

# MANUAL DE FORMULARIOS EXCEL

## (USERFORMS y otros)

*(Para versiones xlsx)*

### INDICE DE CONTENIDOS EN ESTA VERSIÓN DEMO

En esta versión se incluye el **Índice de Contenidos completo y el desarrollo de algunos de sus capítulos** a cuyos temas se puede acceder desde el mismo Índice.

#### **CAPÍTULOS DESARROLLADOS: 3, 5, 10 Y 15**

El manual en su versión completa presenta **160 páginas, con más de una macro de ejemplo en cada tema.** (por ejemplo 40 macros en tema Combobox y ListBox).

Además, Imágenes, notas aclaratorias y vínculos a las distintas hojas.

En versión completa se acompañan 10 libros Excel con todos los ejemplos que se desarrollan en este manual.

---

*por Elsa Matilde Meyer*

1. INTRODUCCIÓN
2. LOS MSGBOX
  - 2.1. Sintaxis de un MsgBox
  - 2.2. Botones o *Buttons* en un MsgBox
  - 2.3. Ejemplos de MsgBox con su sintaxis
  - 2.4. Cómo indicar saltos de línea en un mensaje
  - 2.5. Programando MsgBox
  - 2.6. MsgBox con botones personalizados
3. LOS INPUTBOX
  - 3.1. La sintaxis de un InputBox
  - 3.2. Ejemplos de InputBox con su sintaxis
  - 3.3. Programando InputBox
  - 3.4. Cancelar un InputBox: cómo detectarlo
  - 3.5. Seleccionar rango desde un InputBox
4. USO DE CONTROLES
  - 4.1. Tipos de controles
  - 4.2. Como colocar controles en una hoja Excel
  - 4.3. Barra ActiveX
  - 4.4. Barra Formularios
  - 4.5. Cómo reconocer si se trata de un control 'ActiveX' o de 'Formulario'?
  - 4.6. Conocer rutina asociada a un control
5. TIPOS DE FORMULARIOS
  - 5.1. Herramienta 'Formulario' de Excel
  - 5.2. Formulario diseñado en hoja Excel.
  - 5.3. Userforms.
6. LOS USERFORMS
  - 6.1. Tipos de Userforms
  - 6.2. Crear un Userform
  - 6.3. Propiedades y eventos
  - 6.4. Dibujar controles en un Userform
    - 6.4.1. Asignar iguales propiedades a controles
    - 6.4.2. Asignar nombre a los controles
    - 6.4.3. Asignar orden de entrada a los controles.
  - 6.5. Cómo ejecutar o Llamar a un Userform

- 6.6. Cómo programar el formulario y sus controles
  - 6.7. Propiedades del Userform
    - 6.7.1. Lista detallada de Propiedades
  - 6.8. Eventos del Userform
  - 6.9. Ubicación del userform en la pantalla.
  - 6.10. Userform siempre centrado
  - 6.11. Inmovilizar un Userform
  - 6.12. Ocultar un Userform
  - 6.13. Cerrar un Userform
  - 6.14. Acceder al libro desde un Userform abierto
  - 6.15. Inhabilitar botón de Cerrar
  - 6.16. Quitar botón de Cerrar (X)
  - 6.17. Agregar botones de Minimizar y Maximizar en un Userform
7. EL CONTROL LABEL O ETIQUETA
- 7.1. Características de un control Label
  - 7.2. Propiedades y Eventos de un Label
  - 7.3. Inicializando un control Label
  - 7.4. Uso de Label como contador
  - 7.5. Evento MouseMove: Mostrar mensaje al pasar el mouse por el control
  - 7.6. Limpieza de Label
  - 7.7. Color parpadeante en un Label
8. EL CONTROL TEXTBOX o CUADROS DE TEXTO
- 8.1. Características de un control TextBox
  - 8.2. Propiedades y Eventos de un TextBox
  - 8.3. Inicializando un control Textbox
  - 8.4. Textbox con datos de otro Userform
  - 8.5. Asignando orden o foco a los controles
  - 8.6. TextBox para contraseñas
  - 8.7. Cómo y cuándo controlar datos de ingreso
    - 8.7.1. Longitud máxima para un campo
  - 8.8. Evento KEYPRESS para controlar caracteres ingresados.
    - 8.8.1. Caracteres numéricos.
    - 8.8.1. Caracteres especiales para números.
    - 8.8.1. Caracteres alfabéticos.

