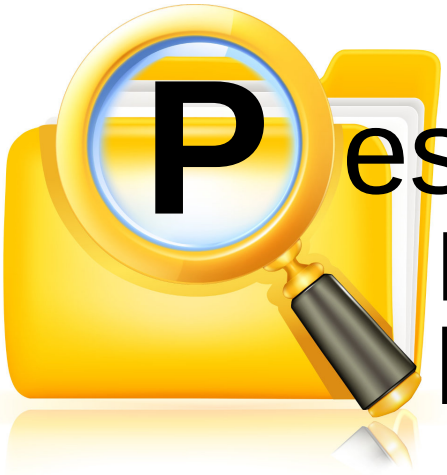


Pesquisa por Expressões Regulares

Por Julio Cezar Neves

Era uma vez um pobre escritor que publicou um livro com cerca de 500 páginas sobre o interpretador *Shell* e que precisava achar uma variável do sistema na qual, dias antes ,ele havia detectado um erro. Mas não lembrava o nome da variável, nem em qual página do livro ela seria encontrada. A única dica que ele tinha é que toda variável do sistema em *Shell*, começa por um cifrão seguido somente por letras maiúsculas.

Seria uma tarefa dolorosa procurar essa agulha no palheiro, mas salvou-o o fato do *LibreOffice* ter uma facilidade incrivelmente útil, mas totalmente desconhecida por 99,99999% de seus usuários, chamada pesquisa por *Expressões Regulares*.

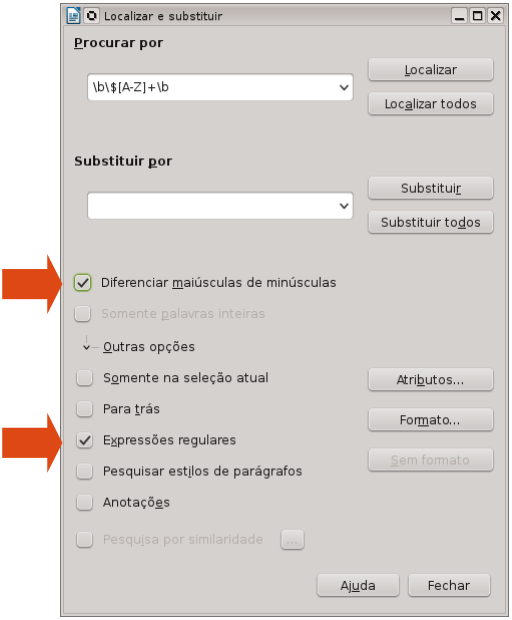
No caso específico ao qual nos referimos, bastou clicar em *Editar* > *Localizar e Substituir* e na caixa de diálogos que se apresentou, clicar em ***Outras Opções*** e marcar ***Expressões Regulares.*** Na caixa ***Procurar por*** escrever:

```
\b\$[A-Z]+\b
```

Onde:

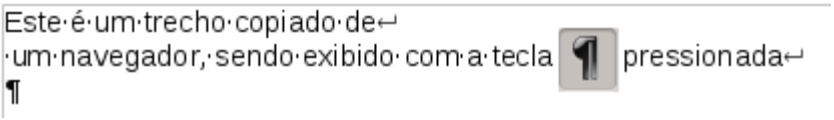
ER	Significa
\b	Casa com os limites de uma palavra (anterior ou posterior). Por exemplo, \bbola casa com bola ou com bolada, bola\b casa com bola e com carambola, mas \bbola\b casa somente com bola.
\$	O cifrão ($) escrito sem a contrabarra (\) o precedendo, é uma *Expressão Regular* que casa com finais de parágrafos. A barra invertida (\) foi colocada para que o cifrão ($) seja interpretado pelo seu valor literal, e não como uma *Expressão Regular*. Isto é, a contrabarra (\) tira os superpoderes do caractere seguinte, exceto para as combinações \n, \t, \> e \<
[A-Z]	Isto é uma lista que casa com qualquer letra, mas indica as maiúsculas
+	Obriga que a entidade anterior (no caso, letra maiúscula) ocorra pelo menos uma vez
\b	O limite do fim da *Expressão Regular*

