**FUNCIONALIDAD PARA LOS FILTROS APLICABLES A LAS DIFERENTES HOJAS**

**‘Definir la funcionalidad**

Sub Filtros()

**‘Definir variables**

Dim wsPreformas As Worksheet

Dim wsGrumos As Worksheet

Dim wsTerminados As Worksheet

Dim lastRow As Long

Dim lastCol As Long

Dim rngData As Range

Dim i As Long

Dim j As Long

Dim descripcion As String

Dim numeroExtraido As Double

Dim stkU As Double

Dim stkP As Double

Dim regex As Object

Dim celdaTituloF As Range

Dim celdaTituloG As Range

Dim sustitutosArray As Variant

Dim sustitutosDict As Object

Dim yaMostradoDict As Object

Dim sustituto As String

Dim encontrado As Boolean

Dim sumaFco As Double

Dim sumaBot As Double

Dim sumaBid As Double

Dim sumaTam As Double

Dim sumaDosif As Double

Dim sumaPomo As Double

Dim sumatapa As Double

Dim sumaTotal As Double

Dim wsMadre As Worksheet

'----------------------HOJA TERMINADOS----------------------'

**' Verificar existencia de la hoja "Terminados"**

On Error Resume Next

Set wsTerminados = ThisWorkbook.Sheets("Terminados")

On Error GoTo 0

If Not wsTerminados Is Nothing Then

**' Establecer el último rango de fila y columna a recorrer**

lastRow = wsTerminados.Cells(wsTerminados.Rows.count, 1).End(xlUp).Row

lastCol = wsTerminados.Cells(1, wsTerminados.Columns.count).End(xlToLeft).Column

**' Sumar valores en "Valor Total" según la "Denominación"**

For i = 2 To lastRow

descripcion = wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Denominación").Column).Value

If InStr(1, descripcion, "Fco", vbTextCompare) = 1 Then

sumaFco = sumaFco + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

ElseIf InStr(1, descripcion, "Bot", vbTextCompare) = 1 Then

sumaBot = sumaBot + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

ElseIf InStr(1, descripcion, "Bid", vbTextCompare) = 1 Then

sumaBid = sumaBid + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

ElseIf InStr(1, descripcion, "Tam", vbTextCompare) = 1 Then

sumaTam = sumaTam + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

ElseIf InStr(1, descripcion, "Dosif", vbTextCompare) = 1 Then

sumaDosif = sumaDosif + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

ElseIf InStr(1, descripcion, "Pomo", vbTextCompare) = 1 Then

sumaPomo = sumaPomo + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

ElseIf InStr(1, descripcion, "Tapa", vbTextCompare) = 1 Then

sumatapa = sumatapa + wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Valor Total").Column).Value

End If

Next i

**' Calcular la suma total**

sumaTotal = sumaFco + sumaBot + sumaBid + sumaTam + sumaDosif + sumaPomo + sumatapa

**' Colocar los resultados en la hoja "Madre"**

Set wsMadre = ThisWorkbook.Sheets("Madre")

wsMadre.Range("G6").Value = sumaTotal

wsMadre.Range("G7").Value = sumaFco

wsMadre.Range("G8").Value = sumaBot

wsMadre.Range("G9").Value = sumaBid

wsMadre.Range("G10").Value = sumaTam

wsMadre.Range("G11").Value = sumaDosif

wsMadre.Range("G12").Value = sumaPomo

wsMadre.Range("G13").Value = sumatapa

End If

**' Verificar existencia de la hoja "Terminados" y establecer referencia**

On Error Resume Next

Set wsTerminados = ThisWorkbook.Sheets("Terminados")

On Error GoTo 0

If Not wsTerminados Is Nothing Then

**' Establecer el rango de datos (asume que los datos están desde la columna A hasta la última columna y fila con datos)**

lastRow = wsTerminados.Cells(wsTerminados.Rows.count, 2).End(xlUp).Row

lastCol = wsTerminados.Cells(1, wsTerminados.Columns.count).End(xlToLeft).Column

**' Establecer el rango completo de las filas para ordenar (desde la columna A hasta la última columna con datos)**

Set rngData = wsTerminados.Range("A1:" & wsTerminados.Cells(1, lastCol).Address & lastRow)

**' Aplicar formato condicional de barra de datos a la columna B**

With wsTerminados.Range("B2:B" & lastRow).FormatConditions.AddDatabar

.MinPoint.Modify xlConditionValueLowestValue

.MaxPoint.Modify xlConditionValueHighestValue

End With

**' Aplicar formato de número tipo "miliar" a la columna B**

wsTerminados.Columns("B").NumberFormat = "#,##0"

wsTerminados.Columns("F").NumberFormat = "#,##0"

With wsTerminados

.Range("A1:F" & .Cells(.Rows.count, "A").End(xlUp).Row).Borders.LineStyle = xlContinuous

End With

**' Buscar el patrón "(#K)" en toda la hoja y compararlo con la columna D "Stk.(U)"**

Set regex = CreateObject("VBScript.RegExp")

regex.Pattern = "\((\d+)K\)"

regex.IgnoreCase = True

For i = 2 To lastRow **' Recorre las filas**

For j = 1 To lastCol **' Recorre las columnas**

descripcion = wsTerminados.Cells(i, j).Value

If regex.Test(descripcion) Then

numeroExtraido = CDbl(regex.Execute(descripcion)(0).SubMatches(0)) \* 1000

stkU = wsTerminados.Cells(i, wsTerminados.Rows(1).Find("Stk.(U)").Column).Value

**' Si el número extraído multiplicado por 1000 es mayor que el valor en "Stk.(U)", pintar la fila de rojo**

If numeroExtraido > stkU Then

' Identificar la última columna con datos en la fila actual

lastColData = wsTerminados.Cells(i, wsTerminados.Columns.count).End(xlToLeft).Column

**' Colorear de rojo hasta la última columna**

wsTerminados.Range(wsTerminados.Cells(i, 1), wsTerminados.Cells(i, lastColData)).Interior.Color = RGB(255, 100, 100)

**' Salir del bucle de columnas, ya que ya se ha pintado toda la fila**

Exit For

End If

End If

Next j

Next i

**' Ordenar los datos de mayor a menor según la columna A**

wsTerminados.Sort.SortFields.Clear

wsTerminados.Sort.SortFields.Add key:=wsTerminados.Range("A2:A" & lastRow), SortOn:=xlSortOnValues, Order:=xlAscending

With wsTerminados.Sort

.SetRange rngData **' Ordenar todas las columnas según la columna A**

.Header = xlYes **' Si la primera fila es un encabezado**

.Apply

End With

**' Ajustar automáticamente el tamaño de las columnas**

wsTerminados.Columns.AutoFit

wsTerminados.Columns("B").ColumnWidth = 35

wsTerminados.Columns("D").ColumnWidth = 20

wsTerminados.Columns("F").ColumnWidth = 20

wsTerminados.Range("A1:F1").Interior.Color = RGB(173, 216, 230) **' Celeste claro**

End If

Set wsMadre = ThisWorkbook.Sheets("Madre")

