

Contenido

9.2. CÓDIGO FUENTE DE LA HERRAMIENTA	2
9.2.1. MÓDULO “ThisWorkbook”	2
9.2.2. Formularios.....	6
9.2.3. Módulo: <i>AjustesTabla</i>	38
9.2.4. <i>Bordes_Tabla</i>	40
9.2.5. <i>CalculoResumen</i>	51
9.2.6. <i>ConfigurarBotones</i>	54
9.2.7. <i>ContadorBlanco</i>	56
9.2.8. <i>CopiarResultados</i>	56
9.2.9. <i>Filtros</i>	62
9.2.10. <i>Formato_encabezados</i>	62
9.2.11. <i>FormatoCondicional</i>	73
9.2.12. <i>FormatoTexto</i>	80
9.2.13. <i>GuardarCopia</i>	82
9.2.14. <i>Main_Copiar_Elementos</i>	82
9.2.15. <i>Modulo_Seguridad</i>	90
9.2.16. <i>Ocultar_Mostrar_Descripcion</i>	91
9.2.17. <i>Orden_3_Niveles</i>	92
9.2.18. <i>TablaValoracionEtapas</i>	97
9.2.19. <i>ValidarDatos</i>	104

9.2. CÓDIGO FUENTE DE LA HERRAMIENTA

En este anexo se encuentra todo el código realizado en Visual Basic 6.0. para elaborar la herramienta de evaluación de impactos sociales en proyecto. Está debidamente comentado y explicado aquellos puntos más complejos.

9.2.1. MÓDULO “ThisWorkbook”

En este primer apartado se encuentra el código perteneciente al módulo “ThisWorkbook”, es decir todos los procedimientos que son activados mediante eventos. Como por ejemplo la apertura o cierre de la herramienta.

La siguiente subrutina se encarga de calcular el tiempo que ha tardado el usuario en realizar la evaluación.

```
*****  
*****
```

‘Subrutina que se ejecuta antes de cerrar la herramienta

```
*****  
*****
```

Sub Workbook_BeforeClose(Cancel As Boolean)

‘Esta subrutina se encarga de calcular y mostrar el tiempo empleado en la evaluación

Worksheets("INDICADORES").Range("M2").Value = Format(Now, "hh:mm:ss")

Valor1 = Worksheets("INDICADORES").Range("M2").Value

valor0 = Worksheets("INDICADORES").Range("L2").Value

Resta = Format(valor0 - Valor1, "hh:mm:ss")

MsgBox ("El tiempo utilizado para la evaluación ha sido: " & Resta)

ActiveWorkbook.Save ‘Guarda el libro activo

End Sub

La siguiente rutina se activa cuando se abre el libro, es decir la herramienta, con ello se consigue poder tener seguridad a la hora de trabajar siempre y cuando estén habilitas las macros como se ha dicho a lo largo de la memoria.

```
*****  
*****
```

‘Subrutina que se ejecuta nada más abrir la herramienta

Sub Workbook_Open()

Application.ScreenUpdating = False ' Desactivamos la actividad visual

On Error Resume Next 'Tratamiento de errores

'Limpiamos los contadores de tiempo que son "invisibles" para los usuarios

Worksheets("INDICADORES").Range("L2").Value = ""

Worksheets("INDICADORES").Range("M2").Value = ""

'Se da valor al contador con la fecha de inicio de la evaluación

Worksheets("INDICADORES").Range("L2").Value = Format(Now, "hh:mm:ss")

ActiveWorkbook.Worksheets("AYUDA").Activate 'Activa la hoja AYUDA

Load Bienvenida 'Carga el formulario de bienvenida.

'Se comienza a preguntar al usuario lo que quiere hacer

Respuesta = MsgBox("Desea comenzar a utilizar la herramienta", vbYesNo _

+ vbQuestion, "Mensaje inicial") 'Mensaje que se muestra

If Respuesta = vbYes Then

 respuesta1 = MsgBox("¿Realizar una nueva evaluación?", vbYesNo _

+ vbQuestion, "Nueva evaluación") 'Mensaje que se muestra

If respuesta1 = vbYes Then

 Bienvenida.Show 'Muestra el formulario de bienvenida.

 Fila_Ini = 5

 Fila = Fila_Ini

 COLUMNA_Indicadores = 6

 ActiveWorkbook.Worksheets("INDICADORES").Activate

 'Poner la celda en blanco

 Range("K3").Select

 With Selection.Font

 .ThemeColor = xlThemeColorDark1

 .TintAndShade = 0

 End With

 Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")

 Fila = Fila + 1

 Loop

 Fila_Max = Fila - 1

 'Se llama a las funciones para dar formato a la tabla, ordenar indicadores de la base etc.

```

Call bordes_tabla_indicadores(Fila_Max)
Call Formato_Encabezado_Indicadores
Call Formateo_Hoja
Call Contador_Blanco
Worksheets("INDICADORES").Range("K2").Value = (Fila_Max - Fila_Ini + 1)
'Mostramos en esa celda el número de indicadores disponible
Call Formato_Condicional_Indicadores(Fila_Max) 'Aplicamos el formato condicional a la
columna de evaluar
'Limpia las hojas RESUMEN, EX-ANTE, IN-ITINERE, EX-POST
Call Orden_Tres_Niveles_Indicadores(Fila_Max)
'Se borra la tabla resuemen del proyecto anterior.
ActiveWorkbook.Worksheets("RESUMEN").Activate
Range("P5:U7").Select
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = 0
        .PatternTintAndShade = 0
    End With
Range(Cells(5, 1), Cells(Fila_Max, 13)).Select
    Selection.ClearContents
Range("P4:U8").Select
    Selection.ClearContents
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    Selection.Borders(xlEdgeBottom).LineStyle = xlNone
    Selection.Borders(xlEdgeRight).LineStyle = xlNone
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Range("H5").Select
ActiveWorkbook.Worksheets("EX-ANTE").Activate
Range(Cells(5, 1), Cells(Fila_Max, 12)).Select

```

```

        Selection.ClearContents
Range("G5").Select
ActiveWorkbook.Worksheets("IN-ITINERE").Activate
    Range(Cells(5, 1), Cells(Fila_Max, 12)).Select
        Selection.ClearContents
Range("G5").Select
ActiveWorkbook.Worksheets("EX-POST").Activate
Range(Cells(5, 1), Cells(Fila_Max, 12)).Select
    Selection.ClearContents
Range("G5").Select
ActiveWorkbook.Worksheets("INDICADORES").Activate
'Se oculta el contador.
Call Contador_Blanco
Range("J5").Select
ActiveWorkbook.Worksheets("AYUDA").Activate 'Activa la hoja AYUDA
Call Seguridad_Nueva ' Llamamos a la función de nueva contraseña
    Range("G5").Select
Else
    'Si la respuesta es que no desea una evaluación nueva, se llama a la función Seguridad.
    Call Seguridad
ActiveWorkbook.Worksheets("INDICADORES").Activate
'Siempre se debe "Ocultar" el contador.
Call Contador_Blanco
Range("G5").Select
End If
Else
ActiveWorkbook.Save 'Guarda el libro activo
Application.Quit 'Cierra la aplicación
End If
Application.ScreenUpdating = True
End Sub

```

9.2.2. Formularios

A continuación, se muestran los dos formularios realizados.

9.2.2.1. Bienvenida

Este primer formulario, Figura 1, da la bienvenida al usuario cuando va a realizar una nueva evaluación. El código para llamar a este formulario se ha podido ver en la subrutina anterior, lo único que se hace es “cargar” el formulario y mostrarle. Este mensaje se ha decidido realizar mediante un formulario para poder darle color y hacerle más atractivo, aunque también se podría haber realizado mediante un “MsgBox” (Caja de mensaje)

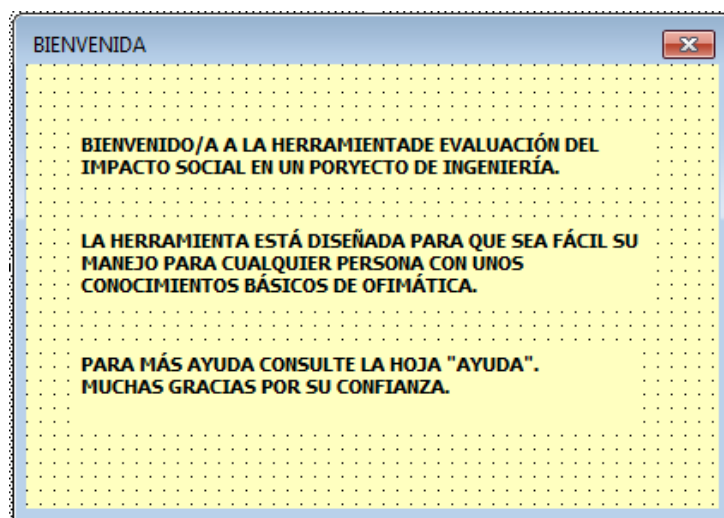


Figura 1. Formulario “Bienvenida”.

9.2.2.2. Modificador de indicadores

En la Figura 2, se muestra el formulario creado para añadir y/o modificar los indicadores de la base de datos. Es complejo de programar como se ha visto en el punto 5 de la memoria, como se puede observar además por la longitud del código.

Este formulario tiene un total de catorce botones, los cuales se activan/desactivan y se muestran/ocultan en función de lo que desee hacer el usuario. Para ello se ha de programar una serie de acciones teniendo en cuenta eventos. Por ejemplo, para añadir color a los botones, al pasar por encima del botón, y ponerle en negro cuando el ratón no este encima. Todo esto, se ha programado utilizando los eventos que nos ofrece Visual Basic.

MODIFICADOR DE INDICADORES

MODIFICACIÓN DE LOS INDICADORES:

FAMILIA

CATEGORIA

INDICADOR

DESCRIPCIÓN

Figura 2. Formulario "Modificador de indicadores"

Como se puede ver en el nombre de las subrutinas, que forman parte del módulo, están compuestas por el nombre del objeto al que hacer referencia, en este caso UserForm, una barra baja y la acción con la que se activa la rutina, en este caso Activate. Como se puede ver en todas las subrutinas de este módulo siguen la misma estructura.

A continuación, se va hacer un breve comentario sobre los objetos utilizados:

- UserForm: Formulario
- Botones
 - CmdLimpiar: Cuando se pulsa se borran todos los cuadros y combos de texto
 - CmdCancelar: Cierra el formulario.
 - CmdAgregarIndicador: Añade el nuevo indicador a la base de datos siempre que este todo relleno correctamente.

- CmdModificarIndicador: Modifica el indicador seleccionado, siempre que este todo correctamente.
- CmdModificarIndicador2: Activa los botones, combos y cuadros de texto para poder modificar los indicadores.
- CmdNuevoIndicador: Activa los botones, combos y cuadros de texto para poder añadir los indicadores.
- CmdAgregarCategoria: Añade la nueva categoria a la base de datos siempre que este todo relleno correctamente.
- CmdModificarCategoria: Modifica la categoria seleccionado, siempre que este todo correctamente.
- CmdModificarCategoria2: Activa los botones, combos y cuadros de texto para poder modificar las categorías.
- CmdNuevaCategoria: Activa los botones, combos y cuadros de texto para añadir las nuevas categorias con los indicadores.
- CmdAgregarFamilia: Añade la nueva familia a la base de datos siempre que este todo relleno correctamente.
- CmdModificarFamilia: Modifica la familia seleccionado, siempre que este todo correctamente.
- CmdModificarFamilia2: Activa los botones, combos y cuadros de texto para poder modificar las familias.
- CmdNuevaFamilia: Activa los botones, combos y cuadros de texto para poder añadir las nuevas familias con los indicadores.
- Cuadros de texto
 - TxtFamilia: Se introduce la familia
 - TxtCategoria: Se introduce la categoria
 - TxtIndicador: Se introduce el indicador
 - TxtDescripcion: Se introduce la descripción
- Combos de texto
 - ComboBox_Familia: Se cargan todas las familias disponibles
 - ComboBox_Categoria: Se cargan todas las categorias disponibles
 - ComboBox_Indicador: Se cargan todos los indicadores disponibles

El código que se muestra en dividido por objetos, para hacer más fácil su comprensión.

```
'*****
*****
```

'User Form

```
'*****
*****
```

```
Private Sub UserForm_Activate()
```

```
    'Subrutina que se ejecuta cuando se activa el formulario
```

```
    Application.ScreenUpdating = False
```


'Tratamiento de errores, se hay error, pasamos a la siguiente instrucción

On Error Resume Next

'Desactivamos los botones de añadir, modificar, limpiar o cancelar

CmdLimpiar.Visible = False

CmdLimpiar.Enabled = False

CmdCancelar.Visible = False

CmdCancelar.Enabled = False

CmdAgregarIndicador.Visible = False

CmdAgregarIndicador.Enabled = False

CmdModificarIndicador.Visible = False

CmdModificarIndicador.Enabled = False

CmdAgregarFamilia.Visible = False

CmdAgregarFamilia.Enabled = False

CmdAgregarCategoria.Visible = False

CmdAgregarCategoria.Enabled = False

CmdAgregarIndicador.Visible = False

CmdAgregarIndicador.Enabled = False

CmdModificarFamilia.Visible = False

CmdModificarFamilia.Enabled = False

CmdModificarCategoria.Visible = False

CmdModificarCategoria.Enabled = False

CmdModificarIndicador.Visible = False

CmdModificarIndicador.Enabled = False

'Vaciamos los textbox y combobox

TxtFamilia.Value = ""

TxtCategoria.Value = ""

TxtIndicador.Value = ""

ComboBox_Familia = ""

ComboBox_Categoria = ""

ComboBox_Indicador = ""

TxtDescripcion.Value = ""

End Sub

En la siguiente subrutina del módulo del formulario, se puede ver que en este caso el evento con el que trabaja es el movimiento del ratón. Esto se ha añadido para poder dar color a los botones visibles.

```
Private Sub UserForm_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)
```

'Subrutina que cambia los colores de los botones, al mover el raton por encima del formulario

```
    CmdCancelar.ForeColor = &H80000012  
    CmdLimpiar.ForeColor = &H80000012  
    CmdModificarIndicador.ForeColor = &H80000012  
    CmdNuevaFamilia.ForeColor = &H80000012  
    CmdNuevaCategoria.ForeColor = &H80000012  
    CmdNuevoIndicador.ForeColor = &H80000012  
    CmdModificarFamilia2.ForeColor = &H80000012  
    CmdModificarCategoria2.ForeColor = &H80000012  
    CmdModificarIndicador2.ForeColor = &H80000012  
    CmdModificarFamilia.ForeColor = &H80000012  
    CmdModificarCategoria.ForeColor = &H80000012  
    CmdModificarIndicador.ForeColor = &H80000012  
    CmdAgregarFamilia.ForeColor = &H80000012  
    CmdAgregarCategoria.ForeColor = &H80000012  
    CmdAgregarIndicador.ForeColor = &H80000012
```

```
End Sub
```

```
'*****
```

```
*****
```

```
'CmdAgregarFamilia
```

```
'*****
```

```
*****
```

```
Private Sub CmdAgregarFamilia_Click()
```

'Subrutina para añadir nuevos indicadores la base de datos.

