Tabla de contenido

[Configuración inicial 2](#_Toc42941951)

[Abrir y cerrar archivos 3](#_Toc42941952)

[Al abrir 4](#_Toc42941953)

[Filtros y orden 4](#_Toc42941954)

[Hojas y libros 5](#_Toc42941955)

[Filas y columnas 7](#_Toc42941956)

[Borrar 7](#_Toc42941957)

[Añadir 7](#_Toc42941958)

[Mostrar columnas o filas ocultas 7](#_Toc42941959)

[For/If 8](#_Toc42941960)

[Texto y operaciones raras 9](#_Toc42941961)

[Formato 10](#_Toc42941962)

[Gráficos 13](#_Toc42941963)

[Mensajes 15](#_Toc42941964)

[Proteger y desproteger 16](#_Toc42941965)

[Error 18](#_Toc42941966)

[Limpieza de código 19](#_Toc42941967)

[Set 19](#_Toc42941968)

[With 19](#_Toc42941969)

[UserForm 20](#_Toc42941970)

[ComboBox 20](#_Toc42941971)

# Configuración inicial

Quitar parpadeo, alertas, eventos y cálculos en manual

* Application.ScreenUpdating = False
* Application.DisplayAlerts = False
* Application.Calculation = xlManual
* Application.EnableEvents = False

Cuando se trabaja con matrices, para omitir el espacio 0:

* Option Base 1

Se puede simplificar con un With Application + End With

.DisplayAlerts = False

.DisplayFullScreen = True

.CommandBars("Worksheet Menu Bar").Enabled = False

.ScreenUpdating = True

.AskToUpdateLinks = True

.DisplayAlerts = True

.StatusBar = ""

.CutCopyMode = True

.DisplayFormulaBar = False

Incluir en los archivos genéricos una parte que sean los datos que hay que cambiar para adaptar el archivo, que estén al principio.

Configuración final:

* Application.ScreenUpdating = True
* Application.DisplayAlerts = True
* Application.Calculation = xlCalculationAutomatic
* Application.EnableEvents = True

Momentáneamente podemos actualizar los cálculos

* ActiveSheet.Calculate

# Abrir y cerrar archivos

Activar un libro ya abierto. Siempre que se cierra un libro mandar activar el libro que queremos para que por error no abra otro

* Workbooks("BOOK4.XLS").Activate

Función para ver si un archivo está abierto:

* Function CheckFileIsOpen(chkSumfile As String) As Boolean

On Error Resume Next

CheckFileIsOpen = (Workbooks(chkSumfile).Name = chkSumfile)

On Error GoTo 0

End Function

El directorio actual

* ruta = ThisWorkbook.Path + "\1 Gestion de RBBA-RBMA\1.GENERAL BAJA.xlsm"

Obtener nombre del archivo mediante búsqueda

* Archivo = Application.GetOpenFilename
* If Archivo = False Then 'Si no seleccionamos nada salimos del proceso

Exit Sub

End If

Abrir archivo y cerrarlo

* Workbooks.Open Filename:=Archivo
* Libro = ActiveWorkbook.Name
* Workbooks(Libro).Close SaveChanges:=False

Abrir un archivo solo en modo lectura. Si no se van a hacer modificaciones, solamente se van a copiar datos, mejor hacerlo así:

* Workbooks.Open Filename:=Archivo, ReadOnly:=True

Si se quiere además que no actualice vínculos externos:

* Workbooks.Open Filename:=Directorio(i), UpdateLinks:=False, ReadOnly:=True

Si se quiere abrir un archivo y que no ejecute las macros de este automáticamente antes y después de la línea que ejecuta el archivo se debe escribir:

* Application.EnableEvents = False
* Application.EnableEvents = True

Guardar un archivo:

* ActiveWorkbook.SaveAs Filename:=Directorio2 & NombreNuevoArchivo

## Al abrir

Si se quiere ejecutar una macro cada vez que se abra el archivo se debe meter la macro en el libro en ‘ThisWorkbook’:

* Private Sub workbook\_open()
* Call Laborables
* Call Proteger
* End Sub

Abrir todos los Excel de una carpeta:

Original = ActiveWorkbook.Name

Dim fso As Object, fCarpeta As Object, tmpCarpeta As Object

Dim Fichero As Object, tmpFichero As Object

Dim strRutaInicial As String

strRutaInicial = "C:\Users\peima120-022\Desktop\New folder" 'Ruta que se procesará

Set fso = CreateObject("Scripting.FileSystemObject")

Set fCarpeta = fso.GetFolder(strRutaInicial)

For Each tmpFichero In fCarpeta.Files

Workbooks.Open tmpFichero.Path

Workbooks.Open tmpFichero.Path

LibroOverAll = ActiveWorkbook.Name

Workbooks(LibroOverAll).Close SaveChanges:=False

Next tmpFichero

## Filtros y orden

Cada vez que abramos un nuevo archivo se debe desactivar los filtros para poder tener acceso a todos los datos:

* If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter
* If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData

Para ordenar:

* Range(Cells(7, 1), Cells(Ultimafila\_Old, 200)).Sort key1:=Range("B8"), order1:=xlAscending, Header:=xlYes
* Ordena el rango determinado poniendo en orden al columna B.