- 8.9. Funciones de validación
  - 8.9.1. Controlar campos numéricos
  - 8.9.2. Controlar caracteres o texto
- 8.10. Validar formatos fecha
- 8.11. Ingresar fechas sin separadores
- 8.12. Funciones de conversión. Pasar datos a hoja
  - 8.12.1. Convertir texto en mayúsculas (o minúsculas)
  - 8.12.2. Convertir dato numérico a formato moneda.
  - 8.12.3. Evitar campos vacíos al momento de guardar
- 8.13. Seleccionar rangos con RefEdit
- 8.14. Limpieza de Textbox
  - 8.14.1. ¿Cómo limpiar todos los controles sin repetir las instrucciones?
- 8.15. Selección del texto completo en un Textbox
- 8.16. Simular marcas de agua en un TextBox
- 9. Los objetos COMBOBOX y LISTBOX
  - 9.1. Características de los Combobox/ListBox
  - 9.2. Propiedades y Eventos de los Combobox y ListBox
  - 9.3. Ejemplos de Principales propiedades de Combobox/ListBox
    - 9.3.1. Asignar rango a un Combo o Lista
    - 9.3.2. Desplegar la lista de un Combobox
    - 9.3.3. Celda asociada al elemento seleccionado
    - 9.3.4. Autocompletar datos en un control
    - 9.3.5. Permitir el ingreso de datos que no están en la lista
    - 9.3.6. Seleccionar varios registros de una lista (solo para LISTBOX)
    - 9.3.7. Mostrar lista con botones de opción (solo para LISTBOX)
    - 9.3.8. Mover elementos desde un Listbox a un Textbox
  - 9.4. Actualizar rango asociado a un Combobox/ListBox
    - 9.4.1. Rango de datos fijo.
    - 9.4.2. Rango dinámico (4 casos)
  - 9.5. Llenar un combobox sin datos duplicados.
  - 9.6. Cómo llenar Combobox/Listbox de varias col
  - 9.7. Controles dependientes (Combos y/o Listas)
  - 9.8. Trabajar con listas o cuadros de varias columnas
    - 9.8.1. Cantidad de columnas

- 9.8.2. ¿Qué valor guardar o mostrar en controles de varias columnas?
- 9.8.3. Ancho de columnas
- 9.8.4. Títulos de columnas
- 9.8.5. ListBox con más de 10 columnas.
- 9.9. Trabajar con 2 (o más) Listas paralelas
- 9.10. Mostrar elemento seleccionado de lista
- 9.11. Seleccionar varios registros de un ListBox
- 9.12. Quitar elementos de una lista
- 9.13. Reordenar elementos de una lista
- 9.14. Contar elementos de una lista
- 9.15. Pasar datos de un Combobox o Lista a la hoja
- 9.16. Mostrar datos filtrados en un control Combobox o Listbox
- 9.17. Limpieza de Combobox/ListBox
- 9.18. Trazar línea para separar elementos de un ListBox .
- 10. El objeto FRAME O MARCO
  - 10.1. Características de un control Frame
  - 10.2. Propiedades y Eventos de un control Frame
  - 10.3. Agrandar un Frame a medida que se agregan datos
  - 10.4. Inhabilitar todos los controles dentro de un Frame.
- 11. El objeto OPTIONBUTTON O BOTONES DE OPCIÓN
  - 11.1. Características de los OptionButton
  - 11.2. Propiedades y Eventos de los Optionbutton
  - 11.3. Reconocer el OptionButton seleccionado
  - 11.4. Ejecutar rutina al seleccionar un botón de opción
  - 11.5. Atajo de teclado para seleccionar un botón de opción
  - 11.6. Limpieza de OptionButtons
- 12. El objeto CHECKBOX o CASILLAS DE VERIFICACIÓN
  - 12.1. Características de los Checkbox
  - 12.2. Propiedades y Eventos de un CheckBox
  - 12.3. Reconocer los CheckBox seleccionados
  - 12.4. Código para selección de CheckBox
  - 12.5. Atajo de teclado para tildar un checkbox (acelerator)
  - 12.6. Limpieza de CheckBox
  - 12.7. Combinar Option y Checkbox