'Desativamos a las "animaciones"

```
Application.ScreenUpdating = False
```

'Tratamiento de errores, se hay error, pasamos a la siguiente instrucción

On Error Resume Next

'Variables a usar

COLUMNA_FECHA = 8

COLUMNA_Descripcion = 7

COLUMNA_Indicadores = 6

COLUMNA_Subcategorias = 5

COLUMNA_Familias = 4

Fila = 5

'Cargamos el valor de los Textbox en variables para hacer un mejor uso

Familia = TxtFamilia.Value

Categoria = TxtCategoria.Value

Indicador = TxtIndicador.Value

Descripcion = TxtDescripcion.Value

'Activamos la hoja INDICADORES para poder trabajar con ella sin problema

ActiveWorkbook.Worksheets("INDICADORES").Activate

'Tenemos que buscar el valor de la ultima fila usamos el bucle do while o la función usedRange

Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")

Fila = Fila + 1

Loop

'Ultimafila = ActiveSheet.UsedRange.Row - 1 + ActiveSheet.UsedRange.Rows.Count

'Lo malo de esta función esque coge la ultima fila usada

Ultimafila = Fila

'Con este If lo que hacemos es evitar que se añadan nuevos indicadores con campos incompletos

If Familia <> "" And Categoria <> "" And Indicador <> "" And Descripcion <> "" Then

'Cargamos los valores de familia, categoria, indicador, descripcion en el sitio correcto

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Familias) = Familia

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Subcategorias) = Categoria

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Indicadores) = Indicador

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Descripcion) = Descripcion

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_FECHA) = Format(Now, "dd/mm/yyyy")

'Cargamos el número de indicadores

Worksheets("INDICADORES").Range("K2").Value = (Ultimafila - 4)

'Eliminamos los valores de los textbox y combobox

TxtFamilia.SetFocus 'Pone el raton en la primera entrada

```

'Formulario_N_Indicador.Hide 'Oculta el formulario

'ComboBox_Familia.Value = ""

'ComboBox_Categoria.Value = ""

'ComboBox_Indicador.Value = ""

TxtFamilia.Value = ""

TxtCategoria.Value = ""

TxtIndicador.Value = ""

TxtDescripcion.Value = ""

'TxtFamilia.SetFocus 'Pone el raton en la primera entrada
Call Orden_Tres_Niveles_Indicadores(Ultimafila)

'Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"

Fila = 5

Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")

    Fila = Fila + 1

Loop

'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select

ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila - 1, 10))

'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select

Else

'Si no estan todos los valores cargados nos saca un mensaje de alerta

    Q = MsgBox("POR FAVOR, RELLENE TODOS LOS CAMPOS CORRECTAMENTE", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")

End If

End Sub

Private Sub CmdAgregarFamilia_MouseDown(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
As Single, ByVal Y As Single)

'Cambiamos el color cuando nos movemos por encima del boton

    CmdAgregarFamilia.ForeColor = &HC00000

End Sub

'*****

*****

' CmdAgregarCategoria

'*****

*****

```

```

Private Sub CmdAgregarCategoria_Click()

    'Subrutina para añadir nuevos indicadores a la base de datos.

    'Desativamos a las "animaciones"
    Application.ScreenUpdating = False

    'Tratamiento de errores, se hay error, pasamos a la siguiente instrucción
    On Error Resume Next

    'Variables a usar
    COLUMNA_FECHA = 8
    COLUMNA_Descripcion = 7
    COLUMNA_Indicadores = 6
    COLUMNA_Subcategorias = 5
    COLUMNA_Familias = 4
    Fila = 5

    'Cargamos el valor de los Combobox y Textbox en variables para hacer un mejor uso
    Familia = ComboBox_Familia.Value
    Categoria = TxtCategoria.Value
    Indicador = TxtIndicador.Value
    Descripcion = TxtDescripcion.Value 'Activamos la hoja INDICADORES para poder trabajar con ella
    sin problema

    ActiveWorkbook.Worksheets("INDICADORES").Activate

    'Tenemos que buscar el valor de la ultima fila.
    Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")
        Fila = Fila + 1
    Loop

    Ultimafila = Fila

    'Con este If lo que hacemos es evitar que se añadan nuevos indicadores con campos incompletos
    If Familia <> "" And Categoria <> "" And Indicador <> "" And Descripcion <> "" Then
        'Cargamos los valores de familia, categoria, indicador, descripcion en el sitio correcto
        Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Familias) = Familia
        Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Subcategorias) = Categoria
        Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Indicadores) = Indicador
        Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Descripcion) = Descripcion
        Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_FECHA) = Format(Now, "dd/mm/yyyy")

        'Cargamos el número de indicadores, en el contador
        Worksheets("INDICADORES").Range("K2").Value = (Ultimafila - 4)
    End If
End Sub

```

```

'Vaciamos los ComboBox y TextBox
'ComboBox_Familia.Value = ""
'ComboBox_Categoria.Value = ""
'ComboBox_Indicador.Value = ""
TxtFamilia.Value = ""
TxtCategoria.Value = ""
TxtIndicador.Value = ""
TxtDescripcion.Value = ""

'TxtFamilia.SetFocus 'Pone el raton en la primera entrada
Call Orden_Tres_Niveles_Indicadores(Ultimafila)
Fila = 5
'Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"
Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")
    Fila = Fila + 1
Loop
'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select
ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila - 1, 10))
'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select
Else
'Si no estan todos los valores cargados nos saca un mensaje de alerta
    Q = MsgBox("POR FAVOR, RELLENE TODOS LOS CAMPOS CORRECTAMENTE", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
End If

End Sub

Private Sub CmdAgregaCategoria_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal
X As Single, ByVal Y As Single)
'Cambiamos el color cuando nos movemos por encima del boton
    CmdAgregaCategoria.ForeColor = &HC000000
End Sub

```

'*****

'CmdAgregarIndicador

'*****

Private Sub CmdAgregarIndicador_Click()

'Subrutina para añadir nuevos indicadores a la base de datos.

'Desativamos a las "animaciones"

Application.ScreenUpdating = False

'Tratamiento de errores, se hay error, pasamos a la siguiente instrucción

On Error Resume Next

COLUMNA_FECHA = 8

COLUMNA_Descripcion = 7

COLUMNA_Indicadores = 6

COLUMNA_Categorias = 5

COLUMNA_Familias = 4

Fila = 5

Familia = ComboBox_Familia.Value

Categoria = ComboBox_Categoria.Value

Indicador = TxtIndicador.Value

Descripcion = TxtDescripcion.Value

ActiveWorkbook.Worksheets("INDICADORES").Activate

'Tenemos que buscar el valor de la ultima fila.

Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")

Fila = Fila + 1

Loop

Ultimafila = Fila

'Con este If lo que hacemos es evitar que se añadan nuevos indicadores con campos incompletos

If Familia <> "" And Categoria <> "" And Indicador <> "" And Descripcion <> "" Then

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Familias) = Familia

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Categorias) = Categoria

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Indicadores) = Indicador

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Descripcion) = Descripcion

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_FECHA) = Format(Now, "dd/mm/yyyy")

```

Worksheets("INDICADORES").Range("K2").Value = (Ultimafila - 4)

'ComboBox_Familia.Value = ""

'ComboBox_Categoria.Value = ""

'ComboBox_Indicador.Value = ""

TxtFamilia.Value = ""

TxtCategoria.Value = ""

TxtIndicador.Value = ""

TxtDescripcion.Value = ""

'TxtFamilia.SetFocus 'Pone el raton en la primera entrada
Call Orden_Tres_Niveles_Indicadores(Ultimafila)

'Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"

Fila = 5

Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")

    Fila = Fila + 1

Loop

'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select

ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila - 1, 10))

'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select

Else

    Q = MsgBox("POR FAVOR, RELLENE TODOS LOS CAMPOS CORRECTAMENTE", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")

End If

End Sub

Private Sub CmdAgregarIndicador_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal
X As Single, ByVal Y As Single)

    CmdAgregarIndicador.ForeColor = &HC00000

End Sub

'*****

' CmdModificarFamilia

'*****

Private Sub CmdModificarFamilia_Click()

'Subrutina para la modificación del los indicadores de las lista

'Subrutina para añadir nuevos indicadores a la base de datos.

'Desativamos a las "animaciones"

```



```

Application.ScreenUpdating = False

'Tratamiento de errores, se hay error, pasamos a la siguiente instrucción
On Error Resume Next

COLUMNNA_FECHA = 8
COLUMNNA_Descripcion = 7
COLUMNNA_Indicadores = 6
COLUMNNA_Categorias = 5
COLUMNNA_Familias = 4

Fila = 5

Familia = ComboBox_Familia.Value
Categoria = TxtCategoria.Value
Indicador = TxtIndicador.Value
Descripcion = TxtDescripcion.Value

ActiveWorkbook.Worksheets("INDICADORES").Activate

If Familia <> "" And Categoria <> "" And Indicador <> "" And Descripcion <> "" Then

'Con este If lo que hacemos es evitar que se añadan nuevos indicadores con campos incompletos

'Hay que buscar el indicador correcto.

Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNNA_Categorias).Value <> "")
    Fila = Fila + 1
Loop

Ultimafila = Fila

Worksheets("INDICADORES").Cells(Ultimafila, COLUMNNA_Familias) = Familia
Worksheets("INDICADORES").Cells(Ultimafila, COLUMNNA_Categorias) = Categoria
Worksheets("INDICADORES").Cells(Ultimafila, COLUMNNA_Indicadores) = Indicador
Worksheets("INDICADORES").Cells(Ultimafila, COLUMNNA_Descripcion) = Descripcion
Worksheets("INDICADORES").Cells(Ultimafila, COLUMNNA_FECHA) = Format(Now, "dd/mm/yyyy")
Worksheets("INDICADORES").Range("K2").Value = (Ultimafila - 4)

'ComboBox_Familia.Value = ""
'ComboBox_Categoria.Value = ""
'ComboBox_Indicador.Value = ""

TxtFamilia.Value = ""
TxtCategoria.Value = ""
TxtIndicador.Value = ""
TxtDescripcion.Value = ""

'TxtFamilia.SetFocus 'Pone el raton en la primera entrada

```

```

Call Orden_Tres_Niveles_Indicadores(Ultimafila)

'Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"

Fila = 5

Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")

    Fila = Fila + 1

Loop

'Range("Tabla_Indicadores[[#Headers],[FAMILIA ]]").Select

ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila - 1, 10))

'Range("Tabla_Indicadores[[#Headers],[FAMILIA ]]").Select

Else

    Q = MsgBox("POR FAVOR, RELLENE TODOS LOS CAMPOS CORRECTAMENTE", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")

End If

End Sub

Private Sub CmdModificarFamilia_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal
X As Single, ByVal Y As Single)

'Cambiamos el color cuando nos movemos por encima del boton

    CmdModificarFamilia.ForeColor = &H40C0&

End Sub

'*****

*****

'CmdModificarCategoria

'*****

*****

Private Sub CmdModificarCategoria_Click()

'Subrutina para la modificación del los indicadores de las lista

'Subrutina para añadir nuevos indicadores a la base de datos.

Application.ScreenUpdating = False

On Error Resume Next

COLUMNA_FECHA = 8

COLUMNA_Descripcion = 7

COLUMNA_Indicadores = 6

COLUMNA_Categorias = 5

COLUMNA_Familias = 4

Fila = 5

```

```

Familia = ComboBox_Familia.Value
Categoria = ComboBox_Categoria.Value
Indicador = TxtIndicador.Value
Descripcion = TxtDescripcion.Value
ActiveWorkbook.Worksheets("INDICADORES").Activate
'Con este If lo que hacemos es evitar que se añadan nuevos indicadores con campos incompletos
If Familia <> "" And Categoria <> "" And Indicador <> "" And Descripcion <> "" Then
    'Hay que buscar el indicador correcto.
    Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Categorias).Value <> "")
        Fila = Fila + 1
    Loop
    Ultimafila = Fila

    Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Familias) = Familia
    Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Categorias) = Categoria
    Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Indicadores) = Indicador
    Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_Descripcion) = Descripcion
    Worksheets("INDICADORES").Cells(Ultimafila, COLUMNA_FECHA) = Format(Now, "dd/mm/yyyy")
    Worksheets("INDICADORES").Range("K2").Value = (Ultimafila - 4)

    'ComboBox_Familia.Value = ""
    'ComboBox_Categoria.Value = ""
    'ComboBox_Indicador.Value = ""
    TxtFamilia.Value = ""
    TxtCategoria.Value = ""
    TxtIndicador.Value = ""
    TxtDescripcion.Value = ""

    'TxtFamilia.SetFocus 'Pone el raton en la primera entrada
    Call Orden_Tres_Niveles_Indicadores(Ultimafila)
    'Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"
    Fila = 5
    Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")
        Fila = Fila + 1
    Loop
    'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select
    ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila - 1, 10))

```

```

        'Range("Tabla_Indicadores[[#Headers],[FAMILIA ]]").Select
    Else
        Q = MsgBox("POR FAVOR, RELLENE TODOS LOS CAMPOS CORRECTAMENTE", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
    End If
End Sub

Private Sub CmdModificarCategoria_MouseDown(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)

'Cambiamos el color cuando nos movemos por encima del boton

    CmdModificarCategoria.ForeColor = &H40C0&
End Sub

'*****
'*****

'CmdModificarIndicador

'*****
'*****

Private Sub CmdModificarIndicador_Click()

'Subrutina para la modificación del los indicadores de las lista

'Subrutina para añadir nuevos indicadores a la base de datos.

    Application.ScreenUpdating = False

    On Error Resume Next

    COLUMNA_FECHA = 8
    COLUMNA_Descripcion = 7
    COLUMNA_Indicadores = 6
    COLUMNA_Categoria = 5
    COLUMNA_Familias = 4

    Fila = 5

    Familia = ComboBox_Familia.Value
    Categoria = ComboBox_Categoria.Value
    Indicador = ComboBox_Indicador.Value
    Descripcion = TxtDescripcion.Value

    ActiveWorkbook.Worksheets("INDICADORES").Activate

```

```

If Familia <> "" And Categoria <> "" And Indicador <> "" And Descripcion <> "" Then
'Con este If lo que hacemos es evitar que se añadan nuevos indicadores con campos incompletos
'Hay que buscar el indicador correcto.
Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> Indicador)
    Fila = Fila + 1
Loop
'MsgBox Fila
Worksheets("INDICADORES").Cells(Fila, COLUMNA_Familias) = Familia
Worksheets("INDICADORES").Cells(Fila, COLUMNA_Categoria) = Categoria
Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores) = Indicador
Worksheets("INDICADORES").Cells(Fila, COLUMNA_Descripcion) = Descripcion
Worksheets("INDICADORES").Cells(Fila, COLUMNA_FECHA) = Format(Now, "dd/mm/yyyy")
Filaseleccionada = Fila
'Cells(Fila, COLUMNA_Indicadores).Select
'ActiveCell.Offset(0, -2).Value = Familia
'ActiveCell.Offset(0, -1).Value = Categoria
'ActiveCell.Offset(0, 1).Value = Descripcion
'ActiveCell.Offset(0, 2).Value = Format(Now, "dd/mm/yyyy")
'ComboBox_Familia.Value = ""
'ComboBox_Categoria.Value = ""
'ComboBox_Indicador.Value = ""
TxtFamilia.Value = ""
TxtCategoria.Value = ""
TxtIndicador.Value = ""
TxtDescripcion.Value = ""
'TxtFamilia.SetFocus 'Pone el raton en la primera entrada
'Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"
Fila = 5
Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")
    Fila = Fila + 1
Loop
Call Orden_Tres_Niveles_Indicadores(Fila)
'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select
ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila - 1, 10))
'Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select

```

```

        Rows(Fila).Select
        Selection.Delete Shift:=xlUp
        Cells(Filaseleccionada, 4).Select
    Else
        Q = MsgBox("POR FAVOR, RELLENE TODOS LOS CAMPOS CORRECTAMENTE", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
    End If
End Sub

Private Sub CmdModificarIndicador_MouseMove(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)

    CmdModificarIndicador.ForeColor = &H40C0&

End Sub

'*****
'*****

                                CmdLimpiar

'*****
'*****

Private Sub CmdLimpiar_Click()

    Application.ScreenUpdating = False

    On Error Resume Next

    'Subrutina para limpiar los cuadros de texto

    TxtFamilia.Value = ""

    TxtCategoria.Value = ""

    TxtIndicador.Value = ""

    ComboBox_Familia = ""

    ComboBox_Categoria = ""

    ComboBox_Indicador = ""

    TxtDescripcion.Value = ""

    'TxtFamilia.SetFocus 'Pone el raton en la primera entrada

End Sub

Private Sub CmdLimpiar_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As
Single, ByVal Y As Single)

    CmdLimpiar.ForeColor = &HC000&

End Sub

'*****
'*****

```

'CmdCancelar

Private Sub CmdCancelar_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)

CmdCancelar.ForeColor = &HFF&

End Sub

Private Sub CmdCancelar_Click()

Unload Formulario_N_Indicador 'Cierra el formulario, lo "descarga"

End Sub

'CmdNuevaFamilia

Private Sub CmdNuevaFamilia_Click()

'Activamos los botones, textbox y combobox, que vamos a utilizar.