# Hojas y libros

Seleccionar una Hoja o Libro por nombre:

* Worksheets(LL).Activate
* Workbooks("BOOK4.XLS").Activate

Seleccionar una hoja por número:

* Worksheets(1).Activate

Cambiar nombre a la hoja seleccionada u obtenerlo de una hoja o libro:

* ActiveSheet.Name = Hoja2
* Hoja=ActiveSheet.Name
* Libro = ActiveWorkbook.Name

Borrar Hoja

* Sheets(i).Delete

Eliminar una hoja sin confirmación

* Application.DisplayAlerts = False  
          WS.Delete  
          Application.DisplayAlerts = True

Copiar una hoja en una posición concreta o al final:

* Sheets(Cliente).Copy Before:=Sheets(7)
* Worksheets("Tarea1").Copy After:=Sheets(Sheets.Count)
* Application.Run "getUnits"
* Sheets("Workbook (2)").Copy Before:=Workbooks("Formato Hysis 2.xlsm").Sheets(1)

Crear una nueva hoja y añadirla después de la activa

* Sheets.Add After:=ActiveSheet

Obtener el número de hojas de un libro:

* Numero\_Hojas = Application.Sheets.Count

Hacer visible hoja/ dejarla muy escondida

* Sheets(Cover).Visible = xlSheetVisible
* Sheets(Cover).Visible = xlSheetVeryHidden

Mover hojas a un libro nuevo

* Sheets(Array(Cover2, Legend2, Cliente2)).Move

Borrar hojas repetidas:

* NombreHoja = "Pivot Table"
* Numero\_Hojas = Application.Sheets.Count
* For i = Numero\_Hojas To 1 Step -1
* If Sheets(i).Name = NombreHoja Then
* Sheets(i).Delete
* End If
* Next i

Abrir u archivo sin notificaciones

Application.EnableEvents = False

Workbooks.Open Filename:=Directorio(1)

Application.EnableEvents = True

# Definiciones

Si queremos tener una variable disponible en todos los archivo la definimos como ‘public’ al principio:

* Public Plano As Integer

10 (36.0.0.249)

# Filas y columnas

Última fila y columna

* UltimaFila = Cells(Rows.Count, 1).End(xlUp).Row
* UltimaColumna = Cells(filaValores, Columns.Count).End(xlToLeft).Column

Estructura para estar seguro de seleccionar la última columna:

For i = 1 To 10

If Cells(Rows.Count, i).End(xlUp).Row > FilaEscritura Then

FilaEscritura = Cells(Rows.Count, i).End(xlUp).Row

End If

Next i

Seleccionar o borrar filas o columnas

* Range(C\_Auxiliar & ":" & C\_Auxiliar).Columns.Delete
* Rows(FilaResultados & ":" & UltimaFilaFormato).Select

Última fila y columna de un rango:

* x = Selection.Rows(1).Row
* y = Selection.Rows.Count + x – 1

Tamaño:

* Selection.ColumnWidth = 12
* Selection.RowHeight = 12

Ajustar en automáticos:

* Selection.EntireRow.AutoFit
* Columns("A:WW").EntireColumn.AutoFit

Centrar texto horizontalmente, verticalmente y que se justifique

* With Selection
* .HorizontalAlignment = xlCenter
* .VerticalAlignment = xlCenter
* .WrapText = True
* End With

## Borrar

Se puede borrar en otras hojas directamente:

* Ws.Rows(i & ":" & i).ClearContents

Para borrar y eliminar formato

* Ws.Rows(i & ":" & i).Clear

Eliminar una línea del todo y que todo se vaya para arriba o a la izquierda

* Selection.Delete Shift:=xlUp
* Selection.Delete Shift:=xlToLeft

Quitar modo de copiar y pegar. También evita que salga el cartel del portapapeles:

* Application.CutCopyMode = False

## Añadir

También se pueden insertar nuevas líneas en blanco por encima de la seleccionada:

* Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove

Pegar:

* Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks:=False, Transpose:=False

## Mostrar columnas o filas ocultas

* Columns("B:ZZ").EntireColumn.Hidden = False
* Selection.EntireRow.Hidden = False ‘Hay que seleccionar primero una celda de dicha columna

# For/If/Case

Para ir al revés o con distinto paso:

* For k = NumeroHojas To 1 Step -1

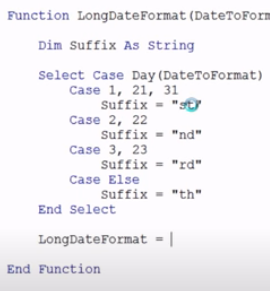
Otra manera de hacer los bucles for (La variable ‘element’ tiene que ser del tipo variant)

* for each element in sarray
* do\_something(element)
* next element

Else if

* ElseIf Cells(FilaColumna, m) <> "" Then

Case puede ser muy útil y sencillo



# Texto

Posición de un determinado texto de izquierda a derecha (a partir del primer carácter) y al revés:

* Posicion = InStr(1, Trim(Cells(FilaResultados, i)), " / ")
* Posicion = InStrRev(Directorio, "\")

Tabulador: vbTab

# Operaciones raras

Resto de una división:

* result = number1 Mod number2

Cociente entero de una división

* result = expression1 \ expression2

Referirte dinámicamente a diferentes hojas mediante fórmulas:

http://spreadsheetpro.net/usar-indirecto-para-dinamicamente-referir-a-hojas-de-trabajo-en-excel-y-google-spreadsheets/

# Filtrar

* Range(Cells(FilaEscritura + 1, ColumnaEscritura + Distancia + Selector), Cells(FilaEscritura + 1, j - 1)).Select
* Selection.AutoFilter

