Calculadora PEPS no Excel

Querendo ajudar os amigos vou criar uma calculadora PEPS ou (em inglês, *First in First Out* = FIFO) para estoques que ele mantinha. Parece que a tributação nas compras de estoque tornam-se crescentemente difíceis quando você está comprando o mesmo estoque múltiplas vezes. Mantendo o P&L sobre um estoque torna-se difícil baseado primeiro que entra, primeiro que sai (PEPS).

Meu entendimento do PEPS é que os bens comprados são vendidos com base nos preços dos primeiros produtos comprados. Usarei um exemplo simples;

Compramos 2 itens por um preço de $10 cada.  Na Terça compramos mais 2 itens por $15 cada.  Agora na Quarta vendemos 3 dos itens por $17 cada.  Precisamos encontrar o lucro ou prejuízo para propósito de tributação. Se usarmos PEPS, teremos um lucro de:

17 \* 2 - 10 \* 2 + 1 \* 17 - 1 \* 15

ou

34 - 20 + 17 - 15

=$16

Então tivemos realizado um lucro de $16 de acordo com o PEPS. Isto torna-se mais e mais difícil rastrear mais itens que são comprados e vendidos por diferentes preços.

 O que segue é um cálculo PEPS mais complexo baseado no arquivo anexado.



Aqui é onde o VBA do Excel pode ajudar.  O que segue é um procedimento CalcPEPS muito útil em que classificaremos o preço de fechamento por meio de um cálculo PEPS.

 Option Explicit

Sub CalcPEPS() 'VBA do Excel VBA para calcular PEPS sobre bens comprados e vendidos.

Dim vendido As Long

Dim i As Integer

Dim j As Integer

Dim cnt As Long

Dim venda As Double

Dim ar As Variant

Dim Var As Variant

Range("G10:G1000").ClearContents

ar= Range("C10", Range("C65536").End(xlUp)) 'Compra

Var=Range("D10", Range("D65536").End(xlUp)) 'Preço

   For i=10 To Range("A" & Rows.Count).End(xlUp).Row

      venda=0

      vendido=Range("E" & i)

      j=1

          Do While vendido > 0

             cnt=ar(j, 1)

             ar(j, 1)=IIf(ar(j, 1) > vendido, ar(j, 1) - vendido, 0) 'iif

             vendido=vendido - (cnt - ar(j, 1))

             venda=venda + (cnt - ar(j, 1)) \* Var(j, 1)