**' Colocar los campos en la hoja “Madre”**

wsMadre.Range("F5").Value = "PRODUCTOS"

wsMadre.Range("G5").Value = "TOTAL s/IVA"

wsMadre.Range("F7").Value = "Frascos:"

wsMadre.Range("F8").Value = "Botellas:"

wsMadre.Range("F9").Value = "Bidones:"

wsMadre.Range("F10").Value = "Tambores:"

wsMadre.Range("F11").Value = "Dosificadores:"

wsMadre.Range("F12").Value = "Pomos:"

wsMadre.Range("F13").Value = "Tapa:"

wsMadre.Range("F6").Value = "Total:"

wsMadre.Range("B5").Value = "COSTOS PREFORMAS:"

wsMadre.Range("D5").Value = "COSTOS GRUMOS:"

Dim rangoSumas As Range

Set rangoSumas = wsMadre.Range("G5:B5")

rangoSumas.Interior.Color = RGB(173, 216, 230)

**‘Formato miliar en las columnas de la hoja “Madre”**

wsMadre.Columns("C").NumberFormat = "#,##0.00000"

wsMadre.Columns("E").NumberFormat = "#,##0.00000"

wsMadre.Columns("G").NumberFormat = "#,##0.00000"

**' Crear diccionarios para almacenar las sumas por sustituto y para controlar la visualización**

Set sustitutosDict = CreateObject("Scripting.Dictionary")

Set yaMostradoDict = CreateObject("Scripting.Dictionary")

**' Array de sustitutos**

sustitutosArray = Array("(A)", "(B)", "(C)", "(D)", "(E)", "(F)", "(G)", "(H)", "(I)", "(J)", "(K)", "(L)", "(M)", "(N)", "(O)", "(P)", "(Q)", "(R)", "(S)", "(T)", "(U)", "(V)", "(W)", "(X)", "(Y)", "(Z)", \_

"(AA)", "(AB)", "(AC)", "(AD)", "(AE)", "(AF)", "(AG)", "(AH)", "(AI)", "(AJ)", "(AK)", "(AL)", "(AM)", "(AN)", "(AO)", "(AP)", "(AQ)", "(AR)", "(AS)", "(AT)", "(AU)", "(AV)", "(AW)", "(AX)", "(AY)", "(AZ)", "(BA)", "(BB)", "(BC)", "(BD)", "(BE)", "(BF)", "(BG)", "(BH)", "(BI)", "(BJ)", "(BK)", "(BL)", "(BM)", "(BN)", "(BO)", "(BP)", "(BQ)", "(BR)", "(BS)", "(BT)", "(BU)", "(BV)", "(BW)", "(BX)", "(BY)", "(BZ)", \_

"(CA)", "(CB)", "(CC)", "(CD)", "(CE)", "(CF)", "(CG)", "(CH)", "(CI)", "(CJ)", "(CK)", "(CL)", "(CM)", "(CN)", "(CO)", "(CP)", "(CQ)", "(CR)", "(CS)", "(CT)", "(CU)", "(CV)", "(CW)", "(CX)", "(CY)", "(CZ)", "(DA)", "(DB)", "(DC)", "(DD)", "(DE)", "(DF)", "(DG)", "(DH)", "(DI)", "(DJ)", "(DK)", "(DL)", "(DM)", "(DN)", "(DO)", "(DP)", "(DQ)", "(DR)", "(DS)", "(DT)", "(DU)", "(DV)", "(DW)", "(DX)", "(DY)", "(DZ)", \_

"(EA)", "(EB)", "(EC)", "(ED)", "(EE)", "(EF)", "(EG)", "(EH)", "(EI)", "(EJ)", "(EK)", "(EL)", "(EM)", "(EN)", "(EO)", "(EP)", "(EQ)", "(ER)", "(ES)", "(ET)", "(EU)", "(EV)", "(EW)", "(EX)", "(EY)", "(EZ)", \_

"(FA)", "(FB)", "(FC)", "(FD)", "(FE)", "(FF)", "(FG)", "(FH)", "(FI)", "(FJ)", "(FK)", "(FL)", "(FM)", "(FN)", "(FO)", "(FP)", "(FQ)", "(FR)", "(FS)", "(FT)", "(FU)", "(FV)", "(FW)", "(FX)", "(FY)", "(FZ)", \_

"(GA)", "(GB)", "(GC)", "(GD)", "(GE)", "(GF)", "(GG)", "(GH)", "(GI)", "(GJ)", "(GK)", "(GL)", "(GM)", "(GN)", "(GO)", "(GP)", "(GQ)", "(GR)", "(GS)", "(GT)", "(GU)", "(GV)", "(GW)", "(GX)", "(GY)", "(GZ)", \_

"(HA)", "(HB)", "(HC)", "(HD)", "(HE)", "(HF)", "(HG)", "(HH)", "(HI)", "(HJ)", "(HK)", "(HL)", "(HM)", "(HN)", "(HO)", "(HP)", "(HQ)", "(HR)", "(HS)", "(HT)", "(HU)", "(HV)", "(HW)", "(HX)", "(HY)", "(HZ)", \_

"(IA)", "(IB)", "(IC)", "(ID)", "(IE)", "(IF)", "(IG)", "(IH)", "(II)", "(IJ)", "(IK)", "(IL)", "(IM)", "(IN)", "(IO)", "(IP)", "(IQ)", "(IR)", "(IS)", "(IT)", "(IU)", "(IV)", "(IW)", "(IX)", "(IY)", "(IZ)", \_

"(JA)", "(JB)", "(JC)", "(JD)", "(JE)", "(JF)", "(JG)", "(JH)", "(JI)", "(JJ)", "(JK)", "(JL)", "(JM)", "(JN)", "(JO)", "(JP)", "(JQ)", "(JR)", "(JS)", "(JT)", "(JU)", "(JV)", "(JW)", "(JX)", "(JY)", "(JZ)", \_

"(KA)", "(KB)", "(KC)", "(KD)", "(KE)", "(KF)", "(KG)", "(KH)", "(KI)", "(KJ)", "(KK)", "(KL)", "(KM)", "(KN)", "(KO)", "(KP)", "(KQ)", "(KR)", "(KS)", "(KT)", "(KU)", "(KV)", "(KW)", "(KX)", "(KY)", "(KZ)", \_

"(LA)", "(LB)", "(LC)", "(LD)", "(LE)", "(LF)", "(LG)", "(LH)", "(LI)", "(LJ)", "(LK)", "(LL)", "(LM)", "(LN)", "(LO)", "(LP)", "(LQ)", "(LR)", "(LS)", "(LT)", "(LU)", "(LV)", "(LW)", "(LX)", "(LY)", "(LZ)", \_

"(MA)", "(MB)", "(MC)", "(MD)", "(ME)", "(MF)", "(MG)", "(MH)", "(MI)", "(MJ)", "(MK)", "(ML)", "(MM)", "(MN)", "(MO)", "(MP)", "(MQ)", "(MR)", "(MS)", "(MT)", "(MU)", "(MV)", "(MW)", "(MX)", "(MY)", "(MZ)", \_

"(NA)", "(NB)", "(NC)", "(ND)", "(NE)", "(NF)", "(NG)", "(NH)", "(NI)", "(NJ)", "(NK)", "(NL)", "(NM)", "(NN)", "(NO)", "(NP)", "(NQ)", "(NR)", "(NS)", "(NT)", "(NU)", "(NV)", "(NW)", "(NX)", "(NY)", "(NZ)", \_

