**FUNCIÓN PARA CORROBORAR EL ESTADO DE LA CHEACK LIST**

**‘Define la función**

Sub Grumos()

**‘Declara variables**

Dim ws As Worksheet

Dim cb As Shape

Dim estado As String

**‘Declara en que hoja trabajara**

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

**' Recorre todos los controles en la hoja**

For Each cb In ws.Shapes

**' Verifica que el control sea un checkbox**

If cb.Type = msoFormControl And cb.FormControlType = xlCheckBox Then

**' Determina el estado del checkbox**

Select Case cb.Name

Case "CheckBox1"

If cb.ControlFormat.Value = xlOn Then

Call Funcion1 **' Llama a la función 1**

End If

Case "CheckBox2"

If cb.ControlFormat.Value = xlOn Then

Call Funcion2 **' Llama a la función 2**

End If

Case "CheckBox3"

If cb.ControlFormat.Value = xlOn Then

Call Funcion3 **' Llama a la función 3**

End If

Case "CheckBox4"

If cb.ControlFormat.Value = xlOn Then

Call Funcion4 **' Llama a la función 4**

End If

End Select

End If

Next cb

Sheets("Madre").Activate

End Sub

**FUNCION PARA EL APARTADO DE GRUMOS EN BASE A LA CHEACK LIST**

**‘Definir Función**

Sub Funcion1()

**‘Definir variables**

Dim wsOriginal As Worksheet

Dim wsGrumos As Worksheet

Dim ultimaFila As Long

Dim i As Long

Dim celda As Range

Dim sumaPorDenominacion As Object ' Diccionario para agrupar denominaciones

Dim uniNegPorDenominacion As Object ' Diccionario para almacenar Uni.Neg por denominación

Dim ultimaDenominacionPorGrupo As Object ' Diccionario para almacenar la última denominación usada

Dim countPorDenominacion As Object ' Diccionario para almacenar la cantidad de términos usados

Dim resultadoFila As Long

Dim headerRow As Long

Dim found As Boolean

Dim columnaStkP As Range

Dim columnaUniNeg As Range

Dim columnaCosto As Range ' Nueva variable para la columna Costo s/IVA

Dim rngDenominacion As Range

Dim wsGrumosLastRow As Long

Dim headerExists As Boolean

Dim denominacion As String

Dim caracteresEspecificos As Variant

Dim agruparDenominacion As String

Dim caracterEncontrado As String

Dim costoPorDenominacion As Object ' Diccionario para almacenar el costo por denominación

' **Establecer la hoja original de la cual extraerá los datos**

Set wsOriginal = ThisWorkbook.Sheets("BD")

**' Crear o seleccionar la hoja llamada "GRUMOS"**

On Error Resume Next

Set wsGrumos = ThisWorkbook.Sheets("GRUMOS")

If wsGrumos Is Nothing Then

Set wsGrumos = ThisWorkbook.Sheets.Add

wsGrumos.Name = "GRUMOS"

End If

On Error GoTo 0

**' Verificar si ya existe una cabecera en "GRUMOS"**

If wsGrumos.Cells(1, 1).Value = "Denominación" And wsGrumos.Cells(1, 2).Value = "Suma Stk.(P)" And \_

wsGrumos.Cells(1, 3).Value = "Uni.Neg" And wsGrumos.Cells(1, 4).Value = "Sustituto" And \_

wsGrumos.Cells(1, 5).Value = "Cantidad de Términos" And wsGrumos.Cells(1, 6).Value = "Costo s/IVA" Then

headerExists = True

Else

headerExists = False

End If

**' Buscar la columna con la cabecera "Denominación"**

found = False

For headerRow = 1 To wsOriginal.Rows.count

Set rngDenominacion = wsOriginal.Rows(headerRow).Find("Denominación", LookIn:=xlValues, LookAt:=xlWhole)

If Not rngDenominacion Is Nothing Then

found = True

Exit For

End If

Next headerRow

If Not found Then

MsgBox "No se encontró la columna 'Denominación'. Asegúrate de que el encabezado esté presente en alguna fila.", vbCritical

Exit Sub

End If

**' Buscar la columna con la cabecera "Stk.(P)"**

found = False

For headerRow = 1 To wsOriginal.Rows.count

Set columnaStkP = wsOriginal.Rows(headerRow).Find("Stk.(P)", LookIn:=xlValues, LookAt:=xlWhole)