13. El objeto COMMANDBUTTON o BOTON DE COMANDO
  - 13.1. Características de un Botón de Comando
  - 13.2. Propiedades y Eventos de los CommandButton
  - 13.3. Ejemplos del uso de un Botón de Comando
  - 13.4. Botones con imágenes
  - 13.5. Atajos de teclado para ejecutar botones.
  - 13.6. Inhabilitar botones según condición
14. FORMULARIOS EN HOJA EXCEL
  - 14.1. Características de estos formularios
  - 14.2. Usar solo celdas habilitadas
  - 14.3. Cómo volcar datos a una hoja resumen
  - 14.4. Cómo limpiar un formulario en hoja
  - 14.5. Uso de fórmulas
  - 14.6. Uso de celdas con validación
  - 14.7. Programando controles
15. UF: APERTURA CON CONTRASEÑA
  - 15.1. Ejemplo de Userform de ingreso
  - 15.2. Asignando propiedades
  - 15.3. Programando un Userform de ingreso
  - 15.4. Llamando al Userform de ingreso
16. UF: APERTURA CON VALORES PREDETERMINADOS
  - 16.1. Ejemplo de Userform con valores iniciales
  - 16.2. Asignar valores de inicio a controles
  - 16.3. Llamando al Userform con valores iniciales
17. UF: INGRESO DE DATOS CON VALIDACIÓN
  - 17.1. Diseño formulario con Validación
  - 17.2. Vuelco de datos VALIDADOS O CONVERTIDOS, a hoja
  - 17.3. Volcar datos como hipervínculos. Buscar vinculación.
18. UF: ALTAS, BAJAS Y CAMBIOS A BASE DE DATOS
  - 18.1. Formulario para Altas, Bajas y Modificaciones
  - 18.2. Permitir ingreso de datos en Combobox
  - 18.3. Botones para Altas, Bajas y Cambios
  - 18.4. Botones para navegar por la base
19. UF: CON CONTROL MULTIPAGINA o 'MULTIPAGE'

- 19.1. Consideraciones en formularios Multipage
- 19.2. Propiedades de un Multipage y sus *Pages*
- 19.3. Eventos de un Multipage
- 19.4. Diseño de un formulario Multipage
- 19.5. Añadir, quitar o mover pestañas
- 19.6. Programando botones comunes
- 19.7. Programando botones de avance
- 19.8. Multipage de colores
- 20. UF: FORMULARIO DE AYUDA
  - 20.1. Diseño de un formulario de Ayuda
  - 20.2. Configurando el formulario de Ayuda
  - 20.3. Programando un formulario de Ayuda
- 21. CONTROLES ESPECIALES
  - 21.1. CONTROL SPINBUTTON o Botón de nro
    - 21.1.1. Propiedades y Eventos de un SpinButton
    - 21.1.2. Ejemplo de código para un SpinButton
  - 21.2. CONTROL SCROLLBAR o Barra de desplazamiento
    - 21.2.1. Propiedades y Eventos de un ScrollBar
    - 21.2.2. Ejemplo de código para un ScrollBar
  - 21.3. CONTROL IMAGE o de Imagen
    - 21.3.1. Propiedades y Eventos de un control Image
    - 21.3.2. Ejemplo de código para un control Image
    - 21.3.3. Control con Imagen animada
    - 21.3.4. Gráficos- Tabla Dinámica –Tabla de datos como imagen
  - 21.4. CONTROL REFEDIT
    - 21.4.1. Propiedades y Eventos de un control RefEdit
    - 21.4.2. Ejemplo de código para un control RefEdit
  - 21.5. CONTROL TOGGLEBUTTON o Botón de alternar
    - 21.5.1. Propiedades y Eventos de un control ToggleButton
    - 21.5.2. Ejemplo de código para un control ToggleButton
  - 21.6. CONTROL TABSTRIP o Barra de tabulaciones
    - 21.6.1. Propiedades y Eventos de un TablStrip
    - 21.6.2. Ejemplo de código para un control TabStrip
  - 21.7. Los CONTROLES CALENDAR Y DATAPICKER

21.7.1. Propiedades y Eventos de controles DTPicker y Calendar

21.7.2. Ejemplos de código para controles Calendar y DTPicker:

22. PROPIEDADES DE LOS CONTROLES

22.1. Propiedadesde USERFORM Y CONTROLES

---

## 1. INTRODUCCIÓN

Al trabajar con Excel no podemos dejar de mencionar los **'formularios'**.

Estos pueden ser diseñados en la misma hoja (con formatos más o menos elaborados) o diseñados en los llamados **Userforms**, que son formularios que se 'muestran' por encima de las hojas.