"(OA)", "(OB)", "(OC)", "(OD)", "(OE)", "(OF)", "(OG)", "(OH)", "(OI)", "(OJ)", "(OK)", "(OL)", "(OM)", "(ON)", "(OO)", "(OP)", "(OQ)", "(OR)", "(OS)", "(OT)", "(OU)", "(OV)", "(OW)", "(OX)", "(OY)", "(OZ)", \_

"(PA)", "(PB)", "(PC)", "(PD)", "(PE)", "(PF)", "(PG)", "(PH)", "(PI)", "(PJ)", "(PK)", "(PL)", "(PM)", "(PN)", "(PO)", "(PP)", "(PQ)", "(PR)", "(PS)", "(PT)", "(PU)", "(PV)", "(PW)", "(PX)", "(PY)", "(PZ)", \_

"(QA)", "(QB)", "(QC)", "(QD)", "(QE)", "(QF)", "(QG)", "(QH)", "(QI)", "(QJ)", "(QK)", "(QL)", "(QM)", "(QN)", "(QO)", "(QP)", "(QQ)", "(QR)", "(QS)", "(QT)", "(QU)", "(QV)", "(QW)", "(QX)", "(QY)", "(QZ)", \_

"(RA)", "(RB)", "(RC)", "(RD)", "(RE)", "(RF)", "(RG)", "(RH)", "(RI)", "(RJ)", "(RK)", "(RL)", "(RM)", "(RN)", "(RO)", "(RP)", "(RQ)", "(RR)", "(RS)", "(RT)", "(RU)", "(RV)", "(RW)", "(RX)", "(RY)", "(RZ)", \_

"(SA)", "(SB)", "(SC)", "(SD)", "(SE)", "(SF)", "(SG)", "(SH)", "(SI)", "(SJ)", "(SK)", "(SL)", "(SM)", "(SN)", "(SO)", "(SP)", "(SQ)", "(SR)", "(SS)", "(ST)", "(SU)", "(SV)", "(SW)", "(SX)", "(SY)", "(SZ)", \_

"(TA)", "(TB)", "(TC)", "(TD)", "(TE)", "(TF)", "(TG)", "(TH)", "(TI)", "(TJ)", "(TK)", "(TL)", "(TM)", "(TN)", "(TO)", "(TP)", "(TQ)", "(TR)", "(TS)", "(TT)", "(TU)", "(TV)", "(TW)", "(TX)", "(TY)", "(TZ)", \_

"(UA)", "(UB)", "(UC)", "(UD)", "(UE)", "(UF)", "(UG)", "(UH)", "(UI)", "(UJ)", "(UK)", "(UL)", "(UM)", "(UN)", "(UO)", "(UP)", "(UQ)", "(UR)", "(US)", "(UT)", "(UU)", "(UV)", "(UW)", "(UX)", "(UY)", "(UZ)", \_

"(VA)", "(VB)", "(VC)", "(VD)", "(VE)", "(VF)", "(VG)", "(VH)", "(VI)", "(VJ)", "(VK)", "(VL)", "(VM)", "(VN)", "(VO)", "(VP)", "(VQ)", "(VR)", "(VS)", "(VT)", "(VU)", "(VV)", "(VW)", "(VX)", "(VY)", "(VZ)", \_

"(WA)", "(WB)", "(WC)", "(WD)", "(WE)", "(WF)", "(WG)", "(WH)", "(WI)", "(WJ)", "(WK)", "(WL)", "(WM)", "(WN)", "(WO)", "(WP)", "(WQ)", "(WR)", "(WS)", "(WT)", "(WU)", "(WV)", "(WW)", "(WX)", "(WY)", "(WZ)", \_

"(XA)", "(XB)", "(XC)", "(XD)", "(XE)", "(XF)", "(XG)", "(XH)", "(XI)", "(XJ)", "(XK)", "(XL)", "(XM)", "(XN)", "(XO)", "(XP)", "(XQ)", "(XR)", "(XS)", "(XT)", "(XU)", "(XV)", "(XW)", "(XX)", "(XY)", "(XZ)", \_

"(YA)", "(YB)", "(YC)", "(YD)", "(YE)", "(YF)", "(YG)", "(YH)", "(YI)", "(YJ)", "(YK)", "(YL)", "(YM)", "(YN)", "(YO)", "(YP)", "(YQ)", "(YR)", "(YS)", "(YT)", "(YU)", "(YV)", "(YW)", "(YX)", "(YY)", "(YZ)", \_

"(ZA)", "(ZB)", "(ZC)", "(ZD)", "(ZE)", "(ZF)", "(ZG)", "(ZH)", "(ZI)", "(ZJ)", "(ZK)", "(ZL)", "(ZM)", "(ZN)", "(ZO)", "(ZP)", "(ZQ)", "(ZR)", "(ZS)", "(ZT)", "(ZU)", "(ZV)", "(ZW)", "(ZX)", "(ZY)", "(ZZ)")

'----------------------HOJA GRUMOS----------------------'

**' Verificar existencia de la hoja "GRUMOS" y establecer referencia**

On Error Resume Next

Set wsGrumos = ThisWorkbook.Sheets("GRUMOS")

On Error GoTo 0

If Not wsGrumos Is Nothing Then

**' Establecer el rango de datos (asume que los datos están desde la columna A hasta la última columna y fila con datos)**

lastRow = wsGrumos.Cells(wsGrumos.Rows.count, 2).End(xlUp).Row

lastCol = wsGrumos.Cells(1, wsGrumos.Columns.count).End(xlToLeft).Column

**' Establecer el rango completo de las filas para ordenar (desde la columna A hasta la última columna con datos)**

Set rngData = wsGrumos.Range("A1:" & wsGrumos.Cells(1, lastCol).Address & lastRow)

**' Aplicar formato condicional de barra de datos a la columna B**

With wsGrumos.Range("B2:B" & lastRow).FormatConditions.AddDatabar

.MinPoint.Modify xlConditionValueLowestValue

.MaxPoint.Modify xlConditionValueHighestValue

End With

**' Ordenar los datos de mayor a menor según la columna B**

wsGrumos.Sort.SortFields.Clear

wsGrumos.Sort.SortFields.Add key:=wsGrumos.Range("B2:B" & lastRow), SortOn:=xlSortOnValues, Order:=xlDescending

With wsGrumos.Sort

.SetRange rngData **' Ordenar todas las columnas según la columna B**

.Header = xlYes **' Si la primera fila es un encabezado**

.Apply

End With

**' Aplicar formato de número tipo "miliar" a la columna B, F, G**

wsGrumos.Columns("B").NumberFormat = "#,##0"

wsGrumos.Columns("F").NumberFormat = "#,##0.000"

wsGrumos.Columns("G").NumberFormat = "#,##0.00"