ComboBox_Familia.Enabled = False

ComboBox_Familia.Visible = False

ComboBox_Categoria.Enabled = False

ComboBox_Categoria.Visible = False

ComboBox_Indicador.Enabled = False

ComboBox_Indicador.Visible = False

'Vaciamos los textbox y combobox

ComboBox_Familia = ""

ComboBox_Categoria = ""

ComboBox_Indicador = ""

TxtFamilia = ""

TxtCategoria = ""

TxtIndicador = ""

Descripcion = ""

TxtFamilia.Visible = True

TxtFamilia.Enabled = True

TxtCategoria.Visible = True

```

    TxtCategoria.Enabled = True
    TxtIndicador.Visible = True
    TxtIndicador.Enabled = True
    TxtDescripcion.Visible = True
    TxtDescripcion.Enabled = True
    CmdAgregarFamilia.Visible = True
    CmdAgregarFamilia.Enabled = True
    CmdAgregarCategoria.Visible = False
    CmdAgregarCategoria.Enabled = False
    CmdAgregarIndicador.Visible = False
    CmdAgregarIndicador.Enabled = False
    CmdModificarFamilia.Visible = False
    CmdModificarFamilia.Enabled = False
    CmdModificarCategoria.Visible = False
    CmdModificarCategoria.Enabled = False
    CmdModificarIndicador.Visible = False
    CmdModificarIndicador.Enabled = False
    'Estos dos botones siempre deben estar.
    CmdLimpiar.Visible = True
    CmdLimpiar.Enabled = True
    CmdCancelar.Visible = True
    CmdCancelar.Enabled = True
End Sub

Private Sub CmdNuevaFamilia_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
As Single, ByVal Y As Single)
    CmdNuevaFamilia.ForeColor = &HC00000
End Sub

'*****
                *****'CmdNuevaCategoria
'*****
*****

Private Sub CmdNuevaCategoria_Click()
    'Activamos los botones, textbox y combobox, que vamos a utilizar.
    ComboBox_Familia.Enabled = True

```



```

ComboBox_Familia.Visible = True
ComboBox_Categoria.Enabled = True
ComboBox_Categoria.Visible = True
ComboBox_Indicador.Enabled = False
ComboBox_Indicador.Visible = False
'Vaciamos los textbox y combobox
ComboBox_Familia = ""
ComboBox_Categoria = ""
ComboBox_Indicador = ""
TxtFamilia = ""
TxtCategoria = ""
TxtIndicador = ""
Descripcion = ""
TxtFamilia.Visible = False
TxtFamilia.Enabled = False
TxtCategoria.Visible = True
TxtCategoria.Enabled = True
TxtIndicador.Visible = True
TxtIndicador.Enabled = True
TxtDescripcion.Visible = True
TxtDescripcion.Enabled = True
CmdAgregarFamilia.Visible = False
CmdAgregarFamilia.Enabled = False
CmdAgregarCategoria.Visible = True
CmdAgregarCategoria.Enabled = True
CmdAgregarIndicador.Visible = False
CmdAgregarIndicador.Enabled = False
CmdModificarFamilia.Visible = False
CmdModificarFamilia.Enabled = False
CmdModificarCategoria.Visible = False
CmdModificarCategoria.Enabled = False
CmdModificarIndicador.Visible = False
CmdModificarIndicador.Enabled = False
'Estos dos botones siempre deben estar.
CmdLimpiar.Visible = True

```

```

    CmdLimpiar.Enabled = True

    CmdCancelar.Visible = True

    CmdCancelar.Enabled = True

End Sub

Private Sub CmdNuevaCategoria_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal
X As Single, ByVal Y As Single)

    CmdNuevaCategoria.ForeColor = &HC00000

End Sub

'*****

*****

'CmdNuevoIndicador

'*****

*****

Private Sub CmdNuevoIndicador_Click()

'Activamos los botones,textbox y combobox, que vamos a utilizar.

    ComboBox_Familia.Enabled = True

    ComboBox_Familia.Visible = True

    ComboBox_Categoria.Enabled = True

    ComboBox_Categoria.Visible = True

    ComboBox_Indicador.Enabled = False

    ComboBox_Indicador.Visible = False

'Vaciamos los textbox y combobox

    ComboBox_Familia = ""

    ComboBox_Categoria = ""

    ComboBox_Indicador = ""

    TxtFamilia = ""

    TxtCategoria = ""

    TxtIndicador = ""

    Descripcion = ""

    TxtFamilia.Visible = False

    TxtFamilia.Enabled = False

    TxtCategoria.Visible = False

    TxtCategoria.Enabled = False

```

```

    TxtIndicador.Visible = True
    TxtIndicador.Enabled = True
    TxtDescripcion.Visible = True
    TxtDescripcion.Enabled = True
    CmdAgregarFamilia.Visible = False
    CmdAgregarFamilia.Enabled = False
    CmdAgregarCategoria.Visible = False
    CmdAgregarCategoria.Enabled = False
    CmdAgregarIndicador.Visible = True
    CmdAgregarIndicador.Enabled = True
    CmdModificarFamilia.Visible = False
    CmdModificarFamilia.Enabled = False
    CmdModificarCategoria.Visible = False
    CmdModificarCategoria.Enabled = False
    CmdModificarIndicador.Visible = False
    CmdModificarIndicador.Enabled = False
    'Estos dos botones siempre deben estar.
    CmdLimpiar.Visible = True
    CmdLimpiar.Enabled = True
    CmdCancelar.Visible = True
    CmdCancelar.Enabled = True
End Sub

Private Sub CmdNuevoIndicador_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X
As Single, ByVal Y As Single)
    CmdNuevoIndicador.ForeColor = &HC00000
End Sub

'*****
          ***** 'CmdModificarFamilia2
'*****
          *****

Private Sub CmdModificarFamilia2_Click()
    ComboBox_Familia.Enabled = True

```

ComboBox_Familia.Visible = True
ComboBox_Categoria.Enabled = False
ComboBox_Categoria.Visible = False

ComboBox_Indicador.Enabled = False
ComboBox_Indicador.Visible = False
ComboBox_Familia = ""
ComboBox_Categoria = ""
ComboBox_Indicador = ""

TxtFamilia = ""

TxtCategoria = ""

TxtIndicador = ""

Descripcion = ""

TxtFamilia.Visible = False

TxtFamilia.Enabled = False

TxtCategoria.Visible = True

TxtCategoria.Enabled = True

TxtIndicador.Visible = True

TxtIndicador.Enabled = True

TxtDescripcion.Visible = True

TxtDescripcion.Enabled = True

CmdAgregarFamilia.Visible = False

CmdAgregarFamilia.Enabled = False

CmdAgregarCategoria.Visible = False

CmdAgregarCategoria.Enabled = False

CmdAgregarIndicador.Visible = False

CmdAgregarIndicador.Enabled = False

CmdModificarFamilia.Visible = True

CmdModificarFamilia.Enabled = True

CmdModificarCategoria.Visible = False

CmdModificarCategoria.Enabled = False

CmdModificarIndicador.Visible = False

CmdModificarIndicador.Enabled = False

'Estos dos botones siempre deben estar.

CmdLimpiar.Visible = True

```

    CmdLimpiar.Enabled = True

    CmdCancelar.Visible = True

    CmdCancelar.Enabled = True

End Sub

Private Sub CmdModificarFamilia2_MouseMove(ByVal Button As Integer, ByVal Shift As Integer, ByVal X As Single, ByVal Y As Single)

    CmdModificarFamilia2.ForeColor = &H40C0&

End Sub

'*****
'***** ' CmdModificarCategoria2
'*****
'*****

Private Sub CmdModificarCategoria2_Click()

'Subrutina que se activa cuando pulsemos el boton de modificar categoria
'Activamos los botones,textbox y combobox, que vamos a utilizar.

    ComboBox_Familia.Enabled = True

    ComboBox_Familia.Visible = True

    ComboBox_Categoria.Enabled = True

    ComboBox_Categoria.Visible = True

    ComboBox_Indicador.Enabled = False

    ComboBox_Indicador.Visible = False

'Vaciamos los textbox y combobox

    ComboBox_Familia = ""

    ComboBox_Categoria = ""

    ComboBox_Indicador = ""

    TxtFamilia = ""

    TxtCategoria = ""

    TxtIndicador = ""

    Descripcion = ""

    TxtFamilia.Visible = False

    TxtFamilia.Enabled = False

    TxtCategoria.Visible = False

    TxtCategoria.Enabled = False

    TxtIndicador.Visible = True

```

```

    TxtIndicador.Enabled = True
    TxtDescripcion.Visible = True
    TxtDescripcion.Enabled = True
    CmdAgregarFamilia.Visible = False
    CmdAgregarFamilia.Enabled = False
    CmdAgregarCategoria.Visible = False
    CmdAgregarCategoria.Enabled = False

    CmdAgregarIndicador.Visible = False
    CmdAgregarIndicador.Enabled = False
    CmdModificarFamilia.Visible = False
    CmdModificarFamilia.Enabled = False
    CmdModificarCategoria.Visible = True
    CmdModificarCategoria.Enabled = True
    CmdModificarIndicador.Visible = False
    CmdModificarIndicador.Enabled = False

    'Estos dos botones siempre deben estar.
    CmdLimpiar.Visible = True
    CmdLimpiar.Enabled = True
    CmdCancelar.Visible = True
    CmdCancelar.Enabled = True
End Sub

Private Sub CmdModificarCategoria2_MouseMove(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)
    'Subrutina para modificar el boton de modificar la categoria
    CmdModificarCategoria2.ForeColor = &H40C0&
End Sub

'*****
'*****

' CmdModificarIndicador2

'*****
'*****

Private Sub CmdModificarIndicador2_Click()

```

ComboBox_Familia.Enabled = True
ComboBox_Familia.Visible = True
ComboBox_Categoria.Enabled = True
ComboBox_Categoria.Visible = True
ComboBox_Indicador.Enabled = True
ComboBox_Indicador.Visible = True
ComboBox_Familia = ""
ComboBox_Categoria = ""
ComboBox_Indicador = ""
TxtFamilia = ""
TxtCategoria = ""
TxtIndicador = ""
Descripcion = ""
TxtFamilia.Visible = False
TxtFamilia.Enabled = False
TxtCategoria.Visible = False
TxtCategoria.Enabled = False
TxtIndicador.Visible = False
TxtIndicador.Enabled = False
TxtDescripcion.Visible = True
TxtDescripcion.Enabled = True
CmdAgregarFamilia.Visible = False
CmdAgregarFamilia.Enabled = False
CmdAgregarCategoria.Visible = False
CmdAgregarCategoria.Enabled = False
CmdAgregarIndicador.Visible = False
CmdAgregarIndicador.Enabled = False
CmdModificarFamilia.Visible = False
CmdModificarFamilia.Enabled = False
CmdModificarCategoria.Visible = False
CmdModificarCategoria.Enabled = False
CmdModificarIndicador.Visible = True
CmdModificarIndicador.Enabled = True
'Estos dos botones siempre deben estar.
CmdLimpiar.Visible = True

```

    CmdLimpiar.Enabled = True

    CmdCancelar.Visible = True

    CmdCancelar.Enabled = True

End Sub

Private Sub CmdModificarIndicador2_MouseMove(ByVal Button As Integer, ByVal Shift As Integer,
ByVal X As Single, ByVal Y As Single)

'Subrutina para modificar el boton de modificar el indicador

    CmdModificarIndicador2.ForeColor = &H40C0&

End Sub

'*****
'*****

' ComboBox_Familia

'*****
'*****

Private Sub ComboBox_Familia_Change()

'Subrutina que se activa cuando se produce algun cambio en el ComboBox_Indicador _
Hace que se cargen los valores de Familia, Categoria y Descripción en los cuadros _
de textos correspondientes en función del indicador seleccionado.

Application.ScreenUpdating = False

On Error Resume Next

COLUMNNA_Familias = 4

Fila = 5

ActiveWorkbook.Worksheets("INDICADORES").Activate

Cells(ComboBox_Indicador.ListIndex + Fila, COLUMNNA_Familias).Select

TxtCategoria = ""

TxtIndicador = ""

TxtDescripcion = ""

TxtCategoria.SetFocus

End Sub

Private Sub ComboBox_Familia_Enter()

```



```

'Subrutina para llenar el ComboBox_Familia de todos los indicadores que hay _
en la hoja INDICADORES
Application.ScreenUpdating = False
On Error Resume Next

'Antes de nada debemos limpiar el comboBox _
Hacemos esto porque sino se nos repetirian los valores.
ComboBox_Familia.Clear
ComboBox_Categoria.Clear
ComboBox_Indicador.Clear
ComboBox_Familia = ""
ComboBox_Categoria = ""
ComboBox_Indicador = ""
TxtDescripcion.Value = ""
Fila = 5
Dim celda As Object
Dim i As Integer

ActiveWorkbook.Worksheets("INDICADORES").Activate
'generamos la coleccion
Set unicos = New Collection
'loop en todas las celdas y agregarlas a la coleccion
Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")
    Fila = Fila + 1
Loop
Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select
ActiveSheet.ListObjects("Tabla_Indicadores").Resize Range(Cells(4, 4), Cells(Fila, 10))
Range("Tabla_Indicadores[[#Headers],[FAMILIA]]").Select
For Each celda In Range(Cells(5, 4), Cells(Fila, 4))
    'cuando encuentre un item repetido, daría un error
    'que salvamos con la instrucción On Error Resume Next
    On Error Resume Next
    'por tanto, nuestra coleccion solo agrega elementos no repetidos
    'ocurre un error si una key especificada duplica la key de un miembro existente de la colección
    unicos.Add celda.Value, CStr(celda.Value)
    On Error GoTo 0

```

```

Next celda

'escribir los datos unicos en el ComboBox
For i = 1 To unicos.Count

    With ComboBox_Familia

        .AddItem (unicos(i))

        '(Worksheets("INDICADORES").Cells(Fila, COLUMNA_Familias).Value)

    End With

Next i

End Sub

'*****
'*****'ComboBox_Categoria

'*****
'*****

Private Sub ComboBox_Categoria_Enter()

    'Subrutina para llenar el ComboBox_Familia de todos los indicadores que hay en la hoja
    INDICADORES

    Application.ScreenUpdating = False

    On Error Resume Next

    'Definimos las constantes y las variables con las que vamos a trabajar.

    'Constante:

    Const N = 1000

    Const COLUMNA_Indicadores = 6

    Const COLUMNA_Categoria = 5

    Const COLUMNA_Familia = 4

    'Array:

    Dim Datos(N) As String

    'Array temporal

    Dim TempArray(N) As String

    'Variable

    Dim X As Integer, x2 As Integer, Y As Integer

    Dim z As Integer, Elemento As Variant

    Dim Fila As Integer

    'Inicializamos los valores de las variables:

