

# FUNCIÓN PARA CORROBORAR EL ESTADO DE LA CHECK LIST

## ‘Define la función

Sub Preformas ()

## ‘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 PREFORMAS EN BASE A LA CHEACK LIST

```

Sub Funcion1()

```

**En este caso la funcion de la cheack list 1 (San Vicente - Cba) se evita ya que es 100% seguro de que aca no se registraron ni se registraran datos sobre las preformas**

```

End Sub

```

## **‘Definimos la función**

Sub Funcion2()

### **‘Declarar variables**

Dim hojaOrigen As Worksheet, hojaDestino As Worksheet

Dim ultimaFilaOrigen As Long, ultimaFilaDestino As Long

Dim i As Long

Dim hojaExiste As Boolean

Dim hoja As Worksheet

Dim columnaDenominacion As Long, columnaStk As Long, columnaUniNeg As Long,  
columnaCosto As Long

Dim celda As Range

Dim datosCopiados As String

Dim sustitutosArray As Variant

Dim sustituto As String

Dim filaDestino As Long

Dim dictSustitutos As Object

Dim valorStk As Variant ' Cambiar a Variant para manejar diferentes tipos de datos

Dim valorCosto As Variant ' Cambiar a Variant para manejar diferentes tipos de datos

Dim filaResultado As Long

Dim valoresSinCaracter As Collection ' Colección para almacenar valores de Stk.(U)  
sin caracteres

Dim costosSinCaracter As Collection ' Colección para almacenar valores de Costo  
s/IVA sin caracteres

Dim dictMaxCosto As Object ' Diccionario para almacenar el costo máximo por cada  
sustituto

### **' Crear el diccionario para sumar los valores de "Stk.(U)" por cada carácter**

Set dictSustitutos = CreateObject("Scripting.Dictionary")

Set dictMaxCosto = CreateObject("Scripting.Dictionary") ' Inicializar el diccionario para costos

Set valoresSinCaracter = New Collection ' Inicializar la colección

Set costosSinCaracter = New Collection ' Inicializar la colección para costos sin caracteres

### **' Definir el array con los caracteres a extraer**

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)"

#### **' Definir la hoja origen**

Set hojaOrigen = Worksheets("BD")

#### **' Buscar las columnas correspondientes a cada encabezado**

For Each celda In hojaOrigen.UsedRange

    If celda.Value = "Denominación" Then columnaDenominacion = celda.Column

    If celda.Value = "Stk.(U)" Then columnaStk = celda.Column

    If celda.Value = "Uni.Neg" Then columnaUniNeg = celda.Column

    If celda.Value = "Costo s/IVA" Then columnaCosto = celda.Column

Next celda

#### **' Verificar si la hoja "Preformas" existe**

hojaExiste = False

For Each hoja In Worksheets

    If hoja.Name = "Preformas" Then

        hojaExiste = True

    Exit For

End If

Next hoja

**' Si la hoja no existe, crearla**

If Not hojaExiste Then

    Set hojaDestino = Worksheets.Add

    hojaDestino.Name = "Preformas"

Else

    Set hojaDestino = Worksheets("Preformas")

End If

**' Verificar si los encabezados ya están presentes**

If hojaDestino.Cells(1, 1).Value <> "Denominación" Then

**' Limpiar el contenido de la hoja destino antes de agregar encabezados**

    hojaDestino.Cells.ClearContents

**' Copiar encabezados a la fila 1**

    hojaDestino.Cells(1, 1).Value = "Denominación"

    hojaDestino.Cells(1, 2).Value = "Stk.(U)"

    hojaDestino.Cells(1, 3).Value = "Uni.Neg"

    hojaDestino.Cells(1, 4).Value = "Sustitutos"

    hojaDestino.Cells(1, 6).Value = "Sustituto"

    hojaDestino.Cells(1, 7).Value = "Suma Stk.(U)"

    hojaDestino.Cells(1, 8).Value = "Máximo Costo s/IVA"

```

hojaDestino.Cells(1, 9).Value = "Total Costo s/IVA"

hojaDestino.Range("A1:D1").Interior.Color = RGB (173, 216, 230) ' Celeste claro

hojaDestino.Range("F1:I1").Interior.Color = RGB (173, 216, 230) ' Celeste claro

End If

```

**' Encontrar la última fila con datos en la hoja origen**

```

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

```

**' Inicializar la fila de destino**

```

filaDestino = hojaDestino.Cells(hojaDestino.Rows.count, 1).End(xlUp).Row + 1 '
Comenzar desde la primera fila vacía

```

**' Iterar a través de los datos en la hoja origen**

```

For i = 2 To ultimaFilaOrigen ' Asumiendo que la fila 1 es el encabezado