If Not columnaStkP Is Nothing Then

found = True

Exit For

End If

Next headerRow

If Not found Then

MsgBox "No se encontró la columna 'Stk.(P)'. Asegúrate de que el encabezado esté presente en alguna fila.", vbCritical

Exit Sub

End If

**' Buscar la columna con la cabecera "Uni.Neg"**

found = False

For headerRow = 1 To wsOriginal.Rows.count

Set columnaUniNeg = wsOriginal.Rows(headerRow).Find("Uni.Neg", LookIn:=xlValues, LookAt:=xlWhole)

If Not columnaUniNeg Is Nothing Then

found = True

Exit For

End If

Next headerRow

If Not found Then

MsgBox "No se encontró la columna 'Uni.Neg'. Asegúrate de que el encabezado esté presente en alguna fila.", vbCritical

Exit Sub

End If

**' Buscar la columna con la cabecera "Costo s/IVA"**

found = False

For headerRow = 1 To wsOriginal.Rows.count

Set columnaCosto = wsOriginal.Rows(headerRow).Find("Costo s/IVA", LookIn:=xlValues, LookAt:=xlWhole)

If Not columnaCosto Is Nothing Then

found = True

Exit For

End If

Next headerRow

If Not found Then

MsgBox "No se encontró la columna 'Costo s/IVA'. Asegúrate de que el encabezado esté presente en alguna fila.", vbCritical

Exit Sub

End If

**' Crear diccionarios para almacenar los valores**

Set sumaPorDenominacion = CreateObject("Scripting.Dictionary")

Set uniNegPorDenominacion = CreateObject("Scripting.Dictionary")

Set ultimaDenominacionPorGrupo = CreateObject("Scripting.Dictionary")

Set countPorDenominacion = CreateObject("Scripting.Dictionary") **' Nuevo diccionario para contar términos**

Set costoPorDenominacion = CreateObject("Scripting.Dictionary") **' Nuevo diccionario para almacenar el costo por denominación**

Set caracteresUsados = CreateObject("Scripting.Dictionary") **' Nuevo diccionario para almacenar caracteres usados**

**' Definir los caracteres específicos**

caracteresEspecificos = 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)")

**' Recorrer las celdas en la columna de "Denominación"**

For i = rngDenominacion.Row + 1 To wsOriginal.Cells(wsOriginal.Rows.count, rngDenominacion.Column).End(xlUp).Row

Set celda = wsOriginal.Cells(i, rngDenominacion.Column)

denominacion = celda.Value

agruparDenominacion = denominacion

caracterEncontrado = ""

**' Validar y ajustar los valores vacíos o invisibles en "Stk.(P)"**

If Trim(wsOriginal.Cells(i, columnaStkP.Column).Value) = "" Then

wsOriginal.Cells(i, columnaStkP.Column).Value = 0

End If

**' Solo considerar filas donde "Uni.Neg" sea igual a "Deposito Materia Prima" y "Stk.(P)" mayor a 0**

If wsOriginal.Cells(i, columnaUniNeg.Column).Value = "San Vicente - Cba" And wsOriginal.Cells(i, columnaStkP.Column).Value > 0 Then ‘Aqui se define el criterio de busqueda por la cheack list

**' Verificar si la denominación contiene alguno de los caracteres específicos**

For Each caracter In caracteresEspecificos

If InStr(denominacion, caracter) > 0 Then

agruparDenominacion = caracter

caracterEncontrado = caracter

**' Almacenar el carácter específico usado**

If Not caracteresUsados.exists(caracter) Then

caracteresUsados.Add caracter, True

End If

Exit For

End If

Next caracter

**' Verificar si la denominación agrupada ya está en el diccionario**

If sumaPorDenominacion.exists(agruparDenominacion) Then

**' Sumar el valor de "Stk.(P)" a la denominación existente**

sumaPorDenominacion(agruparDenominacion) = sumaPorDenominacion(agruparDenominacion) + wsOriginal.Cells(i, columnaStkP.Column).Value

**' Aumentar el contador de términos**

countPorDenominacion(agruparDenominacion) = countPorDenominacion(agruparDenominacion) + 1

**' Actualizar la última denominación usada**

ultimaDenominacionPorGrupo(agruparDenominacion) = denominacion

Else

**' Agregar una nueva entrada con el valor de "Stk.(P)"**

sumaPorDenominacion.Add agruparDenominacion, wsOriginal.Cells(i, columnaStkP.Column).Value

**' Agregar el valor de "Uni.Neg"**

uniNegPorDenominacion.Add agruparDenominacion, wsOriginal.Cells(i, columnaUniNeg.Column).Value