Los usos más frecuentes, son:

- las **aplicaciones comerciales** (como Facturación, Cobranzas, Libro Compras),
- **aplicaciones contables** (Contabilidad, LibroBancos, Inventarios, Gestión),
- **docentes o escolares** (Gestor Docente, Cursos y Exámenes)
- y **tantas otras tareas** que requieren de un formato con diseño.

En este manual trabajaremos en *formato de hoja* pero especialmente con *Userforms*.

También trataremos el uso de otras **ventanas o cuadros de diálogo**, las que nos informan del estado de algún proceso o nos solicitan el ingreso de algún dato para completar alguna tarea. Esto son los **MsgBox** e **InputDialog**.

Separaremos entonces los temas en:

- 1- **Msgbox e Inputbox**: propios de la aplicación Excel con los que podremos recibir información o introducirlos en algún proceso.
- 2- Herramienta **Formulario**: en estos modelos trabajamos directamente con un formulario simple que se activa desde las herramientas Excel.
- 3- Diseño de formulario en **Hoja**. Son los que dibujamos en una hoja Excel. Se utiliza generalmente celdas con funciones, validación de datos y/o controles dibujados para el registro de información.
- 3- **Userforms**: son los formularios creados por el usuario, desde el Editor de Macros.

En este manual nos abocaremos especialmente, a partir del capítulo 6, a todo lo relacionado con los Userforms.

**Nota:** En los libros adjuntos se encuentran todos los ejemplos explicados en este manual, listos para usar.

Aprenderemos las **características** de cada uno de ellos y además:

**a- Cuándo utilizarlos y cómo diseñarlos.**

**b- Ventajas y desventajas** de cada uno de los métodos empleados: celdas con funciones, con validación, con controles, con Userforms.

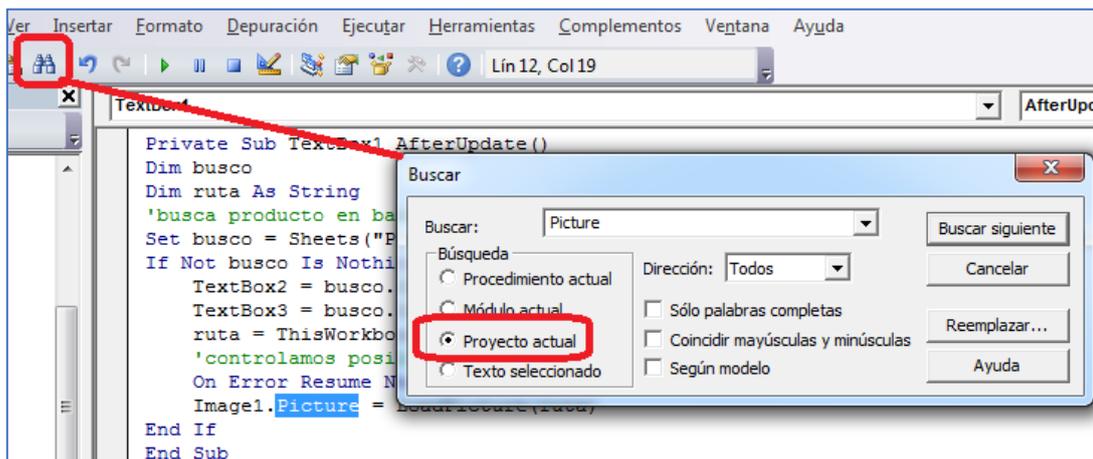
**c- Uso de Controles. Sus Propiedades y Eventos.**

**d- Distintos tipos de Userforms.**

**Recomendación para el estudio de este manual:** trabajar con un libro propio e ir repitiendo los pasos y luego comparar con el modelo del libro *Ejemplos\_Userforms*.

Otros recursos para el estudio del tema:

- 1- **Utilizar el buscador:** para encontrar un tema en particular tal como se muestra en la imagen :



**2- Utilizar los modelos entregados en los libros de ejemplos.**

Cuando el tema lo menciona, observar los ejemplos en los libros Excel que se entregan con este manual.

**3- Observar los tips y las notas para distintas versiones Excel.**

Cuando existen diferencias entre distintas versiones de Excel, aparecerá una nota o comentario.

**4- Seguir los vínculos a temas complementarios.**

Observar vínculos a otros capítulos que amplían o completan el tema tratado.