```

```

Fila = 5

'Limpiamos el ComboBox
ComboBox_Categoria.Clear
ComboBox_Indicador.Clear
ComboBox_Categoria = ""
ComboBox_Indicador = ""
TxtDescripcion.Value = ""

'Activamos la hoja con la que vamos a trabajar
ActiveWorkbook.Worksheets("INDICADORES").Activate

If ComboBox_Familia.Value = "" Then

    Q = MsgBox("POR FAVOR, ELIJA UNA FAMILIA DE LAS DISPONIBLES", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")

Else

'Cargamos los valores en el vector

    Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")

        If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Familia).Value =
ComboBox_Familia.Value Then

            Datos(i) = Worksheets("INDICADORES").Cells(Fila, COLUMNA_Categoria).Value

            i = i + 1

        End If

        Fila = Fila + 1

    Loop

    FilaMax = Fila

    For i = LBound(Datos) To UBound(Datos)

        'Asignamos al array temporal el valor del otro array

        TempArray(i) = Datos(i)

    Next

    For X = 0 To UBound(Datos)

        z = 0

        For Y = 0 To UBound(Datos)

            'Si el elemento del array es igual al array temporal

            If Datos(X) = TempArray(z) And Y <> X Then

                'Entonces Eliminamos el valor duplicado

                Datos(Y) = ""

                'Nduplicado = Nduplicado + 1 'Número de elementos duplicados

```

```

        End If

        z = z + 1

    Next Y

Next X

For i = LBound(Datos) To UBound(Datos)

    If Datos(i) <> "" Then

        ComboBox_Categoria.AddItem Datos(i)

    End If

Next

End If

End Sub

'*****
'*****

' ComboBox_Indicador

'*****
'*****

Private Sub ComboBox_Indicador_Change()

'Subrutina que se activa cuando se produce algun cambio en el ComboBox_Indicador _
Hace que se cargen los valores de Familia, Categoria y Descripción en los cuadros _
de textos correspondientes en función del indicador seleccionado.

    Application.ScreenUpdating = False

    On Error Resume Next

    COLUMNA_Indicadores = 6

    COLUMNA_Descripcion = 7

    Fila = 5

    ActiveWorkbook.Worksheets("INDICADORES").Activate

    Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")

        If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value =
        ComboBox_Indicador.Value Then

            TxtDescripcion = Worksheets("INDICADORES").Cells(Fila, COLUMNA_Descripcion).Value

        End If

        Fila = Fila + 1

    Loop

End Sub

Private Sub ComboBox_Indicador_Enter()

```

```

'Subrutina para llenar el ComboBox_Indicador de todos los indicadores que hay _
en la hoja INDICADORES
Application.ScreenUpdating = False
On Error Resume Next
COLUMNA_Indicadores = 6
Fila = 5
COLUMNA_Categorias = 5
ComboBox_Indicador.Clear
ComboBox_Indicador = ""
TxtDescripcion.Value = ""
ActiveWorkbook.Worksheets("INDICADORES").Activate
If ComboBox_Familia.Value = "" Then
    Q = MsgBox("POR FAVOR, ELIJA UNA FAMILIA DE LAS DISPONIBLES", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
    ElseIf ComboBox_Categoria.Value = "" Then
        Q = MsgBox("POR FAVOR, ELIJA UNA CATEGORIA DE LAS DISPONIBLES", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
    Else
        'Tenemos que buscar el valor de la ultima fila.
        'Buscamos con el bucle do while
        Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")
            If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Categorias).Value =
ComboBox_Categoria.Value Then
                With ComboBox_Indicador
                    .AddItem (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value)
                End With
            End If
            Fila = Fila + 1
        Loop
    End If
End Sub

```

Tras ver toda la programación que tiene el formulario para añadir/modificar la base de datos, hay que pasar a los demás módulos de la herramienta, están ordenadas alfabéticamente para facilitar las posibles búsquedas

9.2.3. Módulo: AjustesTabla

Contiene todos los procedimientos para poder ajustar el tamaño de todas las tablas que forman la herramienta.

```
'*****  
*****
```

'Ajustes Tabla: CONTIENE LAS SUBROUTINAS PARA AJUSTAR EL TAMAÑO DE LAS
CELDA

```
'*****  
*****
```

Sub Ajuste_Tabla_RESUMEN(Fila_Max)

'Subrutina para dar el tamaño elegido a cada celdas

Dim Fila As Integer

ActiveWorkbook.Worksheets("RESUMEN").Activate

'D = FAMILIA

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

'E = CATEGORIA

Columns("E:E").ColumnWidth = 20

'F = INDICADORES

Columns("F:F").ColumnWidth = 40

'G = DESCRIPCION

Columns("G:G").ColumnWidth = 80

'H = EX-ANTE

Columns("H:H").ColumnWidth = 20

'J = IN-ITINERE

Columns("J:J").ColumnWidth = 20

'L = EX-POST

Columns("L:L").ColumnWidth = 20

'M=RESUMEN

Columns("M:M").ColumnWidth = 15

For Fila = 5 To Fila_Max 'Fila_Max

Rows(Fila).RowHeight = 30

Next Fila

End Sub

```

Sub Ajuste_Tabla_ETAPAS(Fila_Max)
'Subrutina para dar el tamaño elegido a cada celdas
    Dim Fila As Integer
'D=FAMILIA
    Columns("D:D").ColumnWidth = 20
'E = CATEGORIA
    Columns("E:E").ColumnWidth = 20
'F= INDICADORES
    Columns("F:F").ColumnWidth = 40
'G= PARÁMETROS
    Columns("G:G").ColumnWidth = 18
'H= PARÁMETROS
    Columns("H:H").ColumnWidth = 18
'I= PARÁMETROS
    Columns("I:I").ColumnWidth = 18
'J= PARÁMETROS
    Columns("J:J").ColumnWidth = 18
'K= PARÁMETROS
    Columns("K:K").ColumnWidth = 18
'L = RESUMEN
    Columns("L:L").ColumnWidth = 15

    For Fila = 5 To Fila_Max + 4 'Fila_Max
        Rows(Fila).RowHeight = 30
    Next Fila
End Sub

```

```

Sub Ajuste_Tabla_INDICADORES(Fila_Max)
'Subrutina para dar el tamaño elegido a cada celdas
    Dim Fila As Integer

    ActiveWorkbook.Worksheets("INDICADORES").Activate

'D = FAMILIA
    Columns("D:D").ColumnWidth = 20

```

```

'E = CATEGORIA
    Columns("E:E").ColumnWidth = 20
'F= INDICADORES
    Columns("F:F").ColumnWidth = 20
'G= DESCRIPCION
    Columns("G:G").ColumnWidth = 80
'H = FECHA
    Columns("H:H").ColumnWidth = 15
'J = EVALUAR
    Columns("J:J").ColumnWidth = 15

    Rows(2).RowHeight = 30

    For Fila = 5 To Fila_Max 'Fila_Max
        Rows(Fila).RowHeight = 30
    Next Fila
End Sub

```

9.2.4. Bordes_Tabla

```

'*****
*****

```

ENCUENTRAN TODAS LAS SUBROUTINAS PARA FORMATEAR LAS TABLAS

'RELLENAR DE BLANCO Y DAR FORMATO A LAS CELDAS QUE CONTIENE EL
CONTADOR

```

'*****
*****

```

Esta primera rutina, ha sido grabada gracias a la función de Excel para grabar Macro, como se puede ver, para que el procedimiento funcione correctamente se le debe introducir el número de filas en la llamada que se haga

```

'*****
*****

```

' Rutina grabada que da formato a la tabla de indicadores

```

'*****
*****

```

Sub bordes_tabla_indicadores(num_filas)

Application.ScreenUpdating = False


```

On Error Resume Next
ActiveWorkbook.Worksheets("INDICADORES").Activate
Range(Cells(5, 4), Cells(num_filas - 1, 10)).Select
With Selection
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)

```

```

.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
Range(Cells(5, 4), Cells(num_filas - 1, 4)).Select
With Selection.Font
.Name = "Calibri"
.Size = 11
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.ThemeColor = xlThemeColorLight1
.TintAndShade = 0
.ThemeFont = xlThemeFontNone
End With

Range(Cells(5, 4), Cells(num_filas - 1, 4)).Select
Selection.Font.Bold = True
With Selection.Interior

```

```

        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorAccent5
        .TintAndShade = 0.399975585192419
        .PatternTintAndShade = 0
    End With

```

```

End Sub

```

```

'*****
*****

```

'Rutina grabada que da formato a la tabla resumen

```

'*****
*****

```

```

Sub bordes_tabla_resumen(num_filas)
    Application.ScreenUpdating = False
    On Error Resume Next
    ActiveWorkbook.Worksheets("RESUMEN").Activate
    Range(Cells(5, 4), Cells(num_filas - 1, 13)).Select
    With Selection
        .HorizontalAlignment = xlLeft
        .VerticalAlignment = xlBottom
        .WrapText = False
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)

```

```

.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
.LineStyle = xlContinuous
.ColorIndex = 0
.TintAndShade = 0
.Weight = xlThin
End With

```

```

Range(Cells(5, 4), Cells(num_filas - 1, 4)).Select
With Selection.Font
    .Name = "Calibri"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ThemeColor = xlThemeColorLight1
    .TintAndShade = 0
    .ThemeFont = xlThemeFontNone
End With

Range(Cells(5, 4), Cells(num_filas - 1, 4)).Select
Selection.Font.Bold = True
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent5
    .TintAndShade = 0.399975585192419
    .PatternTintAndShade = 0
End With
End Sub

'*****
*****

'Rutina grabada que da formato a la tabla, de las fases del ciclo de vida del
proyecto evaluadas

'*****
*****

Sub bordes_tabla_ExAnte_InItinere_ExPost(num_filas)
    Application.ScreenUpdating = False
    On Error Resume Next
    Range(Cells(5, 4), Cells(num_filas - 1, 12)).Select

```

With Selection

```
.HorizontalAlignment = xlLeft  
.VerticalAlignment = xlBottom  
.WrapText = False  
.Orientation = 0  
.AddIndent = False  
.IndentLevel = 0  
.ShrinkToFit = False  
.ReadingOrder = xlContext  
.MergeCells = False
```

End With

```
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
```

```
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
```

With Selection.Borders(xlEdgeLeft)

```
.LineStyle = xlContinuous  
.ColorIndex = 0  
.TintAndShade = 0  
.Weight = xlThin
```

End With

With Selection.Borders(xlEdgeTop)

```
.LineStyle = xlContinuous  
.ColorIndex = 0  
.TintAndShade = 0  
.Weight = xlThin
```

End With

With Selection.Borders(xlEdgeBottom)

```
.LineStyle = xlContinuous  
.ColorIndex = 0  
.TintAndShade = 0  
.Weight = xlThin
```

End With

With Selection.Borders(xlEdgeRight)

```
.LineStyle = xlContinuous  
.ColorIndex = 0  
.TintAndShade = 0
```

```

        .Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlInsideHorizontal)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Range(Cells(5, 4), Cells(num_filas - 1, 4)).Select
With Selection.Font
    .Name = "Calibri"
    .Size = 11
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ThemeColor = xlThemeColorLight1
    .TintAndShade = 0
    .ThemeFont = xlThemeFontNone
End With

Range(Cells(5, 4), Cells(num_filas - 1, 4)).Select
Selection.Font.Bold = True
With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent5

```

```

        .TintAndShade = 0.399975585192419

        .PatternTintAndShade = 0

    End With

End Sub

'*****
*****

'Macro para formatear las hojas

'*****
*****

Sub Formateo_Hoja()

    Application.ScreenUpdating = False

    On Error Resume Next

    Cells.Select

    With Selection

        .WrapText = True

        .Orientation = 0

        .AddIndent = False

        .IndentLevel = 0

        .ShrinkToFit = False

        .ReadingOrder = xlContext

        .MergeCells = False

    End With

    With Selection.Font

        .Name = "Calibri"

        .Strikethrough = False

        .Superscript = False

        .Subscript = False

        .OutlineFont = False

        .Shadow = False

        .Underline = xlUnderlineStyleNone

        .ThemeColor = xlThemeColorLight1

        .TintAndShade = 0

        .ThemeFont = xlThemeFontMinor

    End With

```


With Selection.Font

.Name = "Calibri"
.Size = 11
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.ThemeColor = xlThemeColorLight1
.TintAndShade = 0
.ThemeFont = xlThemeFontMinor

End With

With Selection

.VerticalAlignment = xlCenter
.WrapText = True
.Orientation = 0
.AddIndent = False
.IndentLevel = 0
.ShrinkToFit = False
.ReadingOrder = xlContext
.MergeCells = False

End With

With Selection

.HorizontalAlignment = xlCenter
.VerticalAlignment = xlCenter
.WrapText = True
.Orientation = 0
.AddIndent = False
.IndentLevel = 0
.ShrinkToFit = False
.ReadingOrder = xlContext
.MergeCells = False

End With

Cells.EntireColumn.AutoFit

```

Cells.EntireColumn.AutoFit
Cells.EntireColumn.AutoFit
Cells.EntireColumn.AutoFit
End Sub

'*****
          *****

'Macro para rellenar de blanco todas las celdas
'*****
*****Sub Blanco()

Cells.Select

    With Selection.Interior

        .PatternColorIndex = xlAutomatic

        .ThemeColor = xlThemeColorDark1

        .TintAndShade = 0

        .PatternTintAndShade = 0

    End With

End Sub

'*****
          *****

'Subrutina grabada que da formato a los contadores tanto de indicadores como el
del tiempo
'*****
*****Sub Formato_Contadores()

    ActiveWorkbook.Worksheets("INDICADORES").Activate

    Range("J4").Select

    Selection.Copy

    Range("J2").Select

    Selection.PasteSpecial Paste:=xlPasteFormats, Operation:=xlNone, _
        SkipBlanks:=False, Transpose:=False

    Application.CutCopyMode = False

    Range("K2").Select

    Selection.Font.Bold = True

    Range("L2").Select

    With Selection.Font

        .ThemeColor = xlThemeColorDark1

```

```

        .TintAndShade = 0
    End With
    Columns("K:K").ColumnWidth = 10
End Sub

```

9.2.5. CalculoResumen

```

'*****
'*****

        'SE ENCUENTRAN LAS SUBROUTINA PARA CALCULAR EL RESUMEN

'*****
'*****

'*****
'*****

        'Subrutina para calcular el resumen del indicador. Para su utilización debemos
        activar previamente la hoja

'*****
'*****

Sub Calculo_Resumen()

    Application.ScreenUpdating = False

    On Error Resume Next

    Fila_Ini = 5

    Fila = Fila_Ini

    COLUMNA_Indicador = 6

    COLUMNA_Parametro1 = 7

    COLUMNA_Parametro2 = 8

    COLUMNA_Parametro3 = 9

    COLUMNA_Parametro4 = 10

    COLUMNA_Parametro5 = 11

    COLUMNA_Resumen = 12

    Num_Cero = 5

    mensaje = 0 'Variable para controlar el mensaje de parametros vacios, para que solo salga una
    vez

    Do While (Cells(Fila, COLUMNA_Indicador).Value <> "")