# Formato

Formato decimales

* Selection.NumberFormat = "0.0"

Condicional del color de una celda

* If Cells(i, 11).Interior.Color = 32896 Then

Negrita

* Selection.Font.Bold = True

Poner cuadro

* Selection.Borders(xlEdgeLeft).LineStyle = xlContinuous
* Selection.Borders(xlEdgeBottom).LineStyle = xlContinuous
* Selection.Borders(xlEdgeTop).LineStyle = xlContinuous
* Selection.Borders(xlEdgeRight).LineStyle = xlContinuous
* Selection.Borders(xlInsideVertical).LineStyle = xlContinuous
* Selection.Borders(xlInsideHorizontal).LineStyle = xlContinuous

Alineación del texto dentro del cuadro

* Selection.HorizontalAlignment = xlCenter
* Selection.VerticalAlignment = xlBottom

Ajustar automáticamente el ancho de las columnas:

* Columns("A:T").EntireColumn.AutoFit

General del tipo de letra

With Selection.Font

.Name = "Arial"

.Size = 7

.Strikethrough = False

.Superscript = False

.Subscript = False

.OutlineFont = False

.Shadow = False

.Underline = xlUnderlineStyleNone

.ColorIndex = xlAutomatic

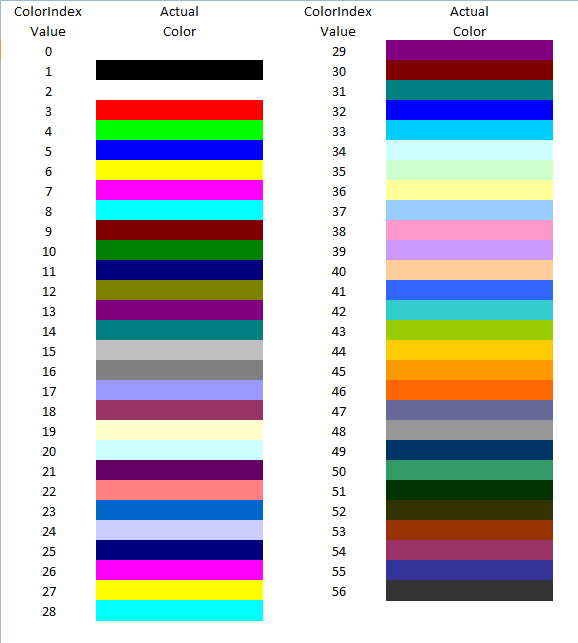
.TintAndShade = 0

.ThemeFont = xlThemeFontNone

End With

Color

Selection.Interior.ColorIndex = 15

1

# Gráficos

Borra gráficos antiguos que pueda haber en la hoja:

* ActiveSheet.ChartObjects.Delete

Se crea el gráfico en la posición que se desea

* Dim grafico As ChartObject
* Set Grafico = wks.ChartObjects.Add(Left:=25, Width:=600, Top:=1380, Height:=300)

Se selecciona el tipo de gráfico

* grafico.Chart.ChartType = xlXYScatterSmooth

Se le da un nuevo nombre al gráfico

* Set cht = ActiveWorkbook.Sheets(1).ChartObjects("Grafico\_1").Chart

Meter una nueva serie de datos. Se pueden meter tantas como uno quiera:

* cht.SeriesCollection.NewSeries.Select
* With Selection
* .Name = "Material"
* .XValues = Range("C59:C76")
* .Values = Range("D59:D76")
* End With

Título gráfico

* cht.HasTitle = True
* cht.ChartTitle.Text = "Condiciones de trabajo"
* 'cht.Axes(xlValue, xlPrimary).AxisTitle.Text = "Presión"
* With grafico.Chart.Axes(xlCategory, xlPrimary)
* .HasTitle = True ' Se agrega el título , sin recuadro
* .AxisTitle.Characters.Text = "Temperatura"
* End With
* With grafico.Chart.Axes(xlValue, xlPrimary)
* .HasTitle = True
* .AxisTitle.Characters.Text = "Presión"
* End With
* End Sub

Coger el rango de otra hoja para los gráficos:

* ActiveChart.FullSeriesCollection(1).XValues = wsAux.Range(wsAux.Cells(FilaUno, 2), wsAux.Cells(FilaFin - 1, 2))
* ActiveChart.FullSeriesCollection(1).Values = wsAux.Range(wsAux.Cells(FilaUno, 4), wsAux.Cells(FilaFin - 1, 4))

ActiveChart.FullSeriesCollection(1).ApplyDataLabels

# Mensajes

Mensaje informativo:

* MsgBox "Hello World!"

Introducir datos

* Num = Application.InputBox("Enter a number")

Barra de estado

* Application.StatusBar = "Contenedor\_ATI"

With Selection.Validation

.Delete

.Add Type:=xlValidateInputOnly, AlertStyle:=xlValidAlertStop, Operator \_

:=xlBetween

.IgnoreBlank = True

.InCellDropdown = True

.InputTitle = "Valor anterior:"

.InputMessage = "Línea nueva"

.ShowInput = True

.ShowError = True

End With

Pregunta:

* answer = MsgBox("Al traer la Lista de Líneas se borrará la que ya está en la Hoja " & NombreHoja & ". ¿Desea continuar?", vbQuestion + vbYesNo + vbDefaultButton2, "Form")
* If answer = vbYes Then

# Proteger y desproteger

Proteger/desproteger (para cada hoja):

