



## Como importar um arquivo texto para uma planilha do Calc

Por Johnny Furusato

Outro dia tive um pequeno trabalho e, o fato de conhecer um pouco de programação, foi um grande auxiliou.

Recebi um arquivo que era um relatório em formato de texto gerado em um sistema da IBM daqueles antigos, feito em COBOL, onde era necessário descobrir diversos tipos de equipamentos, suas características e os correspondentes locais em que eles estavam.

Ótimo! Basta colocar em uma planilha do Calc, aplicar filtros e já está resolvido!

Foi o que pensei sem ter olhado o arquivo.

Mas, não foi bem assim!

Parece hoje algo trivial, mas quando usam as impressoras matriciais, os relatórios são adequados com tamanho de linhas e margens fixas. Essas impressoras foram bastante usadas com formulários já pré-impressos, na forma de formulários contínuos e um bom exemplo são as Notas Fiscais de empresas.

Era assim até pouco tempo antes da chegada da Nota Fiscal Eletrônica.

Os relatórios quando são exportados para o formato digital, não mantém o mesmo aspecto do papel.



Assim, uma primeira dificuldade é saber qual o tamanho de caracteres que ocupam uma linha de impressão e qual é a quantidade de linhas que formam uma página.

Se você abrir esse arquivo de texto em um editor como o Bloco de Notas ou mesmo o Writer, fica difícil saber isso.

Então a primeira coisa é usar um editor hexadecimal que mostrará os caracteres ASCII invisíveis que simbolizam o final de uma linha e retorno de carro (CR/LF).

Outra opção é usar um editor gratuito como o Notepad++ para Windows ou Notepadqq para Linux.

Deste modo pude descobrir que o meu relatório possui linhas em branco, intercaladas, e 80 colunas.

Os dados do relatório mostrados aqui são fictícios.

Abaixo o arquivo aberto no Bloco de Notas - Notepad comum.

```
CONTROLADORIA GERAL PAGINA: 11
589361 - CENTRO DE CONTROLE E PESQUISA INTERNAC/SP
589361 - CENTRO DE CONTROLE E PESQUISA INTERNAC/SP
00001 - TESOURO NACIONAL
INVENTARIO DOS BENS MOVEIS EMISSAO: 19/01/2015
-----
CONTA : 92.512.09.07 - EQUIPAMENTOS DE PROCESSAMENTO DE DADOS
33.412.88.00
-----
UORG : 002519 - DIVISAO DE ACOMPANHAMENTO E CONTROLE CIENTIFICO DA
-----
ITEM | SITUACAO / CODIGO E DESCRICAO DO MATERIAL | VALOR DEPRECIADO | VALOR BEM
TRANSPORTE | | 57.824,62 | 190.335,03
000102 | BOM / (00044219-5) INJETOR DE ENERGIA PD-9902 G-ENT MIDSPAN/POWER | |
50 A 60 HZ 110/220 VAC R3 45 FEMES IEEE 802.3AF COM CABO DE ALIM | 21,20 | 283,00
0005647881.0000000000 24/10/2014 | |
TOTAL | | 57.845,82 | 190.618,03
-----
UORG : 003336 - CENTRO DE CONTROLE E PESQUISA LIVRE INTERNACIONAIS
-----
ITEM | SITUACAO / CODIGO E DESCRICAO DO MATERIAL | VALOR DEPRECIADO | VALOR BEM
PATRIMONIO NR. ANTERIOR TOMBAMENTO | |
000001 | BOM / (00054701-9) AGENDA ELETRONICA TIPO DE BOLSO POCKET PC, MAR | 696,96 | 774,06
CA TUNGSTEN C | 0008486283.0000000000 31/07/2006 | |
000002 | BOM / (00021239-2) PALM TOP COM CELULAR / IPHONE, 8GB, MARCA APPL | 27,00 | 30,00
E, MODELO A | 0007561653.0000000000 24/02/2010 | |
```



Em na imagem a seguir, um zoom de trecho aberto no Notepad. Aparentemente, não há linhas em branco e intercaladas.

000005	BOM / (00027746-0) MICROCOMPUTADOR DE MESA UDP INTEL CORE 3 I5 34 70, 4GB RAM, HD 500 GB, DVD-RW 0004826093 0000000000 24/05/2013	666,80	2.223,00
000006	BOM / (00014432-0) MONITOR MICROCOMPUTADOR MONITOR LED 18.5" 0004835261 0000000000 24/05/2013	83,60	279,00
A TRANSPORTAR			8.113,81

Agora veja a visualização correta no Notepad++, mostrando as linhas em branco.

```
1318 |-----|
1319 |
1320 |000005 | BOM / (00027746-0) MICROCOMPUTADOR DE MESA UDP INTEL CORE 3 I5 34 |
1321 |
1322 |      | 70, 4GB RAM, HD 500 GB, DVD-RW |
1323 |
1324 |      | 0003826293 0000000000 24/05/2013      666,80      2.223,00 |
1325 |
1326 |-----|
1327 |
1328 |000006 | BOM / (00014432-0) MONITOR MICROCOMPUTADOR MONITOR LED 18.5" |
1329 |
1330 |      | 0003835461 0000000000 24/05/2013      83,60      279,00 |
1331 |
1332 |-----|
1333 |
1334 | A TRANSPORTAR      8.113,81 |
1335 |
1336 |-----|
```

A minha dificuldade era como transportar esse relatório de mais de 80 páginas e milhares de linhas para uma planilha do Calc, de modo que pudesse classificar depois pela UORG, já que era a informação que orientava a “quebra” de folhas do relatório. E ter os itens distribuídos pelas colunas: ITEM, SITUAÇÃO, CÓDIGO E DESCRIÇÃO DO MATERIAL, PATRIMÔNIO E VALORES.