```

```

Fila = Fila + 1
Loop
Fila_Max = Fila
For Fila = Fila_Ini To (Fila_Max - 1)
    Valor1 = Cells(Fila, COLUMNA_Parametro1).Value
    Valor2 = Cells(Fila, COLUMNA_Parametro2).Value
    Valor3 = Cells(Fila, COLUMNA_Parametro3).Value
    Valor4 = Cells(Fila, COLUMNA_Parametro4).Value
    Valor5 = Cells(Fila, COLUMNA_Parametro5).Value
    'Si el indicar tiene algun párametro con 1, el resultado final será 1
    If (Cells(Fila, COLUMNA_Parametro1).Value = 1 Or Cells(Fila, COLUMNA_Parametro2).Value = 1
Or _
    Cells(Fila, COLUMNA_Parametro3).Value = 1 Or Cells(Fila, COLUMNA_Parametro4).Value = 1 Or
-
    Cells(Fila, COLUMNA_Parametro5).Value = 1) Then
        Cells(Fila, COLUMNA_Resumen).Value = 1
    Else
        If Cells(Fila, COLUMNA_Parametro1).Value = 6 Then
            Num_Cero = Num_Cero - 1
            Valor1 = 0
        End If
        If Cells(Fila, COLUMNA_Parametro2).Value = 6 Then
            Num_Cero = Num_Cero - 1
            Valor2 = 0
        End If
        If Cells(Fila, COLUMNA_Parametro3).Value = 6 Then
            Num_Cero = Num_Cero - 1
            Valor3 = 0
        End If
        If Cells(Fila, COLUMNA_Parametro4).Value = 6 Then
            Num_Cero = Num_Cero - 1
            Valor4 = 0
        End If
        If Cells(Fila, COLUMNA_Parametro5).Value = 6 Then
            Num_Cero = Num_Cero - 1

```

```

Valor5 = 0

End If

If (Valor1 = 0 And Valor2 = 0 And Valor3 = 0 And Valor4 = 0 And Valor5 = 0) Then
    resumen = 6
Else
    resumen = Fix((Valor1 + Valor2 + Valor3 + Valor4 + Valor5) / Num_Cero)
End If

If resumen = 0 Then
    Cells(Fila, COLUMNA_Resumen).Value = 1
End If

If resumen > 6 Then
    Cells(Fila, COLUMNA_Resumen).Value = 5
Else
    Cells(Fila, COLUMNA_Resumen).Value = resumen
End If

End If

'Si el indicar tiene algun párametro vacio no evaluamos nada.
If (Cells(Fila, COLUMNA_Parametro1).Value = "" Or _
Cells(Fila, COLUMNA_Parametro2).Value = "" Or _
Cells(Fila, COLUMNA_Parametro3).Value = "" Or _
Cells(Fila, COLUMNA_Parametro4).Value = "" Or _
Cells(Fila, COLUMNA_Parametro5).Value = "") Then
    Cells(Fila, COLUMNA_Resumen).Value = ""

    If mensaje = 0 Then
        Q = MsgBox("EVALUE TODOS LOS PARÁMETROS," & vbCr & "SI NO QUIERE EVALUAR
ALGUNO INSERTE UN 6", vbExclamation, " HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
        mensaje = 1
    End If
End If

Num_Cero = 5

Next Fila

End Sub

'*****
*****

```

'Subrutina para calcular el resumen de Ex-Ante


```
Sub Calculo_Resumen_Ex_Ante()  
    Application.ScreenUpdating = False  
    On Error Resume Next  
    ActiveWorkbook.Worksheets("EX-ANTE").Activate  
    Calculo_Resumen  
End Sub  
  
Sub Calculo_Resumen_In_Itinere()  
    Application.ScreenUpdating = False  
    On Error Resume Next  
    ActiveWorkbook.Worksheets("IN-ITINERE").Activate  
    Calculo_Resumen  
End Sub  
  
Sub Calculo_Resumen_Ex_Post()  
    Application.ScreenUpdating = False  
    On Error Resume Next  
    ActiveWorkbook.Worksheets("EX-POST").Activate  
    Calculo_Resumen  
End Sub
```

9.2.6. ConfigurarBotones

‘CONFIGURACIÓN DE BOTONES DE LAS HOJAS

'Subrutina para llamar al formulario de ingreso de indicadores


```
Sub INGRESO()  
    Application.ScreenUpdating = False  
    On Error Resume Next
```

```

Load Formulario_N_Indicador
Formulario_N_Indicador.Show
End Sub

'*****

*****

'Subroutines para activar las hojas del libro, se utilizan en los botones

'*****

*****

Sub Activar_Hoja_Indicadores()
Application.ScreenUpdating = False
On Error Resume Next
Worksheets("INDICADORES").Activate
End Sub

Sub Activar_Hoja_Ayuda()
Application.ScreenUpdating = False
On Error Resume Next
Worksheets("AYUDA").Activate
End Sub

Sub Activar_Hoja_Base_Indicadores()
Application.ScreenUpdating = False
On Error Resume Next
Worksheets("BASE DE DATOS INICIAL").Activate
End Sub

Sub Activar_Hoja_Ex_Ante()
Application.ScreenUpdating = False
On Error Resume Next
Worksheets("EX-ANTE").Activate
End Sub

Sub Activar_Hoja_In_Itinere()
Application.ScreenUpdating = False
On Error Resume Next
Worksheets("IN-ITINERE").Activate
End Sub

Sub Activar_Hoja_Ex_Post()
Application.ScreenUpdating = False

```

```

    On Error Resume Next
    Worksheets("EX-POST").Activate
End Sub

Sub Activar_Hoja_Resumen()
    Application.ScreenUpdating = False
    On Error Resume Next
    Worksheets("RESUMEN").Activate
End Sub

```

9.2.7. ContadorBlanco

```

'*****
'
'Subrutina que poner en blanco el contador de la hoja indicadores
'
'*****

Sub Contador_Blanco()
    Columns("K:K").ColumnWidth = 5
    Range("L2").Select
    With Selection.Font
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = 0
    End With
    Range("K3").Select
    With Selection.Font
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = 0
    End With
    ActiveWindow.SmallScroll Down:=-10
    Range("E5").Select
End Sub

```

9.2.8. CopiarResultados

```

'*****
'

```


'CONTIENE LAS SUBROUTINAS PARA OBTENER EL RESUMEN DE LA EVALUACIÓN

'Subrutina que se encarga de copiar los valores resúmenes de las hojas Ex-Ante, In-Itinere y Ex-Post a la hoja Resumen para visibilidad de la evaluación completa del proyecto

Sub Copiar_Resultados()

Application.ScreenUpdating = False

On Error Resume Next

Application.ScreenUpdating = False

Dim Fila As Integer

Dim COLUMNA_Indicadores As Integer

Dim COLUMNA_R_Ex_Ante As Integer

Dim COLUMNA_R_In_Itinere As Integer

Dim COLUMNA_R_Ex_Post As Integer

Dim Fila_Ini As Integer

Dim media As Integer

COLUMNA_Indicadores = 6 'Nº de columna de la hoja Indicadores donde se encuentran los _ indicadores.

COLUMNA_Resumen = 12 'Nº de columna de las hojas, Ex-Ante, In-Itinere y Ex-Post donde _ se encuentra el resumen de la evaluación.

COLUMNA_R_Ex_Ante = 8 'Nº de la columna de la hoja Resumen donde se muestra el resumen _ de Ex-Ante.

COLUMNA_R_In_Itinere = 10 'Nº de la columna de la hoja Resumen donde se muestra el _ resumen de In-Itinere.

COLUMNA_R_Ex_Post = 12 'Nº de la columna de la hoja Resumen donde se muestra el _ resumen de Ex-Post.

COLUMNA_Resumen_Final = 13 'Nº de la columna de la hoja Resumen donde se muestra el _ resumen final

Fila_Ini = 5

Fila = Fila_Ini

```

vacio = 0

Num_Cero = 3

'Buscamos el valor máximo de filas, aunque tambien se lo podemos pasar como parámetro.
Do While (Worksheets("EX-ANTE").Cells(Fila, COLUMNA_Indicadores).Value <> "")

    Fila = Fila + 1

Loop

Fila_Max = Fila

For Fila = Fila_Ini To Fila_Max

    Application.CutCopyMode = True

    Worksheets("EX-ANTE").Cells(Fila, COLUMNA_Resumen).Copy

    Sheets("RESUMEN").Select

    Worksheets("RESUMEN").Cells(Fila, COLUMNA_R_Ex_Ante).Select

    ActiveSheet.Paste

    Worksheets("IN-ITINERE").Cells(Fila, COLUMNA_Resumen).Copy

    Sheets("RESUMEN").Select

    Worksheets("RESUMEN").Cells(Fila, COLUMNA_R_In_Itinere).Select

    ActiveSheet.Paste

    Worksheets("EX-POST").Cells(Fila, COLUMNA_Resumen).Copy

    Sheets("RESUMEN").Select

    Worksheets("RESUMEN").Cells(Fila, COLUMNA_R_Ex_Post).Select

    ActiveSheet.Paste

    Application.CutCopyMode = False 'Borra la memoria

Next Fila

For Fila = Fila_Ini To Fila_Max - 1

    Worksheets("RESUMEN").Activate

    Valor1 = Cells(Fila, COLUMNA_R_Ex_Ante).Value

    Valor2 = Cells(Fila, COLUMNA_R_In_Itinere).Value

    Valor3 = Cells(Fila, COLUMNA_R_Ex_Post).Value

    If Cells(Fila, COLUMNA_R_Ex_Ante).Value = "" Or Cells(Fila, COLUMNA_R_In_Itinere).Value = "" _
    Or Cells(Fila, COLUMNA_R_Ex_Post).Value = "" Then

        Worksheets("RESUMEN").Cells(Fila, COLUMNA_Resumen_Final).Value = ""

        vacio = 1

    Else

        If Cells(Fila, COLUMNA_R_Ex_Ante).Value = 6 Then

            Num_Cero = Num_Cero - 1

```

```

Valor1 = 0
End If

If Cells(Fila, COLUMNA_R_In_Itinere).Value = 6 Then
    Num_Cero = Num_Cero - 1
    Valor2 = 0
End If

If Cells(Fila, COLUMNA_R_Ex_Post).Value = 6 Then
    Num_Cero = Num_Cero - 1
    Valor3 = 0
End If

If (Valor1 = 0 And Valor2 = 0 And Valor3 = 0) Then
    Worksheets("RESUMEN").Cells(Fila, COLUMNA_Resumen_Final).Value = 6
Else
    media = Round((Valor1 + Valor2 + Valor3) / Num_Cero)
    Worksheets("RESUMEN").Cells(Fila, COLUMNA_Resumen_Final).Value = media
End If

1 _ If Cells(Fila, COLUMNA_R_Ex_Ante).Value = 1 Or Cells(Fila, COLUMNA_R_In_Itinere).Value =
    Or Cells(Fila, COLUMNA_R_Ex_Post).Value = 1 Then
        Worksheets("RESUMEN").Cells(Fila, COLUMNA_Resumen_Final).Value = 1
    End If
End If

Num_Cero = 3
Next Fila

'CALCULAMOS EL RESUMEN FINAL

Call Tabla_Valoracion_Etapas 'Llamamos a la subrutina para la creación de la tabla resumen
etapas

Call Formato_Tabla_Final

Num_Cero1 = Fila_Max - Fila_Ini
Num_Cero2 = Fila_Max - Fila_Ini
Num_Cero3 = Fila_Max - Fila_Ini
Valorfinal1 = 0
Valorfinal2 = 0

```

```

Valorfinal3 = 0
For Fila = Fila_Ini To Fila_Max - 1
'Valores de la columna Ex-Ante
    Valor1 = Cells(Fila, COLUMNA_R_Ex_Ante).Value
    If Valor1 = 6 Or Valor1 = "" Then
        ValorSum1 = 0
        Num_Cero1 = Num_Cero1 - 1
    Else
        ValorSum1 = Valor1
    End If
    Valorfinal1 = Valorfinal1 + ValorSum1
'Valores de la columna In-Itinere
    Valor2 = Cells(Fila, COLUMNA_R_In_Itinere).Value
    If Valor2 = 6 Or Valor2 = "" Then
        ValorSum2 = 0
        Num_Cero2 = Num_Cero2 - 1
    Else
        ValorSum2 = Valor2
    End If
    Valorfinal2 = Valorfinal2 + ValorSum2
'Valores de la columna Ex-Post
    Valor3 = Cells(Fila, COLUMNA_R_Ex_Post).Value
    If Valor3 = 6 Or Valor3 = "" Then
        ValorSum3 = 0
        Num_Cero3 = Num_Cero3 - 1
    Else
        ValorSum3 = Valor3
    End If
    Valorfinal3 = Valorfinal3 + ValorSum3
Next Fila
If Valorfinal1 = 0 Or Num_Cero1 = 0 Then
    Valordefinitivo1 = 6
Else
    Valordefinitivo1 = Round(Valorfinal1 / Num_Cero1)
End If

```

```

If Valorfinal2 = 0 Or Num_Cero2 = 0 Then
    Valordefinitivo2 = 6
Else
    Valordefinitivo2 = Round(Valorfinal2 / Num_Cero2)
End If

If Valorfinal3 = 0 Or Num_Cero3 = 0 Then
    Valordefinitivo3 = 6
Else
    Valordefinitivo3 = Round(Valorfinal3 / Num_Cero3)
End If

Worksheets("RESUMEN").Range("R6:R7").Value = Valordefinitivo1
Worksheets("RESUMEN").Range("S6:S7").Value = Valordefinitivo2
Worksheets("RESUMEN").Range("T6:T7").Value = Valordefinitivo3

'Vemos si hay algun indicador con valor limitante 1, puesto que si le han la evaluación será
negativa.
For Fila = Fila_Ini To Fila_Max - 1
    If Cells(Fila, COLUMNNA_R_Ex_Ante).Value = 1 Then
        Worksheets("RESUMEN").Range("R6:R7").Value = 1
    End If
    If Cells(Fila, COLUMNNA_R_In_Itinere).Value = 1 Then
        Worksheets("RESUMEN").Range("S6:S7").Value = 1
    End If
    If Cells(Fila, COLUMNNA_R_Ex_Post).Value = 1 Then
        Worksheets("RESUMEN").Range("T6:T7").Value = 1
    End If
Next Fila

' Con una simple variable decimos si la evaluación es completa o no, sacando un mensaje por
pantalla
If vacio = 1 Then
    Q = MsgBox("EVALUACIÓN INCOMPLETA, POR FAVOR EXAMINELA", vbExclamation, "
HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")
End If
End Sub

```

9.2.9. Filtros

```
'*****  
*****
```

‘CONTIENE LAS SUBROUTINAS PARA APLICAR LOS FILTROS AUTOMATICOS

```
'*****  
*****
```

```
Sub Filtro_Resumen()  
    Range("D4:M4").Select  
    Selection.AutoFilter  
End Sub
```

```
Sub Filtro_Etapas()  
    Range("D4:L4").Select  
    Selection.AutoFilter  
End Sub
```

```
Sub Filtro_Indicadores()  
    Range("D4:J4").Select  
    Selection.AutoFilter  
End Sub
```