## 2. LOS MSGBOX

### 2.1. SINTAXIS DE UN MSGBOX

### 2.2. BOTONES O *BUTTONS* EN UN MSGBOX

### 2.3. EJEMPLOS DE MSGBOX CON SU SINTAXIS

### 2.4. CÓMO INDICAR SALTOS DE LÍNEA EN UN MENSAJE

### 2.5. PROGRAMANDO MSGBOX

### 2.6. MSGBOX CON BOTONES PERSONALIZADOS

### 3. LOS INPUTBOX

Los **InputBox** o **CUADROS DE DIÁLOGO** son los cuadros que se utilizan a la espera del ingreso de datos por parte del usuario.

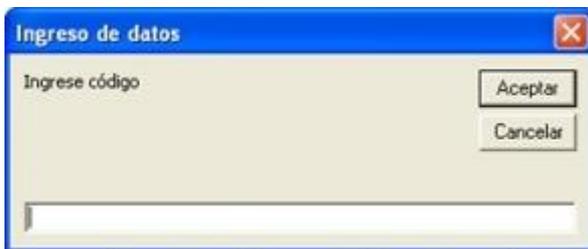
A diferencia del cuadro anterior (**MsgBox**) donde el usuario solo puede presionar ciertos botones, los InputBox además de enviarnos información, presentan un **cuadro de texto** esperando que escribamos algo en él.

También podrá mostrarse un valor predeterminado y dejar que el usuario haga clic en alguno de los botones devolviendo un valor (tipo String) con el contenido de ese cuadro de texto.

**Importante:** *si la aplicación requiere el ingreso de varios datos, será mejor utilizar un Userform para agilizar el proceso.*

#### Ejemplos:

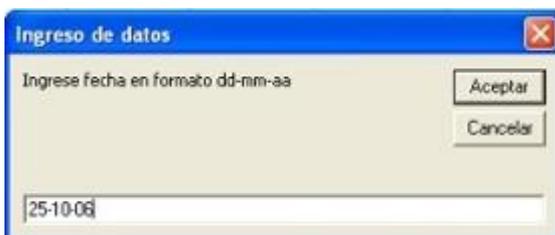
1 - Solicitar una clave de ingreso



2 - Solicitar un criterio para un filtro. Se dejó como predeterminada la primera opción.



3 - Solicitar fecha de actualización, indicando con qué formato debe ser ingresada.

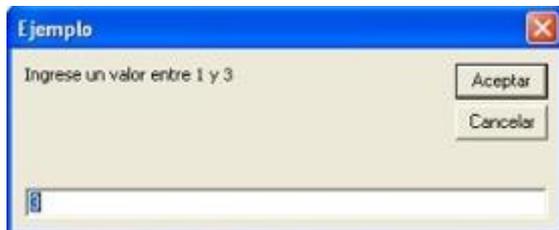


### 3.1. LA SINTAXIS DE UN INPUTBOX

#### SINTAXIS:

**InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context])**

#### Ejemplo:



#### Argumentos:

**Prompt: Requerido.** Expresión de cadena que se muestra como *mensaje* en el cuadro de diálogo. La longitud máxima de **prompt** es de aproximadamente 1024 caracteres, según el ancho de los caracteres utilizados. Si **prompt** consta de más de una línea, puede separarlos utilizando un carácter de retorno de carro (**Chr(13)**), un carácter de avance de línea (**Chr(10)**) o una combinación de los caracteres de retorno de carro-avance de línea (**Chr(13)** y **Chr(10)**) entre cada línea y la siguiente.

**Title: Opcional.** Expresión de cadena que se muestra en la barra de título del cuadro de diálogo. Si omite **title**, en la barra de título se coloca el nombre de la aplicación.

**Default: Opcional.** Expresión de cadena que se muestra en el cuadro de texto como respuesta predeterminada cuando no se suministra una cadena. Si omite **default**, se muestra el cuadro de texto vacío

**Xpos: Opcional.** Expresión numérica que especifica, en twips, la distancia en sentido horizontal entre el borde izquierdo del cuadro de diálogo y el borde izquierdo de la pantalla. Si se omite **xpos**, el cuadro de diálogo se centra horizontalmente.

**Ypos: Opcional.** Expresión numérica que especifica, en twips, la distancia en sentido vertical entre el borde superior del cuadro de diálogo y el borde superior de la pantalla. Si se omite **ypos**, el cuadro de diálogo se coloca a aproximadamente un tercio de la altura de la pantalla, desde el borde superior de la misma.