* ActiveSheet.Protect Password:="Initec2020", UserInterfaceOnly:=True
* ActiveSheet.Unprotect "Initec2020"

Permitir agrupar/desagrupar

* ActiveSheet.EnableOutlining = True

Permitir filtrar:

* ActiveSheet.Protect AllowFiltering:=True

Lista de todo lo que se puede cambiar:

* ActiveSheet.Protect Password:="pdiagramas1", DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowFormattingCells:=True, AllowFormattingColumns:=True, AllowInsertingRows:=True, AllowDeletingRows:=True, AllowSorting:=True, AllowFiltering:=True

Desproteger unas celdas nada más (se pone antes de la acción de protección)

* Worksheets(i).Range("A3:A5000").Locked = False

Todo:

Sub Proteger\_Hoja()

ActiveSheet.Protect Password:="pdiagramas1", DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowFormattingCells:=True, AllowFormattingColumns:=True, AllowInsertingRows:=True, AllowDeletingRows:=True, AllowSorting:=True, AllowFiltering:=True

End Sub

Sub Proteger\_Todo()

For i = 1 To Sheets.Count

Sheets(i).Protect Password:="pdiagramas1", DrawingObjects:=True, Contents:=True, Scenarios:=True, AllowFormattingCells:=True, AllowFormattingColumns:=True, AllowInsertingRows:=True, AllowDeletingRows:=True, AllowSorting:=True, AllowFiltering:=True

Next i

End Sub

Sub Desproteger\_Hoja()

ActiveSheet.Unprotect "pdiagramas1"

End Sub

Sub Desproteger\_Todo()

For i = 1 To Sheets.Count

Sheets(i).Unprotect "pdiagramas1"

Next i

End Sub

Call Desproteger\_Todo

Call Proteger\_Todo

Call DesProteger\_Hoja

Call Proteger\_Hoja

# Error

En caso de error leer siguiente línea

* On Error Resume Next

Error handler

* ErrorHandler:
* If Err.Number <> 0 Then
* MsgBox Err.Description
* End If
* On Error Resume Next
* wbprocesos.Sheets("TEMPLATE").Delete
* On Error GoTo 0

# Limpieza de código

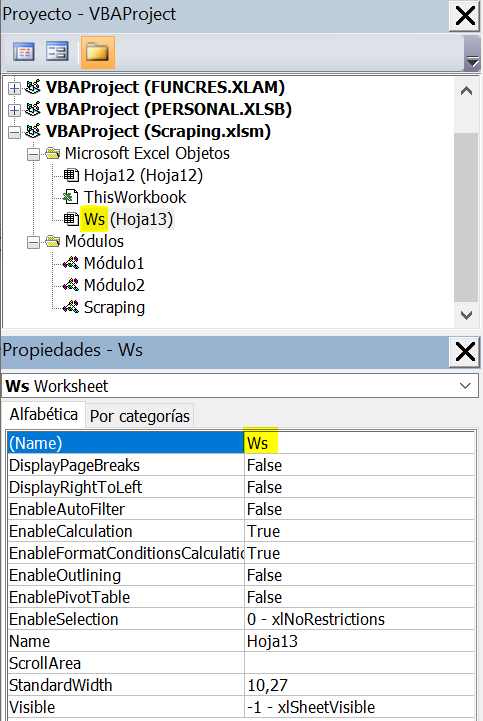
## Set

Sirve para define objetos como libros, hojas o rangos. En general para cualquier de estos objetos que se utilicen mucho

* Dim ws as worksheet
* Set wbprocesos = ActiveWorkbook
* wbprocesos.Sheets("TEMPLATE").Delete
* Set shtCurrIndex = Sheets(ActiveWorkbook.Sheets.Count)
* shtCurrIndex.Cells(24, 1).Value
* Set rng = .Range(.Cells(filaIni, 36), .Cells(lastRow, 36))
* rng.Select
* Set Ws = Worksheets(LibroNuevo)
* WsPCC.Activate
* Set Ws = ThisWorkbook.ActiveSheet

### Poner nombre a las hojas

Como alternativa al set se le puede cambiar el nombre a la hoja:



A partir de ahora podemos llamar a la hoja siempre con instancias como

Ws.Cells(1,1)

## With

Deja el código más ordenado:

* With Selection
* .Borders(xlEdgeLeft).LineStyle = xlContinuous
* End With

# UserForm

Mostrar/Ocultar el UserForm

* UserForm1.Show
* UserForm.Hide

## ComboBox

Coger el valor seleccionado:

* Nombre = UserForm1.ComboBox10.Value

Llenar y vaciar de valores:

* UserForm1.ComboBox10.AddItem Nombres ‘Para llenar lo mejor suele ser un bucle
* UserForm1.ComboBox10.Clear

## ListBox

Subir un elemento dentro de la lista

* Dim ItemNum As Long, TempItem
* With Me.ListBox2
* If .ListIndex <= 0 Then Exit Sub
* ItemNum = .ListIndex
* TempItem = .List(ItemNum)
* .List(ItemNum) = .List(ItemNum - 1)
* .List(ItemNum - 1) = TempItem
* .ListIndex = ItemNum – 1
* End With

Bajar un elemento dentro de la lista

* Dim ItemNum As Long, TempItem
* With Me.ListBox2
* If .ListIndex = .ListCount - 1 Then Exit Sub
* ItemNum = .ListIndex
* TempItem = .List(ItemNum)
* .List(ItemNum) = .List(ItemNum + 1)
* .List(ItemNum + 1) = TempItem
* .ListIndex = ItemNum + 1
* End With