9.2.10. Formato_encabezados

```
'*****  
*****
```

DONDE SE ENCUENTRAN TODAS LAS SUBROUTINAS PARA FORMATERA LOS
ENCABEZADOS

```
'*****  
*****
```

```
'*****  
*****
```

' Macro grabada que da formato al encabezado de la hoja INDICADORES

!*****

```
Sub Formato_Encabezado_Indicadores()  
    Application.ScreenUpdating = False  
    On Error Resume Next  
    ActiveWorkbook.Worksheets("INDICADORES").Activate  
    Range("D4:J4").Select  
    With Selection  
        .HorizontalAlignment = xlGeneral  
        .VerticalAlignment = xlCenter  
        .WrapText = True  
        .Orientation = 0  
        .AddIndent = False  
        .IndentLevel = 0  
        .ShrinkToFit = False  
        .ReadingOrder = xlContext  
        .MergeCells = False  
    End With  
    With Selection  
        .HorizontalAlignment = xlCenter  
        .VerticalAlignment = xlCenter  
        .WrapText = True  
        .Orientation = 0  
        .AddIndent = False  
        .IndentLevel = 0  
        .ShrinkToFit = False  
        .ReadingOrder = xlContext  
        .MergeCells = False  
    End With  
    Selection.Font.Bold = True  
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone  
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone  
    With Selection.Borders(xlEdgeLeft)  
        .LineStyle = xlContinuous  
        .ColorIndex = 0
```

```

        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlInsideVertical)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorAccent5
        .TintAndShade = 0.399975585192419
        .PatternTintAndShade = 0
    End With

```


End Sub

'*****

‘Macro que da formato al encabezado de la hoja RESUMEN

'*****

Sub Formato_Encabezado_Resumen()

Application.ScreenUpdating = False

On Error Resume Next

ActiveWorkbook.Worksheets("RESUMEN").Activate

Range("D4:M4").Select

Selection.Font.Bold = True

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlBottom

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

With Selection.Interior

.Pattern = xlSolid

.PatternColorIndex = xlAutomatic

```

.ThemeColor = xlThemeColorAccent5
.TintAndShade = 0.399975585192419
.PatternTintAndShade = 0
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlInsideVertical)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin

```

```

End With

Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

Range("H3:L3").Select

With Selection

    .HorizontalAlignment = xlCenter

    .VerticalAlignment = xlBottom

    .WrapText = False

    .Orientation = 0

    .AddIndent = False

    .IndentLevel = 0

    .ShrinkToFit = False

    .ReadingOrder = xlContext

    .MergeCells = False

End With

Selection.Merge

With Selection

    .HorizontalAlignment = xlCenter

    .VerticalAlignment = xlCenter

    .WrapText = False

    .Orientation = 0

    .AddIndent = False

    .IndentLevel = 0

    .ShrinkToFit = False

    .ReadingOrder = xlContext

    .MergeCells = True

End With

With Selection.Interior

    .Pattern = xlSolid

    .PatternColorIndex = xlAutomatic

    .ThemeColor = xlThemeColorAccent5

    .TintAndShade = 0.399975585192419

    .PatternTintAndShade = 0

End With

Selection.Font.Bold = True

Selection.Borders(xlDiagonalDown).LineStyle = xlNone

```

```

Selection.Borders(xlDiagonalUp).LineStyle = xlNone

With Selection.Borders(xlEdgeLeft)

    .LineStyle = xlContinuous

    .ColorIndex = 0

    .TintAndShade = 0

    .Weight = xlThin

End With

With Selection.Borders(xlEdgeTop)

    .LineStyle = xlContinuous

    .ColorIndex = 0

    .TintAndShade = 0

    .Weight = xlThin

End With

With Selection.Borders(xlEdgeBottom)

    .LineStyle = xlContinuous

    .ColorIndex = 0

    .TintAndShade = 0

    .Weight = xlThin

End With

With Selection.Borders(xlEdgeRight)

    .LineStyle = xlContinuous

    .ColorIndex = 0

    .TintAndShade = 0

    .Weight = xlThin

End With

Selection.Borders(xlInsideVertical).LineStyle = xlNone

Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

End Sub

'*****

' Macro que da formato al encabezado de la hoja EX-ANTE, IN-ITINERE y EX-POST

'*****

Sub Formato_Encabezado_ExAnte_InItinere_ExPost()

```

Application.ScreenUpdating = False

On Error Resume Next

Range("D4:L4").Select

With Selection

.HorizontalAlignment = xlGeneral

.VerticalAlignment = xlCenter

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Font.Bold = True

With Selection.Font

.Name = "Calibri"

.Size = 11

.Strikethrough = False

.Superscript = False

.Subscript = False

.OutlineFont = False

.Shadow = False

.Underline = xlUnderlineStyleNone

.ThemeColor = xlThemeColorLight1

.TintAndShade = 0

```

.ThemeFont = xlThemeFontMinor
End With
With Selection.Font
.Name = "Calibri"
.Size = 11
.Strikethrough = False
.Superscript = False
.Subscript = False
.OutlineFont = False
.Shadow = False
.Underline = xlUnderlineStyleNone
.ThemeColor = xlThemeColorLight1
.TintAndShade = 0
.ThemeFont = xlThemeFontMinor
End With
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
.LineStyle = xlContinuous
.ColorIndex = xlAutomatic
.TintAndShade = 0
.Weight = xlThin
End With

```

```

With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlThin
End With

With Selection.Borders(xlInsideVertical)
    .LineStyle = xlContinuous
    .ColorIndex = xlAutomatic
    .TintAndShade = 0
    .Weight = xlThin
End With

Selection.Borders(xlInsideHorizontal).LineStyle = xlNone

With Selection.Interior
    .Pattern = xlSolid
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorAccent5
    .TintAndShade = 0.399975585192419
    .PatternTintAndShade = 0
End With

Range("G3:K3").Select

With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlCenter
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = False
End With

Selection.Merge

With Selection.Interior
    .Pattern = xlSolid

```

```

        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorAccent5
        .TintAndShade = 0.399975585192419
        .PatternTintAndShade = 0
    End With
    Selection.Font.Bold = True
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeTop)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
End Sub

```


9.2.11. FormatoCondicional

```
*****  
*****
```

'CONTIENE LAS SUBROUTINAS PARA APLICAR FORMATO CONDICIONAL A LAS CELDAS

```
*****  
*****
```

```
*****  
*****
```

'Subrutina para dar formato condicional a la hoja Resuemen

```
*****  
*****
```

Sub FormatoCondicional_Resumen()

Application.ScreenUpdating = False

On Error Resume Next

Cells.FormatConditions.Delete

Columns("H:M").Select

Selection.FormatConditions.AddColorScale ColorScaleType:=3

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority

Selection.FormatConditions(1).ColorScaleCriteria(1).Type = _

xlConditionValueNumber

Selection.FormatConditions(1).ColorScaleCriteria(1).Value = 1

With Selection.FormatConditions(1).ColorScaleCriteria(1).FormatColor

.Color = 255

.TintAndShade = 0

End With

Selection.FormatConditions(1).ColorScaleCriteria(2).Type = _

xlConditionValueNumber

Selection.FormatConditions(1).ColorScaleCriteria(2).Value = 3

With Selection.FormatConditions(1).ColorScaleCriteria(2).FormatColor

.Color = 65535

.TintAndShade = 0

End With

Selection.FormatConditions(1).ColorScaleCriteria(3).Type = _

xlConditionValueNumber

Selection.FormatConditions(1).ColorScaleCriteria(3).Value = 5

```

With Selection.FormatConditions(1).ColorScaleCriteria(3).FormatColor
    .Color = 5287936
    .TintAndShade = 0
End With

ActiveWindow.SmallScroll Down:=0

Selection.FormatConditions.AddIconSetCondition
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1)
    .ReverseOrder = False
    .ShowIconOnly = False
    .IconSet = ActiveWorkbook.IconSets(xl4TrafficLights)
End With

Selection.FormatConditions(1).IconCriteria(1).Icon = xlIconRedCrossSymbol
With Selection.FormatConditions(1).IconCriteria(2)
    .Type = xlConditionValueNumber
    .Value = 3
    .Operator = 7
    .Icon = xlIconYellowExclamationSymbol
End With

With Selection.FormatConditions(1).IconCriteria(3)
    .Type = xlConditionValueNumber
    .Value = 5
    .Operator = 7
    .Icon = xlIconGreenCheckSymbol
End With

With Selection.FormatConditions(1).IconCriteria(4)
    .Type = xlConditionValueNumber
    .Value = 6
    .Operator = 7
    .Icon = xlIconWhiteCircleAllWhiteQuarters
End With

Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="=6"

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1).Interior

```

```

        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = -0.249946592608417
    End With

    Selection.FormatConditions(1).StopIfTrue = False
End Sub

'*****
'*****

' Subrutina para dar formato condicional a las hojas Ex-ante, In-itinere, Ex-post
'*****
'*****

Sub FormatoCondicional_EtapasProyecto()
    Application.ScreenUpdating = False
    On Error Resume Next
    Cells.FormatConditions.Delete
    Columns("G:L").Select
        Selection.FormatConditions.AddColorScale ColorScaleType:=3
    Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
    Selection.FormatConditions(1).ColorScaleCriteria(1).Type = _
        xlConditionValueNumber
    Selection.FormatConditions(1).ColorScaleCriteria(1).Value = 1
    With Selection.FormatConditions(1).ColorScaleCriteria(1).FormatColor
        .Color = 255
        .TintAndShade = 0
    End With
    Selection.FormatConditions(1).ColorScaleCriteria(2).Type = _
        xlConditionValueNumber
    Selection.FormatConditions(1).ColorScaleCriteria(2).Value = 3
    With Selection.FormatConditions(1).ColorScaleCriteria(2).FormatColor
        .Color = 65535
        .TintAndShade = 0
    End With
    Selection.FormatConditions(1).ColorScaleCriteria(3).Type = _
        xlConditionValueNumber
    Selection.FormatConditions(1).ColorScaleCriteria(3).Value = 5

```

```

With Selection.FormatConditions(1).ColorScaleCriteria(3).FormatColor
    .Color = 5287936
    .TintAndShade = 0
End With

ActiveWindow.SmallScroll Down:=0

Selection.FormatConditions.AddIconSetCondition
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1)
    .ReverseOrder = False
    .ShowIconOnly = False
    .IconSet = ActiveWorkbook.IconSets(xl4TrafficLights)
End With

Selection.FormatConditions(1).IconCriteria(1).Icon = xlIconRedCrossSymbol
With Selection.FormatConditions(1).IconCriteria(2)
    .Type = xlConditionValueNumber
    .Value = 3
    .Operator = 7
    .Icon = xlIconYellowExclamationSymbol
End With

With Selection.FormatConditions(1).IconCriteria(3)
    .Type = xlConditionValueNumber
    .Value = 5
    .Operator = 7
    .Icon = xlIconGreenCheckSymbol
End With

With Selection.FormatConditions(1).IconCriteria(4)
    .Type = xlConditionValueNumber
    .Value = 6
    .Operator = 7
    .Icon = xlIconWhiteCircleAllWhiteQuarters
End With

Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="=6"

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1).Interior

```

```

        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorDark1
        .TintAndShade = -0.249946592608417
    End With
    Selection.FormatConditions(1).StopIfTrue = False
End Sub

'*****
'*****

'Subrutina grabada para aplicar un formato condicionar a la columna evaluar de la
hoja de indicadores _

Si la celda contiene:

' "OK" la celda se colora de color verde, indica que se va a evaluar
' "NOK" la celda se colora de color rojo, indica que no se va a colorear
' La celda vacia de color gris, para indicar que todavia no se ha tomado decición sobre
el indicador

'*****
'*****

Sub Formato_Condicional_Indicadores(Fila_Max)
    Worksheets("INDICADORES").Activate
    'Cells.FormatConditions.Delete
    Range(Cells(5, 10), Cells(Fila_Max - 1, 10)).Select
    Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
        Formula1:="=""OK""
    Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
    With Selection.FormatConditions(1).Font
        .Bold = True
        .Italic = False
        .TintAndShade = 0
    End With
    With Selection.FormatConditions(1).Interior
        .PatternColorIndex = xlAutomatic
        .Color = 5296274
    End With
End Sub

```

```

        .TintAndShade = 0
End With
Selection.FormatConditions(1).StopIfTrue = False
Worksheets("INDICADORES").Activate
Range(Cells(5, 10), Cells(Fila_Max - 1, 10)).Select
Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="=""NOK""
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1).Font
    .Bold = True
    .Italic = False
    .TintAndShade = 0
End With
With Selection.FormatConditions(1).Interior
    .PatternColorIndex = xlAutomatic
    .Color = 255
    .TintAndShade = 0
End With
Selection.FormatConditions(1).StopIfTrue = False
Worksheets("INDICADORES").Activate
Range(Cells(5, 10), Cells(Fila_Max - 1, 10)).Select
Selection.FormatConditions.Add Type:=xlExpression, Formula1:= _
    "=LARGO(ESPACIOS(J5))=0"
Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
With Selection.FormatConditions(1).Interior
    .PatternColorIndex = xlAutomatic
    .ThemeColor = xlThemeColorDark1
    .TintAndShade = -0.249946592608417
End With
Selection.FormatConditions(1).StopIfTrue = False
End Sub

'*****
*****

'*****
*****

```

```

Sub Formato_Tabla_Final()
    Application.ScreenUpdating = False
    On Error Resume Next
    'Cells.FormatConditions.Delete
    Range("R6:T7").Select
        Selection.FormatConditions.AddColorScale ColorScaleType:=3
    Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
    Selection.FormatConditions(1).ColorScaleCriteria(1).Type = _
        xlConditionValueNumber
    Selection.FormatConditions(1).ColorScaleCriteria(1).Value = 1
    With Selection.FormatConditions(1).ColorScaleCriteria(1).FormatColor
        .Color = 255
        .TintAndShade = 0
    End With
    Selection.FormatConditions(1).ColorScaleCriteria(2).Type = _
        xlConditionValueNumber
    Selection.FormatConditions(1).ColorScaleCriteria(2).Value = 3
    With Selection.FormatConditions(1).ColorScaleCriteria(2).FormatColor
        .Color = 65535
        .TintAndShade = 0
    End With
    Selection.FormatConditions(1).ColorScaleCriteria(3).Type = _
        xlConditionValueNumber
    Selection.FormatConditions(1).ColorScaleCriteria(3).Value = 5
    With Selection.FormatConditions(1).ColorScaleCriteria(3).FormatColor
        .Color = 5287936
        .TintAndShade = 0
    End With
    ActiveWindow.SmallScroll Down:=0
    Selection.FormatConditions.AddIconSetCondition
    Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority
    With Selection.FormatConditions(1)
        .ReverseOrder = False
        .ShowIconOnly = False
        .IconSet = ActiveWorkbook.IconSets(xl4TrafficLights)
    End With
End Sub

```

```

End With

Selection.FormatConditions(1).IconCriteria(1).Icon = xlIconRedCrossSymbol

With Selection.FormatConditions(1).IconCriteria(2)

    .Type = xlConditionValueNumber

    .Value = 3

    .Operator = 7

    .Icon = xlIconYellowExclamationSymbol

End With

With Selection.FormatConditions(1).IconCriteria(3)

    .Type = xlConditionValueNumber

    .Value = 5

    .Operator = 7

    .Icon = xlIconGreenCheckSymbol

End With

With Selection.FormatConditions(1).IconCriteria(4)

    .Type = xlConditionValueNumber

    .Value = 6

    .Operator = 7

    .Icon = xlIconWhiteCircleAllWhiteQuarters

End With

Selection.FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="=6"

Selection.FormatConditions(Selection.FormatConditions.Count).SetFirstPriority

With Selection.FormatConditions(1).Interior

    .PatternColorIndex = xlAutomatic

    .ThemeColor = xlThemeColorDark1

    .TintAndShade = -0.249946592608417

End With

Selection.FormatConditions(1).StopIfTrue = False

End Sub