**' Buscar el patrón "(#K)" en toda la hoja y compararlo con la columna D "Stk.(U)"**

Set regex = CreateObject("VBScript.RegExp")

regex.Pattern = "\((\d+)K\)"

regex.IgnoreCase = True

For i = 2 To lastRow **' Recorre las filas**

For j = 1 To lastCol **' Recorre las columnas**

descripcion = wsGrumos.Cells(i, j).Value

If regex.Test(descripcion) Then

numeroExtraido = CDbl(regex.Execute(descripcion)(0).SubMatches(0)) \* 1000

stkU = wsGrumos.Cells(i, wsGrumos.Rows(1).Find("Suma Stk.(P)").Column).Value

**' Si el número extraído multiplicado por 1000 es mayor que el valor en "Stk.(U)", pintar la fila de rojo**

If numeroExtraido > stkU Then

**' Identificar la última columna con datos en la fila actual**

lastColData = wsGrumos.Cells(i, wsGrumos.Columns.count).End(xlToLeft).Column

**' Colorear de rojo hasta la última columna**

wsGrumos.Range(wsGrumos.Cells(i, 1), wsGrumos.Cells(i, lastColData)).Interior.Color = RGB(255, 100, 100)

**' Salir del bucle de columnas, ya que ya se ha pintado toda la fila**

Exit For

End If

End If

Next j

Next i

End If

On Error Resume Next

Set wsGrumos = ThisWorkbook.Sheets("GRUMOS")

On Error GoTo 0

If Not wsGrumos Is Nothing Then

**‘Definir más variables**

Dim hojaGrumos As Worksheet

Dim filaUltima As Long

Dim fila As Long

Dim arraySustitutos As Variant

Dim sustitutivo As Variant

Dim existe As Boolean

arraySustitutos = Array("(A)", "(B)", "(C)", "(D)", "(E)", "(F)", "(G)", "(H)", "(I)", "(J)", "(K)", "(L)", "(M)", "(N)", "(O)", "(P)", "(Q)", "(R)", "(S)", "(T)", "(U)", "(V)", "(W)", "(X)", "(Y)", "(Z)", \_

"(AA)", "(AB)", "(AC)", "(AD)", "(AE)", "(AF)", "(AG)", "(AH)", "(AI)", "(AJ)", "(AK)", "(AL)", "(AM)", "(AN)", "(AO)", "(AP)", "(AQ)", "(AR)", "(AS)", "(AT)", "(AU)", "(AV)", "(AW)", "(AX)", "(AY)", "(AZ)", "(BA)", "(BB)", "(BC)", "(BD)", "(BE)", "(BF)", "(BG)", "(BH)", "(BI)", "(BJ)", "(BK)", "(BL)", "(BM)", "(BN)", "(BO)", "(BP)", "(BQ)", "(BR)", "(BS)", "(BT)", "(BU)", "(BV)", "(BW)", "(BX)", "(BY)", "(BZ)", \_

"(CA)", "(CB)", "(CC)", "(CD)", "(CE)", "(CF)", "(CG)", "(CH)", "(CI)", "(CJ)", "(CK)", "(CL)", "(CM)", "(CN)", "(CO)", "(CP)", "(CQ)", "(CR)", "(CS)", "(CT)", "(CU)", "(CV)", "(CW)", "(CX)", "(CY)", "(CZ)", "(DA)", "(DB)", "(DC)", "(DD)", "(DE)", "(DF)", "(DG)", "(DH)", "(DI)", "(DJ)", "(DK)", "(DL)", "(DM)", "(DN)", "(DO)", "(DP)", "(DQ)", "(DR)", "(DS)", "(DT)", "(DU)", "(DV)", "(DW)", "(DX)", "(DY)", "(DZ)", \_

"(EA)", "(EB)", "(EC)", "(ED)", "(EE)", "(EF)", "(EG)", "(EH)", "(EI)", "(EJ)", "(EK)", "(EL)", "(EM)", "(EN)", "(EO)", "(EP)", "(EQ)", "(ER)", "(ES)", "(ET)", "(EU)", "(EV)", "(EW)", "(EX)", "(EY)", "(EZ)", \_

"(FA)", "(FB)", "(FC)", "(FD)", "(FE)", "(FF)", "(FG)", "(FH)", "(FI)", "(FJ)", "(FK)", "(FL)", "(FM)", "(FN)", "(FO)", "(FP)", "(FQ)", "(FR)", "(FS)", "(FT)", "(FU)", "(FV)", "(FW)", "(FX)", "(FY)", "(FZ)", \_

"(GA)", "(GB)", "(GC)", "(GD)", "(GE)", "(GF)", "(GG)", "(GH)", "(GI)", "(GJ)", "(GK)", "(GL)", "(GM)", "(GN)", "(GO)", "(GP)", "(GQ)", "(GR)", "(GS)", "(GT)", "(GU)", "(GV)", "(GW)", "(GX)", "(GY)", "(GZ)", \_

"(HA)", "(HB)", "(HC)", "(HD)", "(HE)", "(HF)", "(HG)", "(HH)", "(HI)", "(HJ)", "(HK)", "(HL)", "(HM)", "(HN)", "(HO)", "(HP)", "(HQ)", "(HR)", "(HS)", "(HT)", "(HU)", "(HV)", "(HW)", "(HX)", "(HY)", "(HZ)", \_

"(IA)", "(IB)", "(IC)", "(ID)", "(IE)", "(IF)", "(IG)", "(IH)", "(II)", "(IJ)", "(IK)", "(IL)", "(IM)", "(IN)", "(IO)", "(IP)", "(IQ)", "(IR)", "(IS)", "(IT)", "(IU)", "(IV)", "(IW)", "(IX)", "(IY)", "(IZ)", \_

"(JA)", "(JB)", "(JC)", "(JD)", "(JE)", "(JF)", "(JG)", "(JH)", "(JI)", "(JJ)", "(JK)", "(JL)", "(JM)", "(JN)", "(JO)", "(JP)", "(JQ)", "(JR)", "(JS)", "(JT)", "(JU)", "(JV)", "(JW)", "(JX)", "(JY)", "(JZ)", \_

"(KA)", "(KB)", "(KC)", "(KD)", "(KE)", "(KF)", "(KG)", "(KH)", "(KI)", "(KJ)", "(KK)", "(KL)", "(KM)", "(KN)", "(KO)", "(KP)", "(KQ)", "(KR)", "(KS)", "(KT)", "(KU)", "(KV)", "(KW)", "(KX)", "(KY)", "(KZ)", \_

"(LA)", "(LB)", "(LC)", "(LD)", "(LE)", "(LF)", "(LG)", "(LH)", "(LI)", "(LJ)", "(LK)", "(LL)", "(LM)", "(LN)", "(LO)", "(LP)", "(LQ)", "(LR)", "(LS)", "(LT)", "(LU)", "(LV)", "(LW)", "(LX)", "(LY)", "(LZ)", \_

"(MA)", "(MB)", "(MC)", "(MD)", "(ME)", "(MF)", "(MG)", "(MH)", "(MI)", "(MJ)", "(MK)", "(ML)", "(MM)", "(MN)", "(MO)", "(MP)", "(MQ)", "(MR)", "(MS)", "(MT)", "(MU)", "(MV)", "(MW)", "(MX)", "(MY)", "(MZ)", \_

"(NA)", "(NB)", "(NC)", "(ND)", "(NE)", "(NF)", "(NG)", "(NH)", "(NI)", "(NJ)", "(NK)", "(NL)", "(NM)", "(NN)", "(NO)", "(NP)", "(NQ)", "(NR)", "(NS)", "(NT)", "(NU)", "(NV)", "(NW)", "(NX)", "(NY)", "(NZ)", \_