Repare que, não é possível simplesmente pedir para o Calc importar o arquivo como se fosse delimitado por colunas, pois os campos estão dispersos pelas linhas.



Outra dificuldade é que o campo DESCRIÇÃO às vezes ocupa mais de uma linha, bem como a quebra de uma nova UORG não ocorre sempre no começo da página.

Como resolvi?

Recorri a velha programação procedural.

Fiz uma macro em LibreOffice Basic no Calc, que examina o arquivo linha a linha e procura determinados padrões.

Assim, os padrões que me importavam foram colocados em um campo do tipo strings e sempre que encontrava um deles tomava uma decisão adequada.

No meu relatório, como exemplo, pude notar que sempre há uma palavra “UORG” ocupando as posições a partir da coluna 03, um campo “(“ em outra posição fixa e um campo com conteúdo “0000000000” na coluna 22.

A partir disso, consegui separar as informações por coluna no Calc, recorrendo sempre à função Mid do Basic.

Tive que tomar um cuidado adicional para as descrições que ocupavam mais de uma linha.

Fiz um código rápido, funcional e sujo - sem me preocupar se era a melhor forma de programar, mas que me atendeu exatamente como precisava.

Os principais recursos que usei foram a função MID, e os métodos: GetCellByPosition, setString e setValue.

- A função MID - Mid (String, posição inicial, quantidade de caracteres), recupera parte de uma string.
- O método GetCellByPosition (coluna, linha), localiza a célula da planilha.
- O método setString(String), armazena na célula referenciada anteriormente pelo método GetCellByPosition um conteúdo do tipo String.
- O método setValue(Número), armazena na célula referenciada anteriormente pelo método GetCellByPosition um conteúdo do tipo Numérico.



Abaixo uma visão da planilha do jeito que precisava.

A linha 1 - cabeçalho - com os nomes dos campos foi incluída manualmente.

A	B	C	D	E	F	G	H	I	J	
1	UORG	ITEM	SITUACAO	CODIGO	DESCRICAO	PATRIMONIO	NUM_ANTERIOR	DTOMBAMENTO	DEPRECIADO	VALOR_BEM
2	1029	180M	00031441-0	MONITOR - CRISTAL LIQUIDO 17 POLEGADA ANTI-REFLEXIVA 30/81 KHZ 60/76 HZ 270 MM - 338 MM - VIDEO COLOR		9712200000000000	19/10/2005	445,70	460,20	
3	1259	280M	00037421-0	MONITOR - CRISTAL LIQUIDO 17 POLEGADA ANTI-REFLEXIVA 30/81 KHZ 60/76 HZ 270 MM - 338 MM - VIDEO COLOR		9717200000000000	27/06/2008	351,00	390,00	
4	11165	1180M	00014432-0	MONITOR MICROCOMPUTADOR MONITOR LED 18.5"		8835491000000000	24/05/2013	83,60	279,00	
5	11103	1280M	00014432-0	MONITOR MICROCOMPUTADOR MONITOR LED 18.5"		8835519000000000	24/05/2013	83,60	279,00	
6	11105	1280M	00014432-0	MONITOR MICROCOMPUTADOR MONITOR LED 18.5"		8835519000000000	24/05/2013	83,60	279,00	