Subirlo, pero todos los da la columna

* Dim ItemNum As Long, TempItem
* For i = 0 To Me.ListBox2.ColumnCount - 1
* With Me.ListBox2
* If .ListIndex <= 0 Then Exit Sub
* ItemNum = .ListIndex
* TempItem = .List(ItemNum, i)
* .List(ItemNum, i) = .List(ItemNum - 1, i)
* .List(ItemNum - 1, i) = TempItem
* End With
* Next i
* Me.ListBox2.ListIndex = ItemNum – 1

Para eliminar elementos

* For i = 0 To ListBox1.ListCount - 1
* If ListBox1.Selected(i) Then
* ListBox1.RemoveItem (i)
* End If
* Next i

Cambiar un elemento de lista haciendo doble click

* Private Sub ListBox1\_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
* On Error Resume Next
* Set a = UserForm2.ListBox1
* Set b = UserForm2.ListBox2
* fila = UserForm2.ListBox1.ListIndex
* b.AddItem a.List(fila, 0)
* b.List(b.ListCount - 1, 1) = a.List(fila, 1)
* a.RemoveItem a.ListIndex
* End Sub

Imprimir los elementos de una lista en Excel

* For i = 0 To UserForm2.ListBox2.ListCount – 1
* Cells(i + 6, 9) = UserForm2.ListBox2.List(i, 0)
* Next i

Para añadir múltiples columnas

lsbListBox1.ColumnCount = 3

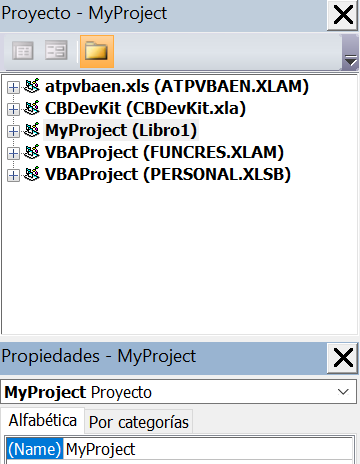
lsbListBox1.ColumnWidths = "50,50,50"

* For icount = 1 to 25
* lsbListBox1.AddItem
* lsbListBox1.List(iCount - 1, 0) = "Item " & iCount
* lsbListBox1.List(iCount - 1, 1) = "Item " & iCount
* lsbListBox1.List(iCount - 1, 2) = "Item " & iCount
* Next iCount

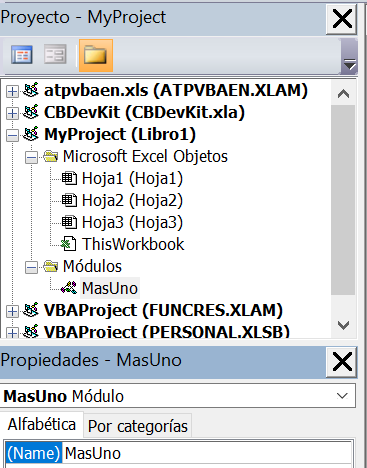
# Complementos

## Crear un complemento

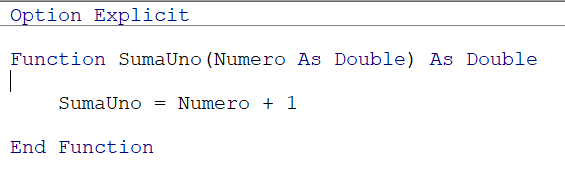
1. Cambiar el nombre del proyecto



1. Crear un módulo y darle un nombre

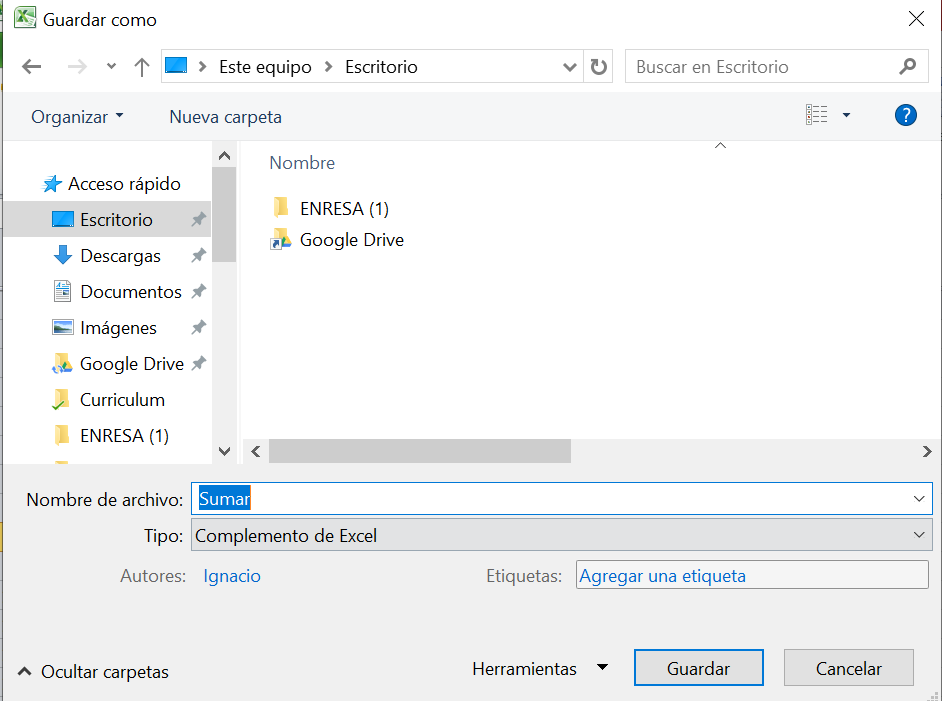


1. Crear la function



1. F

Save as Add-in



Cargar el complemento desde opciones. Se puede acceder al código del complemento si no se protege