             j=j + 1

         Loop

      Range("G65536").End(xlUp)(2)=venda 'saída

   Next i

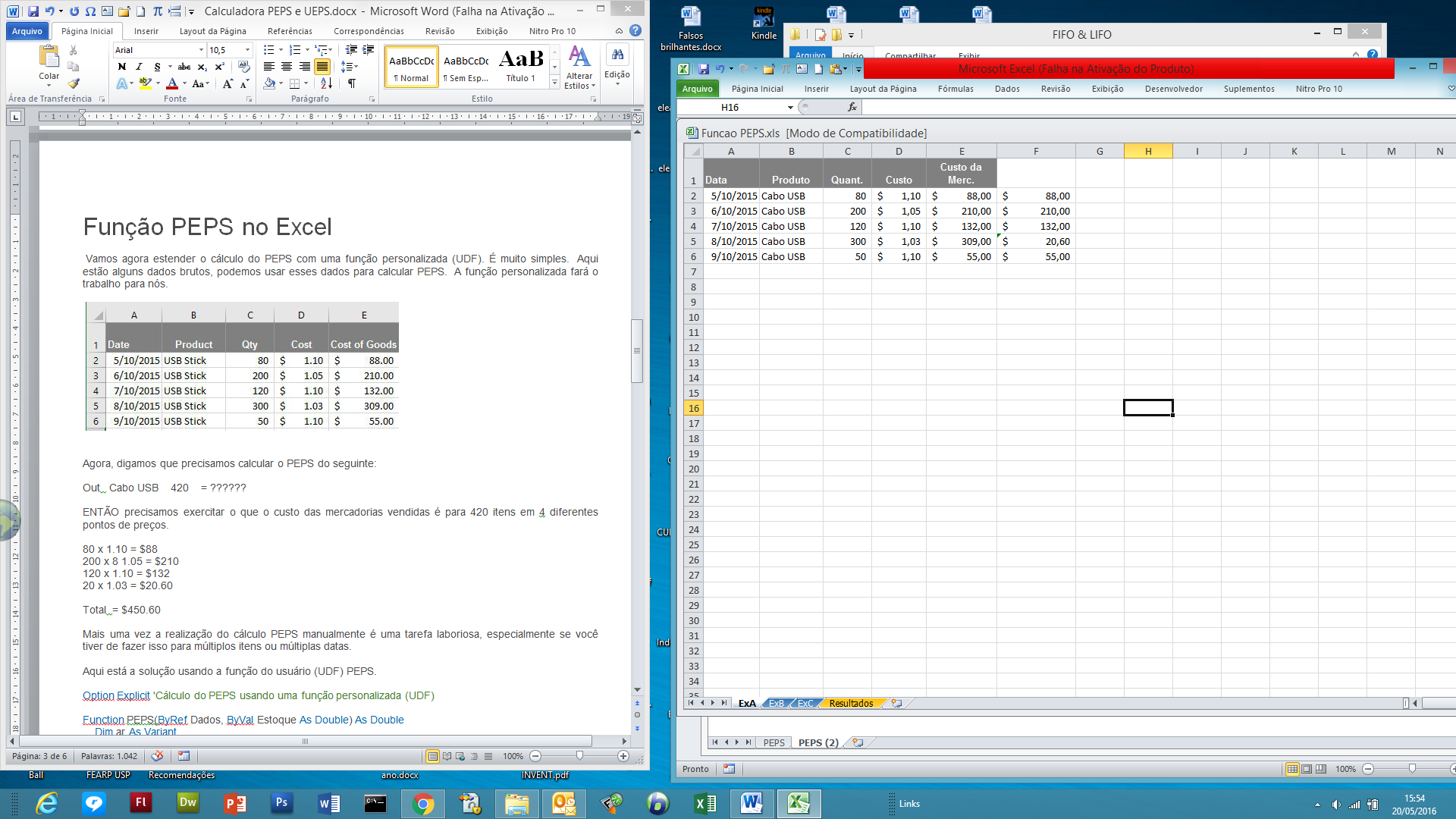
End Sub

O arquivo Excel que segue mostra um exemplo mais complexo do método PEPS acima usando VBA.

 Calculadora PEPS e UEPS.xls

Função PEPS no Excel

 Vamos agora estender o cálculo do PEPS com uma função personalizada (UDF). É muito simples.  Aqui estão alguns dados brutos, podemos usar esses dados para calcular PEPS.  A função personalizada fará o trabalho para nós.



Agora, digamos que precisamos calcular o PEPS do seguinte:

Out   Cabo USB    420    = ??????

ENTÃO precisamos exercitar o que o custo das mercadorias vendidas é para 420 itens em 4 diferentes pontos de preços.

80 x 1.10 = $88

200 x 8 1.05 = $210

120 x 1.10 = $132

20 x 1.03 = $20.60

Total  = $450.60

Mais uma vez a realização do cálculo PEPS manualmente é uma tarefa laboriosa, especialmente se você tiver de fazer isso para múltiplos itens ou múltiplas datas.

Aqui está a solução usando a função do usuário (UDF) PEPS.