"(OA)", "(OB)", "(OC)", "(OD)", "(OE)", "(OF)", "(OG)", "(OH)", "(OI)", "(OJ)", "(OK)", "(OL)", "(OM)", "(ON)", "(OO)", "(OP)", "(OQ)", "(OR)", "(OS)", "(OT)", "(OU)", "(OV)", "(OW)", "(OX)", "(OY)", "(OZ)", \_

"(PA)", "(PB)", "(PC)", "(PD)", "(PE)", "(PF)", "(PG)", "(PH)", "(PI)", "(PJ)", "(PK)", "(PL)", "(PM)", "(PN)", "(PO)", "(PP)", "(PQ)", "(PR)", "(PS)", "(PT)", "(PU)", "(PV)", "(PW)", "(PX)", "(PY)", "(PZ)", \_

"(QA)", "(QB)", "(QC)", "(QD)", "(QE)", "(QF)", "(QG)", "(QH)", "(QI)", "(QJ)", "(QK)", "(QL)", "(QM)", "(QN)", "(QO)", "(QP)", "(QQ)", "(QR)", "(QS)", "(QT)", "(QU)", "(QV)", "(QW)", "(QX)", "(QY)", "(QZ)", \_

"(RA)", "(RB)", "(RC)", "(RD)", "(RE)", "(RF)", "(RG)", "(RH)", "(RI)", "(RJ)", "(RK)", "(RL)", "(RM)", "(RN)", "(RO)", "(RP)", "(RQ)", "(RR)", "(RS)", "(RT)", "(RU)", "(RV)", "(RW)", "(RX)", "(RY)", "(RZ)", \_

"(SA)", "(SB)", "(SC)", "(SD)", "(SE)", "(SF)", "(SG)", "(SH)", "(SI)", "(SJ)", "(SK)", "(SL)", "(SM)", "(SN)", "(SO)", "(SP)", "(SQ)", "(SR)", "(SS)", "(ST)", "(SU)", "(SV)", "(SW)", "(SX)", "(SY)", "(SZ)", \_

"(TA)", "(TB)", "(TC)", "(TD)", "(TE)", "(TF)", "(TG)", "(TH)", "(TI)", "(TJ)", "(TK)", "(TL)", "(TM)", "(TN)", "(TO)", "(TP)", "(TQ)", "(TR)", "(TS)", "(TT)", "(TU)", "(TV)", "(TW)", "(TX)", "(TY)", "(TZ)", \_

"(UA)", "(UB)", "(UC)", "(UD)", "(UE)", "(UF)", "(UG)", "(UH)", "(UI)", "(UJ)", "(UK)", "(UL)", "(UM)", "(UN)", "(UO)", "(UP)", "(UQ)", "(UR)", "(US)", "(UT)", "(UU)", "(UV)", "(UW)", "(UX)", "(UY)", "(UZ)", \_

"(VA)", "(VB)", "(VC)", "(VD)", "(VE)", "(VF)", "(VG)", "(VH)", "(VI)", "(VJ)", "(VK)", "(VL)", "(VM)", "(VN)", "(VO)", "(VP)", "(VQ)", "(VR)", "(VS)", "(VT)", "(VU)", "(VV)", "(VW)", "(VX)", "(VY)", "(VZ)", \_

"(WA)", "(WB)", "(WC)", "(WD)", "(WE)", "(WF)", "(WG)", "(WH)", "(WI)", "(WJ)", "(WK)", "(WL)", "(WM)", "(WN)", "(WO)", "(WP)", "(WQ)", "(WR)", "(WS)", "(WT)", "(WU)", "(WV)", "(WW)", "(WX)", "(WY)", "(WZ)", \_

"(XA)", "(XB)", "(XC)", "(XD)", "(XE)", "(XF)", "(XG)", "(XH)", "(XI)", "(XJ)", "(XK)", "(XL)", "(XM)", "(XN)", "(XO)", "(XP)", "(XQ)", "(XR)", "(XS)", "(XT)", "(XU)", "(XV)", "(XW)", "(XX)", "(XY)", "(XZ)", \_

"(YA)", "(YB)", "(YC)", "(YD)", "(YE)", "(YF)", "(YG)", "(YH)", "(YI)", "(YJ)", "(YK)", "(YL)", "(YM)", "(YN)", "(YO)", "(YP)", "(YQ)", "(YR)", "(YS)", "(YT)", "(YU)", "(YV)", "(YW)", "(YX)", "(YY)", "(YZ)", \_

"(ZA)", "(ZB)", "(ZC)", "(ZD)", "(ZE)", "(ZF)", "(ZG)", "(ZH)", "(ZI)", "(ZJ)", "(ZK)", "(ZL)", "(ZM)", "(ZN)", "(ZO)", "(ZP)", "(ZQ)", "(ZR)", "(ZS)", "(ZT)", "(ZU)", "(ZV)", "(ZW)", "(ZX)", "(ZY)", "(ZZ)")

**' Establecer la hoja de trabajo**

Set hojaGrumos = ThisWorkbook.Sheets("GRUMOS")

**' Obtener la última fila con datos en la columna D**

filaUltima = hojaGrumos.Cells(hojaGrumos.Rows.count, "D").End(xlUp).Row

**' Comprobar si hay datos en la columna D**

If filaUltima < 2 Then

Exit Sub

End If

**' Recorrer las celdas de la columna D**

For fila = 2 To filaUltima **' Asumiendo que la fila 1 tiene encabezados**

existe = False

**' Verificar si el valor se encuentra en el array**

For Each sustitutivo In arraySustitutos

If hojaGrumos.Cells(fila, "D").Value = sustitutivo Then

existe = True

Exit For

End If

Next sustitutivo

**' Si no se encontró, cambiar el valor**

If Not existe Then

hojaGrumos.Cells(fila, "D").Value = ""

End If

Next fila

End If

Dim wsResultados As Worksheet

Dim hojaMadre As Worksheet

On Error Resume Next

Set wsResultados = ThisWorkbook.Sheets("GRUMOS")

Set hojaMadre = ThisWorkbook.Worksheets("Madre")

On Error GoTo 0

If Not wsResultados Is Nothing Then

Dim filaFinal As Long

filaFinal = wsResultados.Cells(wsResultados.Rows.count, 1).End(xlUp).Row

**' Colocar el encabezado en la columna G, fila 1**

wsResultados.Cells(1, 7).Value = "Costos total s/IVA"

Dim filaContadora As Long

For filaContadora = 2 To filaFinal

**' Multiplicar los valores de las filas "Suma Stk.(P)" (Columna B) y "Costo s/IVA" (Columna F)**

wsResultados.Cells(filaContadora, 7).Value = wsResultados.Cells(filaContadora, 2).Value \* wsResultados.Cells(filaContadora, 6).Value

Next filaContadora

**' Calcular la suma de toda la columna "Costos total s/IVA"**

Dim sumaCostosTotal As Double

sumaCostosTotal = Application.WorksheetFunction.Sum(wsResultados.Range("G2:G" & filaFinal))

**' Imprimir la suma en la hoja "Madre" (por ejemplo, en la celda F6)**

hojaMadre.Cells(6, 5).Value = sumaCostosTotal

hojaMadre.Cells(6, 4).Value = "Total Costos s/IVA:"