# Tablas dinámicas

Sub CrearTablaDinamica()

Application.ScreenUpdating = False

Application.DisplayAlerts = False

Application.Calculation = xlManual

Application.EnableEvents = False

Dim ws As Worksheet

Dim pc As PivotCache

Dim pt As PivotTable

NombreHoja = "Pivot Table"

Numero\_Hojas = Application.Sheets.Count

For i = Numero\_Hojas To 1 Step -1

If Sheets(i).Name = NombreHoja Then

Sheets(i).Delete

End If

Next i

Sheets.Add After:=Sheets(Sheets.Count)

ActiveSheet.Name = "Pivot Table"

'definimos la hoja destino....

Set ws = Worksheets("Pivot Table")

Worksheets(1).Activate

If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter

If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData

UltimaFila = Cells(Rows.Count, 1).End(xlUp).Row

UltimaColumna = Cells(1, Columns.Count).End(xlToLeft).Column

'Creamos la memoria cahce de la TD (Pivot cache)

Set pc = ActiveWorkbook.PivotCaches.Create(xlDatabase, Range(Cells(1, 1), Cells(UltimaFila, UltimaColumna)))

'Ahora generamos la TD

Set pt = pc.CreatePivotTable(ws.Range("A4"))

'configuramos la estructura de la TD

With pt

'llevamos al área de filas los campos:

With .PivotFields("Unidad")

.Orientation = xlRowField

.Position = 1

End With

With .PivotFields("Nombre del empleado o candidato")

.Orientation = xlRowField

.Position = 2

End With

'llevamos al área de Columnas los campos:

With .PivotFields("Fecha")

.Orientation = xlColumnField

.Position = 1

End With

'llevamos al área de Filtros los campos:

With .PivotFields("Centro coste emisor")

.Orientation = xlPageField

.Position = 1

End With

'Insertar Valores

With .PivotFields("Cantidad (un.medida)")

.Orientation = xlDataField

.Position = 1

.Function = xlSum

.NumberFormat = "#,##0"

End With

End With

Worksheets(NombreHoja).Activate

Range("E5").Select

'Agrupar fechas por segundos/minutos/horas/dias/meses/trimestres/años

Selection.Group Start:=True, End:=True, Periods:=Array(False, False, False, \_

False, True, True, True)

Application.ScreenUpdating = True

Application.DisplayAlerts = True

Application.Calculation = xlCalculationAutomatic

Application.EnableEvents = True

End Sub

# Funciones

## Ejecutar funciones a partir de una string

Sub StringExecute(s As String)

Dim vbComp As Object

Set vbComp = ThisWorkbook.VBProject.VBComponents.Add(1)

vbComp.CodeModule.AddFromString "Sub foo()" & vbCrLf & s & vbCrLf & "End Sub"

Application.Run vbComp.name & ".foo"

ThisWorkbook.VBProject.VBComponents.Remove vbComp

End Sub

Sub Testing()

StringExecute "MsgBox" & """" & "Job Done!" & """"

End Sub

## Abrir todos los archivos de una carpeta

Sub AbrirArchivos()

Dim Archivos As String

Archivos = Dir("R:\80078-CKPC POLYPROPYLENE\120-PROCESOS\04-UNIT PPA\13-LL\00-REFERENCE\LL GRACE\\*.xlsx")

Do While Archivos <> ""

Workbook.Open "R:\80078-CKPC POLYPROPYLENE\120-PROCESOS\04-UNIT PPA\13-LL\00-REFERENCE\LL GRACE\" & Archivos

ActiveWorkbook.Name

Stop

ActiveWorkbook.Close SaveChanges:=False

Archivos = Dir

Loop

End Sub

## Directorios de todos los archivo de de un árbol de carpetas

Sub LISTAR\_ARCHIVOS()

'Declaramos variables

Stop

Dim sFSO As Object, Directorio As String

Dim dir\_Archivo As Variant

'Abrimos ventana de diálogo para seleccionar carpeta

Set dir\_Archivo = Application.FileDialog(msoFileDialogFolderPicker)

dir\_Archivo.Show

'Si no seleccionamos nada salimos del proceso

If dir\_Archivo.SelectedItems.Count = 0 Then

Exit Sub

End If

'Capturamos el directorio del archivo seleccionado

Directorio = dir\_Archivo.SelectedItems(1)

'Creamos objeto y ejecutamos función Carpeta

Set sFSO = CreateObject("Scripting.FileSystemObject")

CARPETA sFSO.GetFolder(Directorio)

End Sub

'Y esta es la función:

Function CARPETA(ByVal nCarpeta)

'Declaramos variables

Dim j As Long, Subcarpeta As Object

'Con la hoja activa

With ActiveSheet

'Iniciamos dos loop, uno que recorre las carpetas

For Each Subcarpeta In nCarpeta.SubFolders

CARPETA Subcarpeta

Next

j = Application.CountA(.Range("A:A")) + 1

'y otro que recorre los archivos y los indexa y activa hipervínculo

For Each file In nCarpeta.Files

Posicion = InStrRev(file, "\")

Nombre = Right(file, Len(file) - Posicion)

Posicion = InStrRev(file, ".")