```

9.2.12. FormatoTexto

```

'*****
*****

```


CONTIENE LA RUTINA PARA DAR EL FORMATO DE TEXTO ADECUADO A LA HOJA INDICADORES

'Subrutina para poner en mayucular el contenido de las celdas que deseemos

Sub Formato_Hoja_Indicadores()

Application.ScreenUpdating = False 'Código para que no se muestre la ejecución de la macro

On Error Resume Next

Dim Fila As Integer

Fila = 5

Worksheets("INDICADORES").Activate

Do While (Worksheets("INDICADORES").Cells(Fila, 6).Value <> "")

Fila = Fila + 1

Loop

Fila_Max = Fila

For Fila = 5 To (Fila_Max - 1)

'Ponemos en mayusculas la columna FAMILIA

Cells(Fila, 4).Value = UCase(Cells(Fila, 4).Value

'Damos formato de oración a las columnas CATEGORIA, INDICADOR Y DESCRIPCIÓN

Cells(Fila, 5).Value = LCase(Cells(Fila, 5).Value)

Largo5 = Len(Cells(Fila, 5))

Cells(Fila, 5).Value = UCase(Mid(Cells(Fila, 5).Value, 1, 1)) & LCase(Mid(Cells(Fila, 5).Value, 2, Largo5 - 1))

'Cells(Fila, 5).Value = Application.WorksheetFunction.Proper(Cells(Fila, 5).Value

Largo6 = Len(Cells(Fila, 6))

Cells(Fila, 6).Value = LCase(Cells(Fila, 6).Value)

Cells(Fila, 6).Value = UCase(Mid(Cells(Fila, 6).Value, 1, 1)) & LCase(Mid(Cells(Fila, 6).Value, 2, Largo6 - 1))

'Cells(Fila, 6).Value = Application.WorksheetFunction.Proper(Cells(Fila, 6).Value)

Largo7 = Len(Cells(Fila, 7))

Cells(Fila, 7).Value = LCase(Cells(Fila, 7).Value)

Cells(Fila, 7).Value = UCase(Mid(Cells(Fila, 7).Value, 1, 1)) & LCase(Mid(Cells(Fila, 7).Value, 2, Largo7 - 1))

'Cells(Fila, 7).Value = Application.WorksheetFunction.Proper(Cells(Fila, 7).Value)

```

        'Ponemos en mayusculas OK o NOK  columna EVALUAR
        Cells(Fila, 10).Value = UCase(Cells(Fila, 10).Value)

        Next Fila
    End Sub

```

9.2.13. GuardarCopia

```

'*****
'*****

'CONTIENE EL CÓDIGO PARA EXPORTAR UNA COPIA DE LA EVALUACIÓN EN .XLSX
'*****
'*****

Sub Guardar_Copia()

    'Pedimos el nombre que queramos poner al archivo.

    Respuesta = InputBox("Introduzca el nombre del archivo", "Nombre archivo")

    FilePath = ThisWorkbook.Path & "/" & Respuesta & "_" & Format(Now, "dd-mm-yyyy") & ".xlsx"

    'Guardamos el archivo

    ActiveWorkbook.Save

    'Guardamos la copia del archivo, con formato .xlsx

    MsgBox ("La herramienta se guarda y se cierra, queda abierta la copia exportada")

    MsgBox ("El archivo se ha guardado correctamente con nombre: " & vbCr & FilePath)

    ActiveWorkbook.SaveAs Filename:= FilePath, FileFormat:=xlOpenXMLWorkbook,
    CreateBackup:=False

End Sub

```

9.2.14. Main_Copiar_Elementos

```

'*****
'*****

'MODULO PRINCIPAL ➔ subrutina que copia los indicadores de la hoja
"INDICADORES" y les pega en las demas hojas, llamando a los demás
procesimientos

```


Sub Copiar_elementos()

Application.ScreenUpdating = False 'Código para que no se muestre la ejecución de la macro

On Error Resume Next

'ActiveWorkbook.Worksheets("INDICADORES").Activate

'Variables a utilizar

Dim Fila As Integer 'Número de fila

Dim Fila_Ini As Integer 'Fila inicial

Dim COLUMNA_Indicadores As Integer 'Número de columna indicadores

Dim Fila_Max As Integer 'Número de fila máxima

Dim COLUMNA_Categorías As Integer 'Número de columna categorías

Dim COLUMNA_Familias As Integer 'Número de columna Familia

Dim COLUMNA_Orden_Indi As Integer 'Número de columna de indicador, en principio no lo vamos a usar

Dim COLUMNA_Descripcion As Integer 'Número de columna de descripción

'Inicializamos las variables que vamos a usar

Fila_Ini = 5

COLUMNA_Orden_Indi = 10

COLUMNA_Orden = 1

COLUMNA_Descripcion = 7

COLUMNA_Indicadores = 6

COLUMNA_Categorías = 5

COLUMNA_Familias = 4

i = 0

'Inicializamos el número de fila

Fila = Fila_Ini - 1

'Antes de copiar tenemos que dar el formato correcto al texto

Call Formato_Hoja_Indicadores

'Activamos la hoja INDICADORES

ActiveWorkbook.Worksheets("INDICADORES").Activate

'Buscamos el número de fila máxima

Do While (Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Value <> "")

Fila = Fila + 1

Loop

```

Fila_Max = Fila

'Mostramos en esa celda el número de indicadores disponible
Worksheets("INDICADORES").Range("K2").Value = (Fila_Max - Fila_Ini)

'Inicializamos el número de fila
Fila = Fila_Ini

'Con las siguientes instrucciones limpiamos las hojas, con esto lo que nos aseguramos es de que _
cuando se copien los datos no se superpongan evitando problemas
ActiveWorkbook.Worksheets("RESUMEN").Activate
Range(Cells(5, 1), Cells(Fila_Max, 13)).Select
Selection.ClearContents
Range("P4:U8").Select
Selection.ClearContents
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
Selection.Borders(xlEdgeLeft).LineStyle = xlNone
Selection.Borders(xlEdgeTop).LineStyle = xlNone
Selection.Borders(xlEdgeBottom).LineStyle = xlNone
Selection.Borders(xlEdgeRight).LineStyle = xlNone
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Range("H5").Select
ActiveWorkbook.Worksheets("EX-ANTE").Activate
Range(Cells(5, 1), Cells(Fila_Max, 12)).Select
Selection.ClearContents
Range("G5").Select
ActiveWorkbook.Worksheets("IN-ITINERE").Activate
Range(Cells(5, 1), Cells(Fila_Max, 12)).Select
Selection.ClearContents
Range("G5").Select
ActiveWorkbook.Worksheets("EX-POST").Activate
Range(Cells(5, 1), Cells(Fila_Max, 12)).Select
Selection.ClearContents
Range("G5").Select

'Activamos hoja INDICADORES
ActiveWorkbook.Worksheets("INDICADORES").Activate

```

'Aplicamos el formato condicional a la hoja INDICADORES

Call Formato_Condicional_Indicadores(Fila_Max)

'Copiamos los indicadores en las diferentes hojas sin darles formato.

For Fila = Fila_Ini To Fila_Max

If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Orden_Indi) = "OK" Then

'Comprobamos que si quieres evaluar este indicador.

Application.CutCopyMode = True

Worksheets("INDICADORES").Cells(Fila, COLUMNA_Indicadores).Copy

Sheets("EX-POST").Select

Worksheets("EX-POST").Cells(Fila, COLUMNA_Indicadores).Select

ActiveSheet.Paste

Sheets("IN-ITINERE").Select

Worksheets("IN-ITINERE").Cells(Fila, COLUMNA_Indicadores).Select

ActiveSheet.Paste

Sheets("EX-ANTE").Select

Worksheets("EX-ANTE").Cells(Fila, COLUMNA_Indicadores).Select

ActiveSheet.Paste

Sheets("RESUMEN").Select

Worksheets("RESUMEN").Cells((Fila), COLUMNA_Indicadores).Select

ActiveSheet.Paste

Application.CutCopyMode = False 'Borra la memoria

i = i + 1

End If

Next Fila

'Copiamos las Categorías en las diferentes hojas sin darles formato.

For Fila = Fila_Ini To Fila_Max

If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Orden_Indi) = "OK" Then

'Comprobamos que si quieres evaluar este indicador.

Application.CutCopyMode = True

Worksheets("INDICADORES").Cells(Fila, COLUMNA_Categorías).Copy

Sheets("EX-POST").Select

Worksheets("EX-POST").Cells(Fila, COLUMNA_Categorías).Select

ActiveSheet.Paste

Sheets("IN-ITINERE").Select

Worksheets("IN-ITINERE").Cells(Fila, COLUMNA_Categorías).Select

```

ActiveSheet.Paste

Sheets("EX-ANTE").Select

Worksheets("EX-ANTE").Cells(Fila, COLUMNA_Categorias).Select

ActiveSheet.Paste

Sheets("RESUMEN").Select

Worksheets("RESUMEN").Cells((Fila), COLUMNA_Categorias).Select

ActiveSheet.Paste

Application.CutCopyMode = False

End If

Next Fila

'Copiamos las familias en las diferentes hojas sin darles formato.

For Fila = Fila_Ini To Fila_Max

If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Orden_Indi) = "OK" Then

'Comprobamos que si quieres evaluar este indicador.

Application.CutCopyMode = True

Worksheets("INDICADORES").Cells(Fila, COLUMNA_Familias).Copy

Sheets("EX-POST").Select

Worksheets("EX-POST").Cells(Fila, COLUMNA_Familias).Select

ActiveSheet.Paste

Sheets("IN-ITINERE").Select

Worksheets("IN-ITINERE").Cells(Fila, COLUMNA_Familias).Select

ActiveSheet.Paste

Sheets("EX-ANTE").Select

Worksheets("EX-ANTE").Cells(Fila, COLUMNA_Familias).Select

ActiveSheet.Paste

Sheets("RESUMEN").Select

Worksheets("RESUMEN").Cells((Fila), COLUMNA_Familias).Select

ActiveSheet.Paste

Application.CutCopyMode = False

End If

Next Fila

'Copiamos la descripción a la hoja RESUMEN

For Fila = Fila_Ini To Fila_Max

If Worksheets("INDICADORES").Cells(Fila, COLUMNA_Orden_Indi) = "OK" Then

Application.CutCopyMode = True

```

```

Worksheets("INDICADORES").Cells(Fila, COLUMNA_Descripcion).Copy
Sheets("RESUMEN").Select
Worksheets("RESUMEN").Cells((Fila), COLUMNA_Descripcion).Select
ActiveSheet.Paste
Application.CutCopyMode = False
End If
Next Fila

Q = MsgBox("SE VAN A EVALUAR " & i & " INDICADORES " & "DE LOS " & (Fila_Max - Fila_Ini) & "
POSIBLES ", vbInformation, " HERRAMIENTA EVALUACIÓN DE IMPACTOS SOCIALES ")

'Tenemos todos los elementos copiados en todas las hojas, ahora hay que dar formato a las
columnas.

'Con esto rellenos las celdas en blanco
ActiveWorkbook.Worksheets("RESUMEN").Activate
Call Blanco
ActiveWorkbook.Worksheets("EX-ANTE").Activate
Call Blanco
ActiveWorkbook.Worksheets("IN-ITINERE").Activate
Call Blanco
ActiveWorkbook.Worksheets("EX-POST").Activate
Call Blanco
ActiveWorkbook.Worksheets("INDICADORES").Activate
Call Blanco
Call Formato_Condicional_Indicadores(Fila_Max)
Call Validar_Datos_Indicador(Fila_Max)
Call Filtro_Indicadores
Call Ajuste_Tabla_INDICADORES(Fila_Max)
Call Contador_Blanco
'Dejamos seleccionado la celda J5, columna Orden
Range("J5").Select

'Llamamos a todas las subrutinas que nos formatean todas las hojas
N = Fila_Max
Call bordes_tabla_indicadores(N)
Call Formateo_Hoja
Call bordes_tabla_resumen(N)
Call Formateo_Hoja

```

Call Formato_Encabezado_Indicadores

Call Formato_Encabezado_Resumen

ActiveWorkbook.Worksheets("EX-ANTE").Activate

Call bordes_tabla_ExAnte_InItinere_ExPost(N)

Call Formateo_Hoja

Call Formato_Encabezado_ExAnte_InItinere_ExPost

Call Orden_Tres_Niveles_Ex_Ante(Fila_Max)

Call Filtro_Etapas

'Borra celdas sobrantes

Worksheets("EX-ANTE").Range(Cells((i + 5), 4), Cells(Fila_Max, 15)).Delete Shift:=xlUp

Call FormatoCondicional_EtapasProyecto

Call Validar_Datos(i)

Call Ajuste_Tabla_ETAPAS(i)

Rows(Fila_Max).Select

Selection.Delete Shift:=xlUp

Range("G5").Select 'Dejamos seleccionado la celda G5, primera columna para evaluar

ActiveWorkbook.Worksheets("IN-ITINERE").Activate

Call bordes_tabla_ExAnte_InItinere_ExPost(N)

Call Formateo_Hoja

Call Formato_Encabezado_ExAnte_InItinere_ExPost

Call Orden_Tres_Niveles_In_Itinere(Fila_Max)

Call Filtro_Etapas

'Borra celdas sobrantes

Worksheets("IN-ITINERE").Range(Cells((i + 5), 4), Cells(Fila_Max, 15)).Delete Shift:=xlUp

Call FormatoCondicional_EtapasProyecto

Call Validar_Datos(i)

Call Ajuste_Tabla_ETAPAS(i)

Rows(Fila_Max).Select

Selection.Delete Shift:=xlUp

Range("G5").Select 'Dejamos seleccionado la celda G5, primera columna para evaluar

ActiveWorkbook.Worksheets("EX-POST").Activate

Call bordes_tabla_ExAnte_InItinere_ExPost(N)

Call Formateo_Hoja

Call Formato_Encabezado_ExAnte_InItinere_ExPost

Call Orden_Tres_Niveles_Ex_Post(Fila_Max)


```

Call Filtro_Etapas
'Borra celdas sobrantes
Worksheets("EX-POST").Range(Cells((i + 5), 4), Cells(Fila_Max, 15)).Delete Shift:=xlUp
Call FormatoCondicional_EtapasProyecto
Call Validar_Datos(i)
Call Ajuste_Tabla_ETAPAS(i)
Rows(Fila_Max).Select
Selection.Delete Shift:=xlUp
Range("G5").Select 'Dejamos seleccionado la celda G5, primera columna para evaluar

ActiveWorkbook.Worksheets("RESUMEN").Activate
Call Orden_Tres_Niveles_Resumen(Fila_Max)
Call FormatoCondicional_Resumen
Call Filtro_Resumen
'Borra celdas sobrantes
Worksheets("RESUMEN").Range(Cells((i + 5), 4), Cells(Fila_Max, 15)).Delete Shift:=xlUp
Rows(Fila_Max).Select
Selection.Delete Shift:=xlUp
'Se ocultan las columnas I y K de esta hoja
Columns("I:I").EntireColumn.Hidden = True
Columns("K:K").EntireColumn.Hidden = True
'Dejamos seleccionado la celda H5
Range("H5").Select
'Aplicamos el formato a las celdas que contiene el contadore de indicadores
Call Formato_Contadores
'Ocultamos la columna I de la hoja INDICADORES
Columns("I:I").EntireColumn.Hidden = True
'Ajustamos el tamaños de las tablas
Call Ajuste_Tabla_RESUMEN(i)
'Dejamos la hoja Ex-ante activa
ActiveWorkbook.Worksheets("EX-ANTE").Activate
'Código para que no se muestre la ejecución de la macro
Application.ScreenUpdating = True

End Sub