**‘Declarar más variables**

Dim ultimaFila As Long

Dim rangoCosto As Range

Dim rangoStock As Range

Dim rangoTermino As Range

Dim celdaActual As Range

Dim listaTerminos As Variant

Dim terminoActual As Variant

Dim resultadoMultiplicacion As Double

Dim filaResultado As Long

**' Asignar las hojas a las variables**

Set hojaGrumos = ThisWorkbook.Worksheets("GRUMOS")

Set hojaMadre = ThisWorkbook.Worksheets("Madre")

**' Definir el array con los términos a buscar**

listaTerminos = Array("(A)", "(B)", "(C)", "(D)", "(E)", "(F)", "(G)", "(H)", "(I)", "(J)", "(K)", "(L)", "(M)", "(N)", "(O)", "(P)", "(Q)", "(R)", "(S)", "(T)", "(U)", "(V)", "(W)", "(X)", "(Y)", "(Z)", \_

"(AA)", "(AB)", "(AC)", "(AD)", "(AE)", "(AF)", "(AG)", "(AH)", "(AI)", "(AJ)", "(AK)", "(AL)", "(AM)", "(AN)", "(AO)", "(AP)", "(AQ)", "(AR)", "(AS)", "(AT)", "(AU)", "(AV)", "(AW)", "(AX)", "(AY)", "(AZ)", "(BA)", "(BB)", "(BC)", "(BD)", "(BE)", "(BF)", "(BG)", "(BH)", "(BI)", "(BJ)", "(BK)", "(BL)", "(BM)", "(BN)", "(BO)", "(BP)", "(BQ)", "(BR)", "(BS)", "(BT)", "(BU)", "(BV)", "(BW)", "(BX)", "(BY)", "(BZ)", \_

"(CA)", "(CB)", "(CC)", "(CD)", "(CE)", "(CF)", "(CG)", "(CH)", "(CI)", "(CJ)", "(CK)", "(CL)", "(CM)", "(CN)", "(CO)", "(CP)", "(CQ)", "(CR)", "(CS)", "(CT)", "(CU)", "(CV)", "(CW)", "(CX)", "(CY)", "(CZ)", "(DA)", "(DB)", "(DC)", "(DD)", "(DE)", "(DF)", "(DG)", "(DH)", "(DI)", "(DJ)", "(DK)", "(DL)", "(DM)", "(DN)", "(DO)", "(DP)", "(DQ)", "(DR)", "(DS)", "(DT)", "(DU)", "(DV)", "(DW)", "(DX)", "(DY)", "(DZ)", \_

"(EA)", "(EB)", "(EC)", "(ED)", "(EE)", "(EF)", "(EG)", "(EH)", "(EI)", "(EJ)", "(EK)", "(EL)", "(EM)", "(EN)", "(EO)", "(EP)", "(EQ)", "(ER)", "(ES)", "(ET)", "(EU)", "(EV)", "(EW)", "(EX)", "(EY)", "(EZ)", \_

"(FA)", "(FB)", "(FC)", "(FD)", "(FE)", "(FF)", "(FG)", "(FH)", "(FI)", "(FJ)", "(FK)", "(FL)", "(FM)", "(FN)", "(FO)", "(FP)", "(FQ)", "(FR)", "(FS)", "(FT)", "(FU)", "(FV)", "(FW)", "(FX)", "(FY)", "(FZ)", \_

"(GA)", "(GB)", "(GC)", "(GD)", "(GE)", "(GF)", "(GG)", "(GH)", "(GI)", "(GJ)", "(GK)", "(GL)", "(GM)", "(GN)", "(GO)", "(GP)", "(GQ)", "(GR)", "(GS)", "(GT)", "(GU)", "(GV)", "(GW)", "(GX)", "(GY)", "(GZ)", \_

"(HA)", "(HB)", "(HC)", "(HD)", "(HE)", "(HF)", "(HG)", "(HH)", "(HI)", "(HJ)", "(HK)", "(HL)", "(HM)", "(HN)", "(HO)", "(HP)", "(HQ)", "(HR)", "(HS)", "(HT)", "(HU)", "(HV)", "(HW)", "(HX)", "(HY)", "(HZ)", \_

"(IA)", "(IB)", "(IC)", "(ID)", "(IE)", "(IF)", "(IG)", "(IH)", "(II)", "(IJ)", "(IK)", "(IL)", "(IM)", "(IN)", "(IO)", "(IP)", "(IQ)", "(IR)", "(IS)", "(IT)", "(IU)", "(IV)", "(IW)", "(IX)", "(IY)", "(IZ)", \_

"(JA)", "(JB)", "(JC)", "(JD)", "(JE)", "(JF)", "(JG)", "(JH)", "(JI)", "(JJ)", "(JK)", "(JL)", "(JM)", "(JN)", "(JO)", "(JP)", "(JQ)", "(JR)", "(JS)", "(JT)", "(JU)", "(JV)", "(JW)", "(JX)", "(JY)", "(JZ)", \_

"(KA)", "(KB)", "(KC)", "(KD)", "(KE)", "(KF)", "(KG)", "(KH)", "(KI)", "(KJ)", "(KK)", "(KL)", "(KM)", "(KN)", "(KO)", "(KP)", "(KQ)", "(KR)", "(KS)", "(KT)", "(KU)", "(KV)", "(KW)", "(KX)", "(KY)", "(KZ)", \_

"(LA)", "(LB)", "(LC)", "(LD)", "(LE)", "(LF)", "(LG)", "(LH)", "(LI)", "(LJ)", "(LK)", "(LL)", "(LM)", "(LN)", "(LO)", "(LP)", "(LQ)", "(LR)", "(LS)", "(LT)", "(LU)", "(LV)", "(LW)", "(LX)", "(LY)", "(LZ)", \_

"(MA)", "(MB)", "(MC)", "(MD)", "(ME)", "(MF)", "(MG)", "(MH)", "(MI)", "(MJ)", "(MK)", "(ML)", "(MM)", "(MN)", "(MO)", "(MP)", "(MQ)", "(MR)", "(MS)", "(MT)", "(MU)", "(MV)", "(MW)", "(MX)", "(MY)", "(MZ)", \_

"(NA)", "(NB)", "(NC)", "(ND)", "(NE)", "(NF)", "(NG)", "(NH)", "(NI)", "(NJ)", "(NK)", "(NL)", "(NM)", "(NN)", "(NO)", "(NP)", "(NQ)", "(NR)", "(NS)", "(NT)", "(NU)", "(NV)", "(NW)", "(NX)", "(NY)", "(NZ)", \_

"(OA)", "(OB)", "(OC)", "(OD)", "(OE)", "(OF)", "(OG)", "(OH)", "(OI)", "(OJ)", "(OK)", "(OL)", "(OM)", "(ON)", "(OO)", "(OP)", "(OQ)", "(OR)", "(OS)", "(OT)", "(OU)", "(OV)", "(OW)", "(OX)", "(OY)", "(OZ)", \_

"(PA)", "(PB)", "(PC)", "(PD)", "(PE)", "(PF)", "(PG)", "(PH)", "(PI)", "(PJ)", "(PK)", "(PL)", "(PM)", "(PN)", "(PO)", "(PP)", "(PQ)", "(PR)", "(PS)", "(PT)", "(PU)", "(PV)", "(PW)", "(PX)", "(PY)", "(PZ)", \_