Segue como curiosidade o código com alguns poucos comentários. Ele pode ser adaptado e usado para, a partir de um outro texto, aprender como preencher células em uma planilha do Calc.

```
1 Public Sub LeArquivoTexto()  
2 ' Autor JOHNNY TAKARA FURUSATO - sp24horas@gmail.com  
3 ' Grava em uma planilha do Calc dados extraídos de um arquivo texto  
4 ' quando não é possível a importação direta parametrizada por colunas  
5  
6 Dim Arquivo As Integer  
7 Dim CaminhoArquivo As String  
8 Dim MyString As String  
9 Dim teste, farol_uorg As String  
10 Dim ContLinha As Long  
11 Dim sinal as Integer  
12 Dim oMovPasta as Object  
13 Dim oCelula as Object  
14 Dim oPasta as Object  
15 Dim oDocumento as Object  
16 ' Declara que vai usar a primeira planilha  
17 oDocumento = ThisComponent  
18 oPasta = oDocumento.Sheets  
19 oMovPasta = oPasta(0)  
20 ' Inicializa variaveis  
21 ContLinha = 0  
22 sinal = 0  
23 teste = " "  
24 'Configura a leitura do arquivo  
25 Arquivo = FreeFile  
26 CaminhoArquivo = "d:\teste\revista.txt"  
27  
28 'Abre o arquivo para leitura  
29 Open CaminhoArquivo For Input As Arquivo  
30  
31 'Lê o conteúdo do arquivo linha a linha  
32 ' Enquanto não for fim do arquivo  
33 Do While Not EOF(Arquivo)  
34 ' Lê o conteúdo da linha e armazena em MyString  
35 Line Input #Arquivo, MyString  
36 ' Coloca na váriavel teste uma string de 4 posições iniciando na coluna 03  
37 teste = Mid(MyString, 3, 4)  
38 ' Se nesta linha tiver o padrão "UORG"  
39 if teste = "UORG" then  
40 ' Vou acrescentar uma linha na planilha
```



```
41     contlinha = contlinha + 1
42     ' vou colocar o código UORG de 06 posições iniciando na coluna 11
43     farol_uorg = Mid(MyString, 11, 6)
44     ' Na célula A + Contlinha
45     oCelula = oMovPasta.GetCellByPosition(0, ContLinha)
46     ' Coloco o valor numérico de farol_uorg
47     oCelula.SetValue(farol_uorg)
48     ' Se for a primeira UORG descoberta, quero continuar armazenando na mesma linha
49     contlinha = contlinha - 1
50 endif
51 ' comparo a linha e vejo se é outro padrão
52 teste = Mid(MyString, 17, 1)
53 if teste = "(" then
54     REM gravo informações na mesma linha da UORG
55     contlinha = contlinha + 1
56     ' sei que nessa linha encontrei o padrão
57     sinal = 1
58     oCelula = oMovPasta.GetCellByPosition(0, ContLinha)
59     oCelula.SetValue(farol_uorg)
60     item = Mid(MyString, 2, 6)
61     oCelula = oMovPasta.GetCellByPosition(1, ContLinha)
62     oCelula.SetValue(item)
63     situacao = Mid(MyString, 11, 3)
64     oCelula = oMovPasta.GetCellByPosition(2, ContLinha)
65     oCelula.SetString(situacao)
66     codigo = Mid(MyString, 18, 10)
67     oCelula = oMovPasta.GetCellByPosition(3, ContLinha)
68     oCelula.SetString(codigo)
69     descricao = Mid(MyString, 30, 46)
70     'avanço duas linhas para ver se a descrição ocupa duas linhas
71     Line Input #Arquivo, MyString
72     Line Input #Arquivo, MyString
73 endif
74 'se já achei o padrão anterior "("
75 if sinal = 1 then
76     'se o campo não tiver o padrão abaixo, preciso completar o campo descrição e gravar
77     teste = Mid(MyString, 22, 10)
78     if teste <> "0000000000" then
79         descricao2 = descricao & Mid(MyString, 11, 58)
80         oCelula = oMovPasta.GetCellByPosition(4, ContLinha)
81         oCelula.SetString(descricao2)
82     else
83         oCelula = oMovPasta.GetCellByPosition(4, ContLinha)
84         oCelula.SetString(descricao)
85     endif
86     sinal = 0
87 endif
88 'vejo se atende o padrão "0000000000"
89 teste = Mid(MyString, 22, 10)
90 if teste = "0000000000" then
91     ' gravo o conteúdo numérico de patrimônio
92     PATRIMONIO = Mid(MyString, 11, 10)
93     oCelula = oMovPasta.GetCellByPosition(5, ContLinha)
94     oCelula.SetValue(PATRIMONIO)
95     ' gravo o conteúdo string de NR_ANTERIOR
```



```
96 NR_ANTERIOR = Mid(MyString, 22, 10)
97 oCelula = oMovPasta.GetCellByPosition(6, ContLinha)
98 oCelula.setString(NR_ANTERIOR)
99 ' gravo o conteúdo de TOMBAMENTO
100 TOMBAMENTO = Mid(MyString, 33, 10)
101 oCelula = oMovPasta.GetCellByPosition(7, ContLinha)
102 oCelula.setString(TOMBAMENTO)
103 ' gravo o conteúdo de VAL_DEPRECIADO
104 VAL_DEPRECIADO = Mid(MyString, 48, 10)
105 oCelula = oMovPasta.GetCellByPosition(8, ContLinha)
106 oCelula.setString(VAL_DEPRECIADO)
107 'gravo o conteúdo de VALOR_BEM
108 VALOR_BEM = Mid(MyString, 69, 10)
109 oCelula = oMovPasta.GetCellByPosition(9, ContLinha)
110 oCelula.setString(VALOR_BEM)
111 endif
112 'conclui o comando loop - para ler todas as linhas
113 Loop
114
115 'Fecha o arquivo
116 Close Arquivo
117 'Fecha a macro
118 End Sub
119
```

Espero que seja útil para você.

Boa sorte!



**Johnny Furusato** - Advogado e Gerente de TI com ampla experiência na área de programação, segurança e tecnologia da informação. Sempre que pode compartilha o pouco que sabe. Acredita que podemos transformar o mundo com educação de qualidade e a vivência das virtudes.