para objetos da nossa Planilha.

Exemplo:

- “**doc**” aponta para o “workbook” atual, obtido a partir de “ThisComponent”;

Calcular a média é trivial:

```
for i = 0 to intervalo.getRows.getCount - 1
    media = media + intervalo.getCellByPosition(0,i).value
next

media = media / 10
```

- “**plan**” aponta para a primeira planilha, obtida a partir da coleção de planilhas (“Sheets”) do “workbook”: “doc.Sheets(0)”;
- “**intervalo**” aponta para o nosso intervalo A1:A10, obtido a partir do método “getCellRangeByName” (obter intervalo de células por nome): “plan.getCellRangeByName("A1:A10")”;

Uma vez que se tenha essas referências, é possível acessar o intervalo de células através da *variável intervalo*, ou mesmo a planilha, através da *variável plan*.

# COMO FAZER | *tutorial*



O *método* `getCellByPosition` obtém o valor de uma célula de um intervalo. Os parâmetros são: Coluna e Linha. Estamos obtendo o valor da primeira célula, da primeira linha (linha zero). A *propriedade* `value` retorna o conteúdo da célula.

Então, fazemos um “Loop” repetitivo, incrementando a *variável* `i`, de zero até a quantidade de linhas do intervalo. O método “`getCount`” (obter contagem), chamado a partir do método “`getRows`” (obter linhas), retorna a contagem de linhas.

O cálculo do desvio padrão também é simples.

```
for i = 0 to intervalo.getRows.getCount - 1
    acumula = acumula +
        (intervalo.getCellByPosition(0,i).value - media) ^2
next

desvio = sqr(acumula / 9)
```

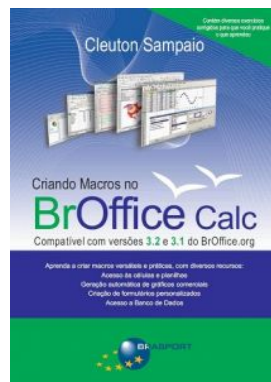
Em Basic, o operador de potenciação é o circunflexo (^).

Para calcular a raiz quadrada, usamos a *função* `sqr()`.

Ok. Só falta uma coisa: colocar os dois resultados na planilha.

- Média: “`plan.getCellByPosition(0,10).value = media`”;
- Desvio: “`plan.getCellByPosition(0,11).value = desvio`”;

A *propriedade* `value`, das células, serve para obter ou alterar o seu conteúdo.

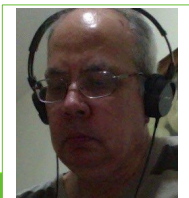


*Livro “Criando macros no BrOffice Calc*

# COMO FAZER | *tutorial*

## *E agora?*

Vimos o básico, do básico, do básico da programação de macros em Basic. Estude bem a maneira como eu me referi aos *objetos workbook*, *planilha* e *intervalo*, e reveja a maneira como eu li e alterei seus conteúdos.



**Cleuton Sampaio** - Mestre, formado pelo IBMEC, professor, desenvolvedor e escritor profissional, sendo o autor do livro: *Criando macros no BrOffice Calc*, editado pela Brasport.