"(QA)", "(QB)", "(QC)", "(QD)", "(QE)", "(QF)", "(QG)", "(QH)", "(QI)", "(QJ)", "(QK)", "(QL)", "(QM)", "(QN)", "(QO)", "(QP)", "(QQ)", "(QR)", "(QS)", "(QT)", "(QU)", "(QV)", "(QW)", "(QX)", "(QY)", "(QZ)", \_

"(RA)", "(RB)", "(RC)", "(RD)", "(RE)", "(RF)", "(RG)", "(RH)", "(RI)", "(RJ)", "(RK)", "(RL)", "(RM)", "(RN)", "(RO)", "(RP)", "(RQ)", "(RR)", "(RS)", "(RT)", "(RU)", "(RV)", "(RW)", "(RX)", "(RY)", "(RZ)", \_

"(SA)", "(SB)", "(SC)", "(SD)", "(SE)", "(SF)", "(SG)", "(SH)", "(SI)", "(SJ)", "(SK)", "(SL)", "(SM)", "(SN)", "(SO)", "(SP)", "(SQ)", "(SR)", "(SS)", "(ST)", "(SU)", "(SV)", "(SW)", "(SX)", "(SY)", "(SZ)", \_

"(TA)", "(TB)", "(TC)", "(TD)", "(TE)", "(TF)", "(TG)", "(TH)", "(TI)", "(TJ)", "(TK)", "(TL)", "(TM)", "(TN)", "(TO)", "(TP)", "(TQ)", "(TR)", "(TS)", "(TT)", "(TU)", "(TV)", "(TW)", "(TX)", "(TY)", "(TZ)", \_

"(UA)", "(UB)", "(UC)", "(UD)", "(UE)", "(UF)", "(UG)", "(UH)", "(UI)", "(UJ)", "(UK)", "(UL)", "(UM)", "(UN)", "(UO)", "(UP)", "(UQ)", "(UR)", "(US)", "(UT)", "(UU)", "(UV)", "(UW)", "(UX)", "(UY)", "(UZ)", \_

"(VA)", "(VB)", "(VC)", "(VD)", "(VE)", "(VF)", "(VG)", "(VH)", "(VI)", "(VJ)", "(VK)", "(VL)", "(VM)", "(VN)", "(VO)", "(VP)", "(VQ)", "(VR)", "(VS)", "(VT)", "(VU)", "(VV)", "(VW)", "(VX)", "(VY)", "(VZ)", \_

"(WA)", "(WB)", "(WC)", "(WD)", "(WE)", "(WF)", "(WG)", "(WH)", "(WI)", "(WJ)", "(WK)", "(WL)", "(WM)", "(WN)", "(WO)", "(WP)", "(WQ)", "(WR)", "(WS)", "(WT)", "(WU)", "(WV)", "(WW)", "(WX)", "(WY)", "(WZ)", \_

"(XA)", "(XB)", "(XC)", "(XD)", "(XE)", "(XF)", "(XG)", "(XH)", "(XI)", "(XJ)", "(XK)", "(XL)", "(XM)", "(XN)", "(XO)", "(XP)", "(XQ)", "(XR)", "(XS)", "(XT)", "(XU)", "(XV)", "(XW)", "(XX)", "(XY)", "(XZ)", \_

"(YA)", "(YB)", "(YC)", "(YD)", "(YE)", "(YF)", "(YG)", "(YH)", "(YI)", "(YJ)", "(YK)", "(YL)", "(YM)", "(YN)", "(YO)", "(YP)", "(YQ)", "(YR)", "(YS)", "(YT)", "(YU)", "(YV)", "(YW)", "(YX)", "(YY)", "(YZ)", \_

"(ZA)", "(ZB)", "(ZC)", "(ZD)", "(ZE)", "(ZF)", "(ZG)", "(ZH)", "(ZI)", "(ZJ)", "(ZK)", "(ZL)", "(ZM)", "(ZN)", "(ZO)", "(ZP)", "(ZQ)", "(ZR)", "(ZS)", "(ZT)", "(ZU)", "(ZV)", "(ZW)", "(ZX)", "(ZY)", "(ZZ)")

**' Buscar la columna con el encabezado "Costo s/IVA"**

Set rangoCosto = Nothing

For Each celdaActual In hojaGrumos.Rows(1).Cells

If celdaActual.Value = "Costo s/IVA" Then

Set rangoCosto = hojaGrumos.Range(celdaActual.Offset(1, 0), hojaGrumos.Cells(hojaGrumos.Rows.count, celdaActual.Column).End(xlUp))

Exit For

End If

Next celdaActual

' Buscar la columna con el encabezado "Suma Stk.(P)"

Set rangoStock = Nothing

For Each celdaActual In hojaGrumos.Rows(1).Cells

If celdaActual.Value = "Suma Stk.(P)" Then

Set rangoStock = hojaGrumos.Range(celdaActual.Offset(1, 0), hojaGrumos.Cells(hojaGrumos.Rows.count, celdaActual.Column).End(xlUp))

Exit For

End If

Next celdaActual

**' Buscar la columna con el encabezado correspondiente a los términos**

Set rangoTermino = Nothing

For Each celdaActual In hojaGrumos.Rows(1).Cells

If celdaActual.Value = "Sustituto" Then **' Reemplazar "Termino" con el nombre del encabezado de los términos**

Set rangoTermino = hojaGrumos.Range(celdaActual.Offset(1, 0), hojaGrumos.Cells(hojaGrumos.Rows.count, celdaActual.Column).End(xlUp))

Exit For

End If

Next celdaActual

**' Verificar si se encontraron las tres columnas**

If Not rangoCosto Is Nothing And Not rangoStock Is Nothing And Not rangoTermino Is Nothing Then

filaResultado = 7

**' Multiplicar los valores de "Costo s/IVA" y "Suma Stk.(P)" para cada término y obtener el resultado**

For Each terminoActual In listaTerminos

resultadoMultiplicacion = 0

For Each celdaActual In rangoTermino

If celdaActual.Value = terminoActual Then

resultadoMultiplicacion = resultadoMultiplicacion + (celdaActual.Offset(0, rangoStock.Column - rangoTermino.Column).Value \* celdaActual.Offset(0, rangoCosto.Column - rangoTermino.Column).Value)

End If

Next celdaActual

**' Solo imprimir si el resultado es mayor a cero**

If resultadoMultiplicacion > 0 Then

hojaMadre.Cells(filaResultado, 5).Value = resultadoMultiplicacion

hojaMadre.Cells(filaResultado, 4).Value = terminoActual **' Imprimir el término correspondiente para mayor claridad**

filaResultado = filaResultado + 1

End If

Next terminoActual

Else

End If

**‘Asigna tamaño de las columnas en la hoja “Grumos”**

wsGrumos.Columns.AutoFit

wsGrumos.Columns("B").ColumnWidth = 35

**' Formato de tabla**

With wsGrumos

.Range("A1:G" & .Cells(.Rows.count, "A").End(xlUp).Row).Borders.LineStyle = xlContinuous

End With

End If

'----------------------HOJA PREFORMAS----------------------'

**' Verificar existencia de la hoja "Preformas"**

On Error Resume Next

Set wsPreformas = ThisWorkbook.Sheets("Preformas")