Option Explicit 'Cálculo do PEPS usando uma função personalizada (UDF)

Function PEPS(ByRef Dados, ByVal Estoque As Double) As Double  
    Dim ar As Variant  
    Dim i As Long  
    Const ColQuantidade  As Long = 1 'Número COl da coluna quantidade para o cálculo PEPS  
    Const ColCusto As Long = 2 'Número COl da coluna custo para o cálculo PEPS calc  
      
    ar = Dados  
      
    For i = LBound(ar, 1) To UBound(ar, 1)  
        If Estoque < ar(i, ColQuantidade) Then  
            PEPS = PEPS + Estoque \* ar(i, ColCusto)  
            Exit Function  
        Else  
            PEPS = PEPS + (ar(i, ColQuantidade) \* ar(i, ColCusto))  
            Estoque = Estoque - ar(i, ColQuantidade)  
            If Estoque <= 0 Then Exit Function  
        End If  
    Next i  
       
End Function

O que segue é a fórmula PEPS

=PEPS(ExA!$C$2:$D$6;C2)

onde C2:D6 é a quantidade e o custo (nesta ordem). E C2 contém a quantidade vendida PEPS.

O arquivo anexado mostra o cálculo básico e o mais avançado do PEPS.  Ele leva o exemplo simples acima um passo adiante.  É muito útil e pode ser estendido muito além se for necessário.  A função PEPS fornecida acima é um pequeno vislumbre no mundo do cálculo do First in First Out.



O que está acima é um exemplo dos resultados do arquivo. Ele adiante o conceito um passo a mais para calcular o valor do estoque remanescente. Ele mostra como a Função PEPS pode ser estendida para critérios múltiplos e produtos.

Calculadora UEPS

Último que Entra, Primeiro que sai, abreviadamente, UEPS (em inglês, *Last in First Out* = LIFO) é um método semelhante ao método PEPS descrito na página calculadora PEPS. Meu entendimento do UEPS é que as mercadorias compradas são vendidas com base no preço do último produto comprado.  Usarei um exemplo simples;

Usarei o mesmo exemplo da Calculadora PEPS mas ao contrário.

Compramos 2 itens de estoque pelo preço de $10 cada.  Na Terça Feira compramos mais 2 itens de estoque por $15 cada.  Agora na Quarta-Feira vendemos 3 itens por $17 cada. Precisamos encontrar o lucro ou prejuízo para propósitos de tributação. Se usarmos UEPS teremos um lucro de:

17 \* 2 - 15 \* 2 + 1 \* 17 - 1 \* 10

ou

34 - 30 + 17 - 10

=$11

Então temos realizado um lucro de $11. Isto torna-se mais e mais difícil para rastrear quanto mais itens forem comprados e vendidos por diferentes preços.

 O que segue é um cálculo UEPS mais complexo baseado no arquivo anexado.



 Aqui está onde o VBA pode ajudar. O que segue é um procedimento muito simples de como classificaremos os preços de fechamento para o cálculo UEPS.

Option Explicit

Sub CalcUEPS()

Dim vendido As Long

Dim i As Integer

Dim j As Integer

Dim cnt As Long

Dim venda As Double

Dim ar As Variant

Dim Var As Variant

Dim n As Integer

Range("G10:G1000").ClearContents 'limpa a Coluna Venda

ar=Range("C10", Range("C65536").End(xlUp)) 'Compra

Var=Range("D10", Range("D65536").End(xlUp)) 'Preço

n=Cells.Find("Data").Row + 1  'Linha Inicial

   For i=Range("A" & Rows.Count).End(xlUp).Row To n Step -1

      venda=0

      vendido=Range("E" & n)

      j=UBound(ar, 1)

         Do While vendido > 0

            cnt=ar(j, 1)

            ar(j, 1)=IIf(ar(j, 1) > vendido, ar(j, 1) - vendido, 0)

            vendido=vendido - (cnt - ar(j, 1))

            venda=venda + (cnt - ar(j, 1)) \* Var(j, 1)

            j=j - 1

        Loop

     n=n + 1

    Range("G65536").End(xlUp)(2)=venda 'saída

   Next i

End Sub

O arquivo seguinte mostra um exemplo mais complexo do que está acima.