```

9.2.15. Modulo_Seguridad

```
'*****  
****
```

CONTIENE LAS SUBROUTINAS PARA LA SEGURIDAD DEL ARCHIVO

```
'*****  
****
```

```
Sub Seguridad_Nueva()
```

```
    Dim resultado_nueva As String
```

```
    Worksheets("AYUDA").Activate
```

```
    resultado_nueva = InputBox("Introduce la nueva contraseña: ", "Seguridad")
```

```
    If resultado_nueva <> "" Then 'Si la valeur est différente de "" on affiche le résultat
```

```
        'MsgBox resultado_nueva
```

```
        Sheets("AYUDA").Unprotect 'Desprotegemos la hoja
```

```
        Sheets("AYUDA").Cells(1, 1).Value = resultado_nueva
```

```
        Cells(1, 1).Select
```

```
        With Selection.Font
```

```
            .ThemeColor = xlThemeColorDark1
```

```
            .TintAndShade = 0
```

```
        End With
```

```
        'Seleccionada todas las celdas y las bloquea
```

```
        Selection.Locked = True
```

```
        Selection.FormulaHidden = False
```

```
        'Seleccionada todas las celdas y las bloquea
```

```

Cells.SpecialCells(xlCellTypeBlanks).Select
Selection.Locked = False
'Pone contraseña a todas las celdas bloqueadas
ActiveSheet.Protect DrawingObjects:=True, Contents:=True, Scenarios:=True
ActiveSheet.EnableSelection = xlUnlockedCells
Cells(2, 2).Select
End If

MsgBox ("La contraseña es: " & Sheets("AYUDA").Cells(1, 1).Value) ' Esto se puede quitar cuando
se termine la herramienta

Sheets("AYUDA").Unprotect ' Desprotegemos la hoja hasta que terminemos por completo la
herramienta
End Sub

```

```

Sub Seguridad()
Dim resultado As String
resultado = InputBox("Introduce contraseña: ", "Seguridad")
If resultado <> Sheets("AYUDA").Cells(1, 1).Value Then
MsgBox ("Contraseña incorrecta, la herramienta se cierra")
ActiveWorkbook.Save 'Guarda el libro activo
Application.Quit 'Cierra la aplicación
Else
MsgBox ("Contraseña correcta")
End If
End Sub

```

9.2.16. Ocultar_Mostrar_Descripcion

```

'*****
*****

CONTIENE LAS SUBROUTINAS PARA OCULTAR O MOSTRAR LA DESCRIPCIÓN EN LA
HOJA RESUMEN

'*****
*****

```

```

Sub Ocultar_Descripcion()
'Mostrar columna descripción en la hoja resumen
    Application.ScreenUpdating = False
    On Error Resume Next
    Columns("G:G").Select
    Selection.EntireColumn.Hidden = True
    Range("D5").Select
End Sub

```

```

Sub Mostra_Descripcion()
'Ocultar columna descripción en la hoja resumen
    Application.ScreenUpdating = False
    On Error Resume Next
    Columns("G:G").Select
    Selection.EntireColumn.Hidden = False
    Range("D5").Select
End Sub

```

9.2.17. Orden_3_Niveles

```

'*****
*****

```

'Subrutina que ordena las tablas teniendo en cuenta _
 Familia -> Categoría -> Indicador ordenando alfabéticamente _
 primero Familia luego categoría y por último los indicadores.

```

'*****
*****

```

```

Sub Orden_Tres_Niveles_Ex_Ante(Fila_Max)
    Application.ScreenUpdating = False
    On Error Resume Next
    Fila_Ini = 5

```

```

Range("D5").Select

'Lo primero que tenemos que hacer es deshabilitar los posibles filtros que el usuario haya
introducido

    If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter

    If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData


Range("D5").Sort Key1:=Range("D5"), Order1:=xlAscending, Key2:=Range( _
"E5"), Order2:=xlAscending, Key3:=Range("F5"), Order3:=xlAscending, _
Header:=xlYes

    With ActiveWorkbook.Worksheets("EX-ANTE").Sort
        .SetRange Range("A5:N53")

        .SetRange Range(Cells(Fila_Ini, 1), Cells(Fila_Max, 15))

        'Tenemos que tener cuenta que el rango es variable

        .Header = xlNo

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With
End Sub

*****
*****

'Subrutina que ordena las tablas teniendo en cuenta Familia -> Categoría ->
Indicador ordenando alfabéticamente primero Familia luego categoría y por último
los indicadores.

!*****
*****

Sub Orden_Tres_Niveles_In_Itinere(Fila_Max)

    Application.ScreenUpdating = False

    On Error Resume Next

    Fila_Ini = 5 Range("D5").Select

```

'Lo primero que tenemos que hacer es deshabilitar los posibles filtros que el usuario haya introducido

```
        If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter

        If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData

Range("D5").Sort Key1:=Range("D5"), Order1:=xlAscending, Key2:=Range( _
"E5"), Order2:=xlAscending, Key3:=Range("F5"), Order3:=xlAscending, _
Header:=xlYes

        With ActiveWorkbook.Worksheets("IN-ITINERE").Sort
            .SetRange Range("A5:N53")

            .SetRange Range(Cells(Fila_Ini, 1), Cells(Fila_Max, 15))

            'Tenemos que tener cuenta que el rango es variable

            .Header = xlNo

            .MatchCase = False

            .Orientation = xlTopToBottom

            .SortMethod = xlPinYin

            .Apply
        End With
    End Sub
```


'Subrutina que ordena las tablas teniendo en cuenta Familia -> Categoría -> Indicador ordenando alfabéticamente primero Familia luego categoría y por último los indicadores.

Sub Orden_Tres_Niveles_Ex_Post(Fila_Max)

```
        Application.ScreenUpdating = False

        On Error Resume Next

        Fila_Ini = 5

        Range("D5").Select

        'Lo primero que tenemos que hacer es deshabilitar los posibles filtros que el usuario haya introducido

        If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter

        If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData
```

```

Range("D5").Sort Key1:=Range("D5"), Order1:=xlAscending, Key2:=Range( _
"E5"), Order2:=xlAscending, Key3:=Range("F5"), Order3:=xlAscending, _
Header:=xlYes

With ActiveWorkbook.Worksheets("EX-POST").Sort
    .SetRange Range("A5:N53")
    .SetRange Range(Cells(Fila_Ini, 1), Cells(Fila_Max, 15))
    .Tenemos que tener cuenta que el rango es variable
    .Header = xlNo
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply
End With
End Sub

'*****
'*****

'Subrutina que ordena las tablas teniendo en cuenta Familia -> Categoria ->
Indicador ordenando alfabeticamente primero Familia luego categoria y por ultimo
los indicadores.

'*****
'*****

Sub Orden_Tres_Niveles_Resumen(Fila_Max)
    Application.ScreenUpdating = False
    On Error Resume Next
    Fila_Ini = 5
    Range("D5").Select

    'Lo primero que tenemos que hacer es deshabilitar los posibles filtros que el usuario haya
    introducido

    If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter
    If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData
    Range("D5").Sort Key1:=Range("D5"), Order1:=xlAscending, Key2:=Range( _
"E5"), Order2:=xlAscending, Key3:=Range("F5"), Order3:=xlAscending, _
Header:=xlYes

    With ActiveWorkbook.Worksheets("RESUMEN").Sort
        .SetRange Range("A5:N53")
        .SetRange Range(Cells(Fila_Ini, 1), Cells(Fila_Max, 15))

```

```

        'Tenemos que tener cuenta que el rango es variable

        .Header = xlNo

        .MatchCase = False

        .Orientation = xlTopToBottom

        .SortMethod = xlPinYin

        .Apply

    End With

End Sub

'*****

'*****

'Subrutina que ordena las tablas teniendo encuesta Familia -> Categoria ->
Indicador ordenando alfabeticamente primero Familia luego categoria y por ultimo
los indicadores.

'*****
'*****

Sub Orden_Tres_Niveles_Indicadores(Fila_Max)

    Application.ScreenUpdating = False

    On Error Resume Next

    Fila_Ini = 5

    Range("D5").Select

    'Lo primero que tenemos que hacer es deshabilitar los posibles filtros que el usuario haya
    introducido

        If ActiveSheet.AutoFilterMode = True Then Selection.AutoFilter

        If ActiveSheet.FilterMode = True Then ActiveSheet.ShowAllData

    Range("D5").Sort Key1:=Range("D5"), Order1:=xlAscending, Key2:=Range( _
    "E5"), Order2:=xlAscending, Key3:=Range("F5"), Order3:=xlAscending, _
    Header:=xlYes

        With ActiveWorkbook.Worksheets("INDICADORES").Sort

            .SetRange Range("A5:N53")

            .SetRange Range(Cells(Fila_Ini, 1), Cells(Fila_Max, 15))

            'Tenemos que tener cuenta que el rango es variable

            .Header = xlNo

            .MatchCase = False

            .Orientation = xlTopToBottom

            .SortMethod = xlPinYin

```



```

        .Apply
    End With
End Sub

```

9.2.18. TablaValoracionEtapas

```

'*****
*****

```

CONTIENE LAS RUTINAS PARA CREAR Y RELLENAR LA TABLA DE VALORACIÓN DE ETAPAS

```

'*****
*****

```

' Macro grabada para la creación de la tabla de valoración de etapas

```

'*****
*****

```

```

Sub Tabla_Valoracion_Etapas()
    Range("R5").Select
    ActiveCell.FormulaR1C1 = "EX-ANTE"
    Range("S5").Select
    ActiveCell.FormulaR1C1 = "IN-ITINERE"
    Range("T5").Select
    ActiveCell.FormulaR1C1 = "EX-POST"
    Range("Q6").Select
    ActiveCell.FormulaR1C1 = "VALORACIÓN FINAL"
    Range("Q6:Q7").Select
    With Selection
        .HorizontalAlignment = xlCenter
        .VerticalAlignment = xlCenter
        .WrapText = True
        .Orientation = 0
        .AddIndent = False
        .IndentLevel = 0
    End With
End Sub

```

```

        .ShrinkToFit = False
        .ReadingOrder = xlContext
        .MergeCells = False
    End With
    Selection.Merge
    Columns("Q:Q").ColumnWidth = 12.29
    Range("Q5:T5").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlEdgeRight).LineStyle = xlNone
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Range("P6:P7").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    Selection.Borders(xlEdgeBottom).LineStyle = xlNone
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Range("Q6:T7").Select

```

```

Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeTop)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
Range("Q6:Q7,Q6:Q7,Q5").Select
Range("Q5").Activate
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0

```

```

        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Range("R5:R7").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    With Selection.Borders(xlEdgeLeft)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Range("S5:S7").Select

```

```

Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Selection.Borders(xlEdgeTop).LineStyle = xlNone
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
With Selection.Borders(xlEdgeRight)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Selection.Borders(xlInsideVertical).LineStyle = xlNone
Range("T5:T7").Select
Selection.Borders(xlDiagonalDown).LineStyle = xlNone
Selection.Borders(xlDiagonalUp).LineStyle = xlNone
With Selection.Borders(xlEdgeLeft)
    .LineStyle = xlContinuous
    .ColorIndex = 0
    .TintAndShade = 0
    .Weight = xlThin
End With
Selection.Borders(xlEdgeTop).LineStyle = xlNone
With Selection.Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .ColorIndex = 0

```

```

        .TintAndShade = 0
        .Weight = xlThin
    End With
    With Selection.Borders(xlEdgeRight)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Range("R4:T4").Select
    Selection.Borders(xlDiagonalDown).LineStyle = xlNone
    Selection.Borders(xlDiagonalUp).LineStyle = xlNone
    Selection.Borders(xlEdgeLeft).LineStyle = xlNone
    Selection.Borders(xlEdgeTop).LineStyle = xlNone
    With Selection.Borders(xlEdgeBottom)
        .LineStyle = xlContinuous
        .ColorIndex = 0
        .TintAndShade = 0
        .Weight = xlThin
    End With
    Selection.Borders(xlEdgeRight).LineStyle = xlNone
    Selection.Borders(xlInsideVertical).LineStyle = xlNone
    Selection.Borders(xlInsideHorizontal).LineStyle = xlNone
    Range("Q6:Q7,R5,S5,T5").Select
    Range("T5").Activate
    With Selection.Interior
        .Pattern = xlSolid
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorAccent5
        .TintAndShade = 0.399975585192419
        .PatternTintAndShade = 0
    End With
    Selection.Font.Bold = True

```

Range("R6:R7").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.WrapText = True

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Merge

Range("S6:S7").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.WrapText = True

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

.ReadingOrder = xlContext

.MergeCells = False

End With

Selection.Merge

Range("T6:T7").Select

With Selection

.HorizontalAlignment = xlCenter

.VerticalAlignment = xlCenter

.WrapText = True

.Orientation = 0

.AddIndent = False

.IndentLevel = 0

.ShrinkToFit = False

```

        .ReadingOrder = xlContext
        .MergeCells = False
    End With
    Selection.Merge
End Sub

```

9.2.19. ValidarDatos

```

'*****
'*****

```

' Macro para la validación de los datos, los valores de entrada han de estar comprendidos entre 1-6. Donde 5 corresponde al valor de menor impacto y 1 al valor de mayor impacto. Y 6 quiere decir que no se va a evaluar ese indicador. Para la utilización de la macro, le tenemos que introducir como parámetro el valor de filas máximo (Fila_Max)

'Esta subrutina solo se aplicará a las hojas en las que vamos a introducir valores manualmente: "EX-ANTE" "IN-ITINERE" "EX-POST" para ello debemos activas primero la hoja

'El valor 0, indica que no se va evaluar el parametro para el indicador.

```

'*****
'*****

```

```

Sub Validar_Datos(Fila_Max)

```

```

    Application.ScreenUpdating = False

```

```

    On Error Resume Next

```

```

    Range(Cells(5, 7), Cells(Fila_Max, 10)).Select

```

```

    With Selection.Validation

```

```

        .Delete

```

```

        .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _

```

```

        xlBetween, Formula1:="=AYUDA!$K$26:$K$31"

```

```

        .IgnoreBlank = True

```

```

        .InCellDropdown = True

```

```

        .InputTitle = ""

```

```

        .ErrorTitle = "Valor no válido"

```

```

        .InputMessage = ""

```

```

        .ErrorMessage = _

```

"Introduzca un valor comprendido entre 1 y 5. Para más información consulte la ayuda de la herramienta"

```

        .ShowInput = True

```



```

        .ShowError = True
    End With
End Sub

Sub Validar_Datos_Indicador(Fila_Max)
    Application.ScreenUpdating = False
    On Error Resume Next
    Range(Cells(5, 10), Cells(Fila_Max - 1, 10)).Select
        With Selection.Validation
            .Delete
            .Add Type:=xlValidateList, AlertStyle:=xlValidAlertStop, Operator:= _
xlBetween, Formula1:="=AYUDA!$P$17:$P$18"
            .IgnoreBlank = True
            .InCellDropdown = True
            .InputTitle = ""
            .ErrorTitle = ""
            .InputMessage = ""
            .ErrorMessage = _
"Introduzca un valor comprendido entre 1 y 5. Para más información consulte la ayuda de la
herramienta"
            .ShowInput = False
            .ShowError = False
        End With
End Sub

'*****
*****

```

Como se puede comprobar, la herramienta desarrollada es compleja. Con este anexo lo que se quiere conseguir es dar el código en papel a los usuarios, para que puedan trabajarlo pudiendo obtener mejoras de la herramienta.