Formato = Right(file, Len(file) - Posicion)

.Cells(j, 1).Select

.Hyperlinks.Add Anchor:=Selection, Address:=file.Path, TextToDisplay:=file.Path

j = j + 1

Next

End With

End Function

# Ejecutar antes de cerrar/guardar/imprimir o al abrir

Meter en Thisworkbook

Private Sub Workbook\_BeforeSave(ByVal SaveAsUI As Boolean, Cancel As Boolean)  
msgbox "Deseas guardar este archivo"  
End Sub  
  
Sub Workbook\_BeforeClose(Cancel As Boolean)  
msgbox "Deseas cerrar este archivo"  
End Sub  
  
Private Sub Workbook\_BeforePrint(Cancel As Boolean)  
msgbox "Deseas imprimir este archivo"  
End Sub

Private Sub Workbook\_Open()

Call Comprobacion2

End Sub

Borrar el caché de archivo abiertos:

Dim DataObj As New MSForms.DataObject 'empty the clipboard

       DataObj.SetText ""

       DataObj.PutInClipboard

## Obtener el directorio de un archivo

MsgBox "Select the file to import"

Archivo = Application.GetOpenFilename

If Archivo = False Then

Exit Sub

End If

Posicion = InStrRev(Archivo, "\")

Interruptor = 0

Posicion2 = InStrRev(Right(Archivo, Len(Archivo) - (Posicion)), ".")

Cells(4, 65) = Left(Archivo, Posicion)

# Scraping

## Queries

### Importar todas las tablas de un página web

Sub Import()

Dim Ws As Worksheet

Dim qt As QueryTable

Dim URL As String

URL = "https://www.wiseowl.co.uk/courses/"

'Mantener cada tabla que se genere en una pestaña diferente

Set Ws = Worksheets.Add

Set qt = Ws.QueryTables.Add(Connection:="URL;" & URL, Destination:=Range("A1"))

With qt

.RefreshOnFileOpen = True 'Refrescar cada vez que abrimos el archivo

.RefreshPeriod = 0 'Cada cuanto refrescamos

.Name = "Course" 'Añadir un nombre para posteriormente referirnos a ella

'Formato en el que imprime el texto

'.WebFormatting = xlWebFormattingAll 'el de la web

'.WebFormatting = xlWebFormattingNone 'sin formato (quita los hipervínculos)

.WebFormatting = xlWebFormattingRTF 'otro (quita los hipervínculos)

'Que cuadros se pueden seleccionar

'.WebSelectionType = xlAllTables 'Sólamente las tablas

'.WebSelectionType = xlEntirePage 'Toda la página

.WebSelectionType = xlSpecifiedTables 'Tablas específicas

.WebTables = "2,3" 'Las tablas específicas tienen que tener este comando y poniendo el número de las tablas entre comillas y separado por comas. Se puede poner como variable ‘i’ para meterlo en un bucle

.Refresh

End With

End Sub

### Actualizar una tabla ya creada

Sub ImportSrates()

Dim qt As QueryTable

Dim URL As String

URL = "https://www.x-rates.com/table?from=" & Ws.Cells(1, 2).Value & "&amount=" & Ws.Cells(2, 2).Value

'Mantener cada tabla que se genere en una pestaña diferente

Set qt = Ws.QueryTables("XRates") 'No hace falta añadir uno nuevo, se actualiza uno actiguo

'https://www.x-rates.com/table/?from=GBP&amount=1

With qt

.Connection = "URL;" & URL

.Refresh

End With

End Sub

## Navegar por una web

Primero se tiene que obtener el código de la página web. Hay dos opciones, abrir el navegador o pedir el código mediante una acción GET (Abrir el navegador solo es necesario cuando vamos a tener que pulsar botones o rellenar cuadros, por ejemplo para hacer log in)

IMPORTANTE: se puede jugar con el nombre de la página web para evitar tener que rellenar casillas o pulsar valores de manera innecesaria

Cuando se recoger un conjunto de lementos en uno se puede indicar cual se quiere con un paréntesis al final. Se empieza por el (0)

Set HTMLTable = HTMLDoc.getElementsByTagName("table") (0) ‘Se coge la primera tabla de todas las que haya

### Abrir el navegador

Dim IE As New SHDocVw.InternetExplorer

Dim HTMLTable As MSHTML.IHTMLElement

Dim HTMLTables As MSHTML.IHTMLElementCollection

IE.Visible = True

IE.navigate "https://www.whoscored.com/Regions/108/Tournaments/5/Italy-Serie-A"

'Antes de seguir con el código hay que esperar a que Internet Explorer esté abierto

Do While IE.readyState <> READYSTATE\_COMPLETE

Loop

Dim IE As New SHDocVw.InternetExplorer

IE.Visible = True

IE.navigate "en.wikipedia.org/wiki/Main\_Page" ' en internet explorer no necesitamos http

Antes de seguir con el código hay que esperar a que Internet Explorer esté abierto

Do While IE.readyState <> READYSTATE\_COMPLETE

Loop

#### Hacer log in

Sub Loggin()