```

**' Verificar las condiciones para "Denominación", "Stk.(U)", y "Uni.Neg"**

```

If UCase(hojaOrigen.Cells(i, columnaDenominacion).Value) Like "PREF*" And _
    hojaOrigen.Cells(i, columnaStk).Value > 0 And _
    hojaOrigen.Cells(i, columnaUniNeg).Value = "Planta-Administración Central"
Then 'Esta línea define el criterio a partir de la cheack list

```

**' Obtener el valor de "Stk.(U)" y comprobar si es numérico**

```

valorStk = hojaOrigen.Cells(i, columnaStk).Value

```

**' Solo continuar si valorStk es mayor que cero**

```

If IsNumeric(valorStk) And valorStk > 0 Then

```

**' Copiar los valores de las columnas requeridas**

```

    hojaDestino.Cells(filaDestino, 1).Value = hojaOrigen.Cells(i,
columnaDenominacion).Value ' Denominación

```



```
hojaDestino.Cells(filaDestino, 2).Value = valorStk ' Stk.(U)
```

```
hojaDestino.Cells(filaDestino, 3).Value = hojaOrigen.Cells(i,  
columnaUniNeg).Value ' Uni.Neg
```

```
' Extraer los caracteres definidos en el array y concatenarlos
```

```
sustituto = ""
```

```
For Each elemento In sustitutosArray
```

```
    If InStr(1, hojaOrigen.Cells(i, columnaDenominacion).Value, elemento,  
vbTextCompare) > 0 Then
```

```
        sustituto = sustituto & elemento & ", "
```

```
    End If
```

```
Next elemento
```

```
' Quitar la última coma y espacio
```

```
If Len(sustituto) > 0 Then
```

```
    sustituto = Left(sustituto, Len(sustituto) - 2)
```

```
End If
```

```
' Colocar el valor de sustitutos en la columna "Sustitutos"
```

```
hojaDestino.Cells(filaDestino, 4).Value = sustituto
```

```
' Sumar los valores de "Stk.(U)" por cada carácter en "Sustitutos"
```

```
If Len(sustituto) > 0 Then
```

```
    For Each elemento In Split(sustituto, ", ")
```

```
        If dictSustitutos.exists(elemento) Then
```

```
            dictSustitutos(elemento) = dictSustitutos(elemento) + CDbI(valorStk)
```

```
        Else
```

```
            dictSustitutos.Add elemento, CDbI(valorStk)
```

End If

**' Obtener el valor de "Costo s/IVA" y almacenarlo**

valorCosto = hojaOrigen.Cells(i, columnaCosto).Value

If IsNumeric(valorCosto) Then

If dictMaxCosto.exists(elemento) Then

If valorCosto > dictMaxCosto(elemento) Then

dictMaxCosto(elemento) = valorCosto **' Actualizar con el valor**

**máximo**

End If

Else

dictMaxCosto.Add elemento, valorCosto **' Agregar el valor al**

**diccionario**

End If

End If

Next elemento

Else

**' Almacenar el valorStk en la colección si no tiene sustituto**

valoresSinCaracter.Add valorStk

costosSinCaracter.Add hojaOrigen.Cells(i, columnaCosto).Value '

**Almacenar Costo s/IVA sin carácter**

End If

**' Incrementar la fila de destino**

filaDestino = filaDestino + 1

End If

End If

Next i

**' Imprimir los resultados de la suma en la hoja "Preformas"**

filaResultado = 2 ' Iniciar desde la fila 2

For Each elemento In dictSustitutos.Keys

hojaDestino.Cells(filaResultado, 6).Value = elemento ' **Sustituto**

hojaDestino.Cells(filaResultado, 7).Value = dictSustitutos(elemento) ' **Suma Stk.(U)**

If dictMaxCosto.exists(elemento) Then

hojaDestino.Cells(filaResultado, 8).Value = dictMaxCosto(elemento) ' **Máximo Costo s/IVA**

End If

filaResultado = filaResultado + 1

Next elemento

**' Imprimir los valores de Stk.(U) y Costo s/IVA sin carácter**

For i = 1 To valoresSinCaracter.count

hojaDestino.Cells(filaResultado, 6).Value = "s/Sustitutos" ' **Sustituto vacío para los valores sin carácter**

hojaDestino.Cells(filaResultado, 7).Value = valoresSinCaracter(i) ' **Valor de Stk.(U)**

hojaDestino.Cells(filaResultado, 8).Value = costosSinCaracter(i) ' **Costo s/IVA correspondiente**

filaResultado = filaResultado + 1

Next i

End Sub

**Esta función llamada “Función2” se repetirá la cantidad de veces según los ítems de la cheack list, por ejemplo, al pegar otra vez este código por debajo cambiarle el nombre por “Función3” y en la línea adecuada e identificada cambiar el nombre del criterio de búsqueda.**