Veja a caixa de diálogo (que também poderia ter sido gerada por um **CTRL+h**) após preenchida:



Repare que os checkboxes “Expressões Regulares” e “Diferenciar maiúsculas de minúsculas estão ativados. O primeiro por motivos óbvios mas é o segundo que faz com que o casamento, que já era preferencialmente com maiúsculas em virtude da lista **[A-Z],** agora seja obrigatoriamente só com letras em caixa alta.

O uso de Expressões Regulares permite evitar uma série de procedimentos repetitivos. Veja a imagem a seguir que é uma cola de página em HTML:

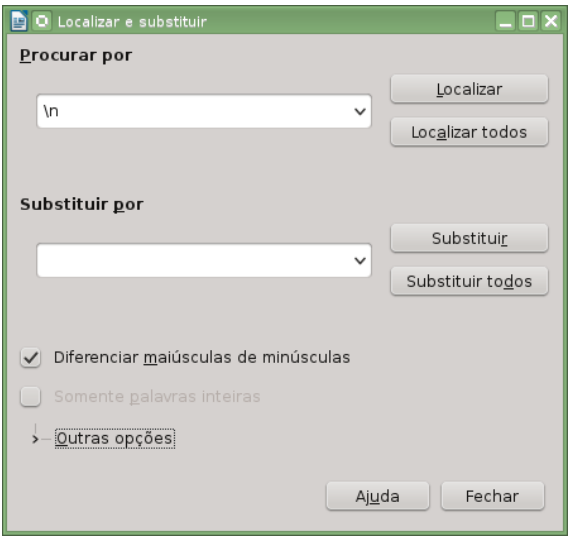


Repare que o texto apresenta dois tipos de terminadores que são exibidos como uma seta quebrada **(↵)** ou como um pi **(¶).** O pi **(¶)** é gerado por um fim de linha convencional (um **<ENTER>**) e a seta quebrada **(↵)** também chamada de *Hard Enter* é conseguida por um **<SHIFT>+<ENTER>**.

Em um texto pequeno como neste que vimos é só trocar "na unha" esses caracteres por um espaço em branco, mas se fossem muitas linhas, você teria de perder um tempão para fazer assim. O macete então é proceder da seguinte forma:

- **<CTRL>+h** – Para chamar o diálogo Substituir e alterar;
- Marcar o *checkbox "Expressões Regulares"* ;
- Na caixa Procurar por, colocar **\n,** que significa fim de linha;
- Na caixa Substituir por dar um espaço em branco;
- Clicar em "Localizar todos" e em seguida em "Substituir todos".

Pronto! Seu texto já está legal.



Eu sei que essa explicação não está das melhores, mas a causa disso é que você não sabe usar Expressões Regulares, mas uma coisa eu garanto: se você usar essa receita de bolo, nunca mais você vai precisar fazer essa tarefa "na unha".

Em mais um exemplo, o texto era assim:

- Flu
- Fluminense
- Fluzinho
- Vasco
- Vasquinho

Mas em virtude da situação atual do Campeonato Brasileiro, na caixa ***Procurar por***, tive que aplicar a seguinte *Expressão Regular*:

- Flu(minense|zinho)?|Vas(co|quinho)

Onde:

- (minense|zinho)?

Nesse caso, os parênteses têm dupla finalidade:

Limitar o escopo do ***ou lógico*** produzido pela barra vertical **(|);**

Servem para agrupar o que se tornará opcional em virtude do ponto de interrogação.

Ou seja, essa *Expressão Regular* casará com **Flu, Fluminense** e com **Fluzinho;**

- (co|quinho)

Da mesma forma que o anterior, o *ou lógico* produzido pela barra vertical **(|)** casa **Vas** com **co** ou **quinho**.