Dim IE As New SHDocVw.InternetExplorer

Dim HTMLTable As MSHTML.IHTMLElement

Dim HTMLTables As MSHTML.IHTMLElementCollection

IE.Visible = True

IE.navigate "https://www.whoscored.com/Regions/108/Tournaments/5/Italy-Serie-A"

'Antes de seguir con el código hay que esperar a que Internet Explorer esté abierto

Do While IE.readyState <> READYSTATE\_COMPLETE

Loop

Dim IE As New SHDocVw.InternetExplorer

IE.Visible = True

IE.navigate "en.wikipedia.org/wiki/Main\_Page" ' en internet explorer no necesitamos http

'Antes de seguir con el código hay que esperar a que Internet Explorer esté abierto

Application.Wait Now + TimeValue("00:00:05")

EnterCredential User, Password

Do While IE.readyState <> READYSTATE\_COMPLETE

Loop

Set HTMLDoc = IE.document

End Sub

Sub EnterCredential(Username As String, Password As String)

Dim SegurityWindow As Object

sert SegurityWindow = CreateObject("WScrip.Shell")

With SegurityWindow

.AppActivate "Windows Segurity"

.SendKeys Username

.SendKeys "{TAB}"

.SendKeys Password

.SendKeys "{ENTER}"

End With

End Sub

### Directamente mediante ‘GET’ (es mucho más rápido)

Dim XMLPage As New MSXML2.XMLHTTP60

Dim HTMLDoc As New MSHTML.HTMLDocument

Dim HTMLTable As MSHTML.IHTMLElement

Dim HTMLTables As MSHTML.IHTMLElementCollection

XMLPage.Open "GET", "https://x-rates.com/table/?from=GBP&amount=3", False

'Cuidado, puede ser "http" o "https"

XMLPage.send

If XMLPage.Status <> 200 Then '200=todo OK

MsgBox XMLPage.Status & " - " & XMLPage.statusText

Exit Sub

End If

HTMLDoc.body.innerHTML = XMLPage.responseText

#### Hacer log in

Sub LogginXMLHTTP()

Dim XMLPage As New MSXML2.XMLHTTP60

XMLPage.Open "GET", "https://x-rates.com/table/?from=GBP&amount=3", False, User, Password

'Cuidado, puede ser "http" o "https"

XMLPage.send

If XMLPage.Status <> 200 Then '200=todo OK

MsgBox XMLPage.Status & " - " & XMLPage.statusText

Exit Sub

End If

HTMLDoc.body.innerHTML = XMLPage.responseText

End Sub

### Búsqueda de la tabla

Se afina la búsqueda hasta que solamente queda la tabla que queremos

'Ver cuantas tablas hay e la web

Set HTMLTables = HTMLDoc.getElementsByTagName("table")

Debug.Print HTMLTables.Length

Set HTMLDoc = IE.document

Set HTMLUl = HTMLDoc.getElementById("tournament-tables-18873-options")

Set HTMLButtons = HTMLUl.getElementsByTagName("a")

### Sacar los datos de la tabla

For Each Section In HTMLTable.Children

For Each HTMLRow In Section.Children 'th' es table header y 'tr' table rows

'For Each HTMLCell In HTMLRow.getElementsByTagName("td") 'td' es cada una de las celdas

For Each HTMLCell In HTMLRow.Children 'Otra forma de hacerlo (y aparecen los títulos de las tablas)

Cells(Row, Column) = HTMLCell.innerText

Column = Column + 1

Next HTMLCell

Row = Row + 1

Column = 1

Next HTMLRow

Next Section

### Interaccionar con la página

* Para clicar:

For Each HTMLA In HTMLAs

Debug.Print HTMLA.innerText, HTMLA.href

HTMLA.Click

Next HTMLA

* Para rellanar un campo

Set HTMLInput = HTMLDoc.getElementById("amount")

HTMLInput.Value = 5

Set HTMLInput = HTMLDoc.getElementById("from")

HTMLInput.Value = "GBP"

Set HTMLInput = HTMLDoc.getElementById("to")

HTMLInput.Value = "USD"

# Access

‘Los archivos tienen que estar en el mismo directorio

## Pasar de Excel a Access

Sub CopiaDatos()

Dim fila As Long, uf As Long, conta As Long

Dim cn As ADODB.Connection, rs As ADODB.Recordset

On Error Resume Next

Application.ScreenUpdating = False

Application.DisplayAlerts = False

Set a = ActiveSheet

Set cn = New ADODB.Connection

Set rs = New ADODB.Recordset

cn.Open "Provider=Microsoft.ACE.OLEDB.12.0; " & "data source=" & ThisWorkbook.Path & "\171 ProgramarExcel.accdb;"

rs.Open "Clientes", cn, adOpenKeyset, adLockOptimistic, adCmdTable

fila = 2

conta = 0

While a.Cells(fila, "A") <> Empty

With rs

.AddNew

.Fields("Id") = Cells(fila, "A")

.Fields("Nombre") = Cells(fila, "B")

.Fields("Telefono") = Cells(fila, "C")

.Fields("Direccion") = Cells(fila, "D")

.Fields("Mail") = Cells(fila, "E")

.Fields("Credito") = Cells(fila, "F")

.Update

End With

conta = conta + 1

fila = fila + 1

Wend

rs.Close

Set rs = Nothing

cn.Close

Set cn = Nothing

MsgBox ("Se procesaron " & conta & " registros con éxito, se omitieron duplicados"), vbInformation, "AVISO"

Application.ScreenUpdating = True

Application.DisplayAlerts = True

End Sub