**' Guardar la primera denominación encontrada para este grupo**

ultimaDenominacionPorGrupo.Add agruparDenominacion, denominacion

**' Inicializar el contador de términos en 1**

countPorDenominacion.Add agruparDenominacion, 1

**' Tomar el costo correspondiente solo la primera vez**

costoPorDenominacion.Add agruparDenominacion, wsOriginal.Cells(i, columnaCosto.Column).Value

End If

End If

Next i

**' Determinar la última fila de la hoja "GRUMOS" para agregar datos**

wsGrumosLastRow = wsGrumos.Cells(wsGrumos.Rows.count, 1).End(xlUp).Row

**' Si no existe la cabecera, agregarla**

If Not headerExists Then

wsGrumos.Cells(wsGrumosLastRow, 1).Value = "Denominación"

wsGrumos.Cells(wsGrumosLastRow, 2).Value = "Suma Stk.(P)"

wsGrumos.Cells(wsGrumosLastRow, 3).Value = "Uni.Neg"

wsGrumos.Cells(wsGrumosLastRow, 4).Value = "Sustituto" wsGrumos.Cells(wsGrumosLastRow, 5).Value = "Cantidad de Términos" wsGrumos.Cells(wsGrumosLastRow, 6).Value = "Costo s/IVA"

wsGrumosLastRow = wsGrumosLastRow + 1

wsGrumos.Range("A1:G1").Interior.Color = RGB(173, 216, 230) ' Celeste claro

End If

**' Escribir los resultados agrupados por denominación en la hoja "GRUMOS"**

resultadoFila = wsGrumosLastRow

**‘Declarar más variables**

Dim key As Variant

Dim isDuplicate As Boolean

Dim existingRow As Long

For Each key In sumaPorDenominacion.Keys

isDuplicate = False

**' Comparar con las filas existentes en GRUMOS**

For existingRow = 2 To wsGrumos.Cells(wsGrumos.Rows.count, 1).End(xlUp).Row **' Comienza desde 2 para saltar la cabecera**

If wsGrumos.Cells(existingRow, 1).Value = ultimaDenominacionPorGrupo(key) And \_

wsGrumos.Cells(existingRow, 2).Value = sumaPorDenominacion(key) And \_

wsGrumos.Cells(existingRow, 3).Value = uniNegPorDenominacion(key) And \_

wsGrumos.Cells(existingRow, 4).Value = key And \_

wsGrumos.Cells(existingRow, 5).Value = countPorDenominacion(key) And \_

wsGrumos.Cells(existingRow, 6).Value = costoPorDenominacion(key) Then

isDuplicate = True

Exit For

End If

Next existingRow

**' Si no es duplicado, añadir a la hoja**

If Not isDuplicate Then

wsGrumos.Cells(resultadoFila, 1).Value = ultimaDenominacionPorGrupo(key) **' Colocar la última denominación usada**

wsGrumos.Cells(resultadoFila, 2).Value = sumaPorDenominacion(key)

wsGrumos.Cells(resultadoFila, 3).Value = uniNegPorDenominacion(key)

**' Verificar si se encontró un carácter específico**

If key <> "" Then

wsGrumos.Cells(resultadoFila, 4).Value = key **' Colocar el carácter específico**

Else

wsGrumos.Cells(resultadoFila, 4).Value = "" **' Si no se encontró, dejar vacío**

End If

wsGrumos.Cells(resultadoFila, 5).Value = countPorDenominacion(key) **' Colocar la cantidad de términos**

**' Tomar el costo correspondiente solo una vez por denominación**

If costoPorDenominacion.exists(key) Then

wsGrumos.Cells(resultadoFila, 6).Value = costoPorDenominacion(key) **' Colocar el costo s/IVA correspondiente**

Else

wsGrumos.Cells(resultadoFila, 6).Value = "" **' Si no se encontró, dejar vacío**

End If

resultadoFila = resultadoFila + 1

End If

Next key

**' Limpiar objetos**

Set sumaPorDenominacion = Nothing

Set uniNegPorDenominacion = Nothing

Set ultimaDenominacionPorGrupo = Nothing

Set countPorDenominacion = Nothing

Set caracteresUsados = Nothing

Set costoPorDenominacion = Nothing

End Sub

**Esta función llamada “Función1” se repetira la cantidad de veces segun los items de la cheack list, por ejemplo, al pegar otra vez este codigo por debajo cambiarle el nombre por “Función2” y en la linea adecuada e identificada cambiar el nombre del criterio de busqueda.**