E na caixa Substituir por, me senti compelido a escrever Mengão, voltando:

- Mengão
- Mengão
- Mengão
- Mengão
- Mengão

Obviamente isso é uma gozação sem nenhuma conotação prática, mas serve para avaliar a flexibilidade do uso de expressões regulares.

Outra coisa bacana é que você pode reter valores para uso futuro. Vamos dizer que meu texto tenha um monte de datas no formato **dd/mm/aaaa** e quero passar para o formato **dd/mm/aa,** mas somente se a data for do século 21, ou seja, com o ano tipo **20NN** onde **NN** pode ser qualquer número. Olha só o que tenho:

- 05/04/1947
- 12/11/2013
- 08/11/1919
- 12/12/2012
- 17/01/1980
- 11/11/2011
- 08/11/1984

Na caixa de diálogos **Procurar por** vou colocar **([0-9]{2}/[0-9]{2}/)20([0-9]{2})** e na caixa **Substitui por** colocarei **$1$2.** Após clicar em "Substituir todos" obterei o seguinte resultado:

- 05/04/1947
- 12/11/13
- 08/11/1919
- 12/12/12
- 17/01/1980
- 11/11/11
- 08/11/1984

Vamos entender o que se passou. Primeiro veremos a entrada:

0-9]{2} é o mesmo que **[0-9][0-9],** ou seja qualquer algarismo ocorrendo duas vezes. Esta ***Expressão Regular*** foi usada para definir o dia, o mês e os dois últimos algarismos do ano que começasse por **20.** Repare que usei um par de parênteses pegando desde o início do dia até após a barra que separa o mês do ano **(dd/mm/)** e usei outro par para abraçar os 2 últimos algarismos do ano **(aa).** Pois é: o segredo são esses parênteses, porque eles guardam o texto que está no seu interior para uso futuro.

- Uso futuro? Que viagem é essa? Fumou orégano?

- Não, os textos casados pelos parênteses podem ser recuperados no diálogo **Substituir por,** usando-se um cifrão precedendo o número de ordem dos pares de parênteses, ou seja, o **$1** receberá **dd/mm/** e **$2** receberá **aa.**

Usando este mesmo macete vamos procurar por palavras repetidas e consecutivas em um texto. Basta colocar na caixa de diálogos **Procurar por** a seguinte *Expressão Regular*:

```
[\b([a-zA-Z]+) \1\b
```

Onde **[a-zA-Z]+** casa com palavras e **\1** faz, na caixa **Procurar por,** o mesmo papel que **$1** exerceria na caixa **Substituir por,** isto é, repete o texto casado no interior dos parênteses. Se porventura não fossem colocados os limites anterior e posterior das palavras **(\b),** textos como **rebate bateria** atenderiam à Expressão Regular pois têm o texto **bate bate.**

Bem amigos do *LibreOffice*, como vocês viram *Expressões Regulares* quebram um galhão, mas são cheias de macetes. É muito importante saber que todos os bons editores (latex, vi, emacs, ...) usam Expressões Regulares, todas as linguagens de programação modernas usam *Expressões Regulares*. Caso você trabalhe com redes, pode otimizar e agilizar as regras de *firewall* ou de *proxy* com *Expressões Regulares*.

Mas o que mostramos aqui foi só um pouquinho. Só um aperitivo para deixá-los com sede de saber mais.

E por falar em sede de saber, você pode aprender a usar *Expressões Regulares*, utilizando a Ajuda do LibreOffice.

JULIO NEVES - O 4º UNIX do mundo nasceu na Cidade Maravilhosa, mais precisamente na Cobra Computadores, onde à época trabalhava o Julio. Foi paixão à 1ª vista! Desde então, (1980) atua nessa área como especialista em Sistemas Operacionais e linguagens de programação. E foi por essa afinidade que quando surgiu o Linux foi um dos primeiros a estudá-lo com profundidade e adotá-lo como Sistema Operacional e filosofia de vida.