On Error GoTo 0

If Not wsPreformas Is Nothing Then

**' Establecer el último rango de fila y columna a recorrer**

lastRow = wsPreformas.Cells(wsPreformas.Rows.count, 1).End(xlUp).Row

lastCol = wsPreformas.Cells(1, wsPreformas.Columns.count).End(xlToLeft).Column

**' Aplicar formato condicional a la columna B**

Set rngData = wsPreformas.Range("B2:B" & lastRow)

With rngData.FormatConditions.AddDatabar

.MinPoint.Modify xlConditionValueLowestValue

.MaxPoint.Modify xlConditionValueHighestValue

End With

Set rngData = wsPreformas.Range("G2:G" & lastRow)

With rngData.FormatConditions.AddDatabar

.MinPoint.Modify xlConditionValueLowestValue

.MaxPoint.Modify xlConditionValueHighestValue

End With

**' Aplicar formato de número tipo "miliar" a las columnas**

wsPreformas.Columns("B").NumberFormat = "#,##0"

wsPreformas.Columns("G").NumberFormat = "#,##0"

wsPreformas.Columns("H").NumberFormat = "#,##0.00000"

wsPreformas.Columns("I").NumberFormat = "#,##0.00000"

**' Buscar el patrón "(#K)" en toda la hoja y compararlo con la columna D "Stk.(U)"**

Set regex = CreateObject("VBScript.RegExp")

regex.Pattern = "\((\d+)K\)"

regex.IgnoreCase = True

For i = 2 To lastRow **' Recorre las filas**

For j = 1 To lastCol **' Recorre las columnas**

descripcion = wsPreformas.Cells(i, j).Value

If regex.Test(descripcion) Then

numeroExtraido = CDbl(regex.Execute(descripcion)(0).SubMatches(0)) \* 1000

stkU = wsPreformas.Cells(i, wsPreformas.Rows(1).Find("Stk.(U)").Column).Value

**' Si el número extraído multiplicado por 1000 es mayor que el valor en "Stk.(U)", pintar la fila de rojo**

If numeroExtraido > stkU Then

**' Identificar la última columna con datos en la fila actual**

lastColData = wsPreformas.Cells(i, wsPreformas.Columns.count).End(xlToLeft).Column

**' Colorear de rojo hasta la última columna**

wsPreformas.Range(wsPreformas.Cells(i, 1), wsPreformas.Cells(i, lastColData)).Interior.Color = RGB(255, 100, 100)

**' Salir del bucle de columnas, ya que ya se ha pintado toda la fila**

Exit For

End If

End If

Next j

Next i

**' Ajustar automáticamente el tamaño de las columnas**

wsPreformas.Columns.AutoFit

wsPreformas.Columns("B").ColumnWidth = 35

wsPreformas.Columns("G").ColumnWidth = 35

wsPreformas.Columns("I").ColumnWidth = 35

**' Formato de tabla**

With wsPreformas

.Range("A1:I" & .Cells(.Rows.count, "A").End(xlUp).Row).Borders.LineStyle = xlContinuous

End With

**‘Definir más variables**

Dim hojaDestino As Worksheet

Dim ultimaFilaDatos As Long

Dim filaActual As Long

**' Establecer la hoja Preformas**

Set hojaDestino = ThisWorkbook.Sheets("Preformas")

**' Encontrar la última fila con datos en la columna G**

ultimaFilaDatos = hojaDestino.Cells(hojaDestino.Rows.count, "G").End(xlUp).Row

**' Recorrer cada fila y multiplicar los valores de las columnas G y H**

For filaActual = 2 To ultimaFilaDatos ' Asumiendo que hay encabezado en la primera fila

hojaDestino.Cells(filaActual, "I").Value = hojaDestino.Cells(filaActual, "G").Value \* hojaDestino.Cells(filaActual, "H").Value

Next filaActual

**' Establecer la hoja Preformas**

Set hojaDestino = ThisWorkbook.Sheets("Preformas")

**' Encontrar la última fila con datos en la columna F**

ultimaFilaDatos = hojaDestino.Cells(hojaDestino.Rows.count, "F").End(xlUp).Row

**' Ordenar las columnas F, G, H, I de menor a mayor por la columna F**

hojaDestino.Sort.SortFields.Clear

hojaDestino.Sort.SortFields.Add key:=hojaDestino.Range("F1:F" & ultimaFilaDatos), \_

SortOn:=xlSortOnValues, Order:=xlAscending, DataOption:=xlSortNormal

With hojaDestino.Sort

.SetRange hojaDestino.Range("F1:I" & ultimaFilaDatos) **' Ajusta el rango a las columnas F, G, H, I**

.Header = xlYes

.MatchCase = False

.Orientation = xlTopToBottom

.SortMethod = xlPinYin

.Apply

End With

hojaMadre.Cells(6, 2).Value = "Total Costos s/IVA:"

Dim hojaOrigen As Worksheet

Dim ultimaFilaOrigen As Long

Dim filaDestino As Long

Dim dict As Object

Dim valorCelda As String

Dim valorSuma As Double

Dim totalSuma As Double

**' Establecer las hojas**

Set hojaOrigen = ThisWorkbook.Sheets("Preformas")

Set hojaDestino = ThisWorkbook.Sheets("Madre")

**' Crear un diccionario para almacenar la suma de valores por carácter**

Set dict = CreateObject("Scripting.Dictionary")

**' Encontrar la última fila con datos en la columna F de la hoja Preformas**

ultimaFilaOrigen = hojaOrigen.Cells(hojaOrigen.Rows.count, "F").End(xlUp).Row

**' Recorrer cada fila de la columna F y sumar los valores correspondientes de la columna I**

For filaActual = 2 To ultimaFilaOrigen **' Asumiendo que hay encabezado en la primera fila**

valorCelda = hojaOrigen.Cells(filaActual, "F").Value

valorSuma = hojaOrigen.Cells(filaActual, "I").Value

If valorCelda <> "" And valorCelda <> "s/Sustitutos" Then

If dict.exists(valorCelda) Then

dict(valorCelda) = dict(valorCelda) + valorSuma

Else

dict.Add valorCelda, valorSuma

End If

End If

Next filaActual

**' Inicializar la fila de destino en la hoja Madre**

filaDestino = 7 ' Corresponde a la fila C7

**' Colocar los resultados en la hoja Madre**

For Each clave In dict.Keys

hojaDestino.Cells(filaDestino, 2).Value = clave ' Columna B

hojaDestino.Cells(filaDestino, 3).Value = dict(clave) ' Columna C

totalSuma = totalSuma + dict(clave) **' Sumar para el total**

filaDestino = filaDestino + 1

Next clave

**' Imprimir la suma total en C6**

hojaDestino.Cells(6, 3).Value = totalSuma ' Columna C, fila 6

End If

Dim hoja As Worksheet

Set hoja = ThisWorkbook.Sheets("Madre") **' Ajusta el nombre de la hoja si es necesario**

**' Dividir el valor de G13 entre A2 y actualizar el resultado en G13**

If hoja.Range("A2").Value <> 0 Then **' Verificar que el divisor no sea cero**

hoja.Range("G13").Value = hoja.Range("G13").Value / hoja.Range("A2").Value

Else

MsgBox "Error: El divisor en la celda A2 no puede ser cero.", vbExclamation

End If

Set wsMadre = ThisWorkbook.Sheets("Madre")

End Sub