**Helpfile: Opcional.** Expresión de cadena que identifica el archivo de Ayuda que se utilizará para proporcionar ayuda interactiva para el cuadro de diálogo. Si se especifica **helpfile**, también deberá especificarse **context**.

**Context: Opcional.** Expresión numérica que es el número de contexto de Ayuda asignado por el autor al tema de Ayuda correspondiente. Si se especifica **context**, también deberá especificarse **helpfile**.

#### Comentarios

Cuando se especifica tanto **helpfile** como **context**, el usuario puede presionar F1 (Windows) o Ayuda (Macintosh) para ver el tema de **Ayuda** correspondiente a **context**.

Si el usuario hace clic en **Aceptar** o presiona Entrar, la función **InputBox** devuelve lo que haya en el cuadro de texto. Si el usuario hace clic en **Cancelar**, la función devuelve una cadena de caracteres de longitud cero ("").

*Para el ejemplo de la imagen se utilizaron estos argumentos:*

**Prompt** = "Ingrese un valor entre 1 y 3"

**Title** = "Ejemplo"

**Default** = 3

**Nota:**

Si deseamos omitir algún argumento, debemos incluir el delimitador de coma correspondiente.

*importe = InputBox("Ingrese importe", "Datos", , 50, 50)*

### 3.2. EJEMPLOS DE INPUTBOX CON SU SINTAXIS

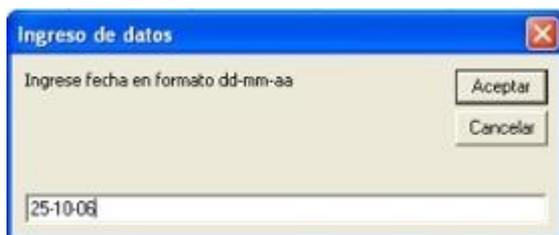
A continuación veremos algunos ejemplos de estos cuadros de dialogo y su sintaxis correspondiente en una rutina VBA:

En estos *formularios* o *cuadros de mensaje*, **siempre** se requiere de una **variable** para guardar el dato ingresado por parte del usuario.

*variable = InputBox(.....)*

**Ejemplo 1:** *Dim miFecha As Date*

*miFecha = InputBox("Ingrese fecha en formato dd-mm-aa", "Ingreso de datos")*



Notar que solo se ingresaron los 2 primeros argumentos: Prompt o texto y Title o título. Los botones de opción son los predeterminados por omisión.

**Ejemplo 2:** *Dim valor As Byte*

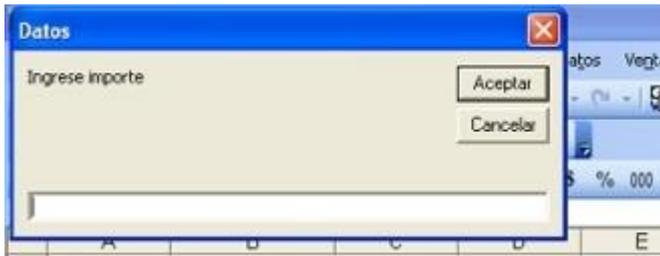
*valor = InputBox("Ingrese un valor entre 1 y 3", "Ejemplo", 3)*



Observar que en este ejemplo se programa como valor por **'default'** el número 3.

**Ejemplo 3: Dim importe**

`importe = InputBox("Ingrese importe", "Datos", , 50, 50)`



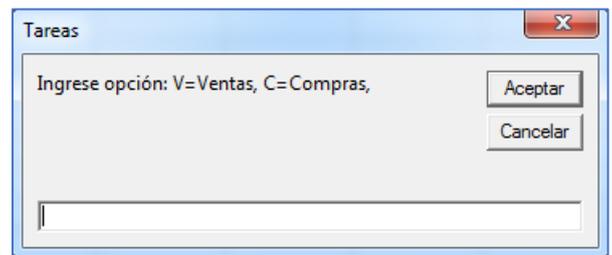
Observar que en este ejemplo, se omite el argumento **'default'** (agregando la coma) y se agregaron los de **posición**, resultando un cuadro ubicado en el **extremo superior izquierdo**.

### 3.3. PROGRAMANDO INPUTBOX

**¿Qué podemos programar en estos cuadros de diálogo?**

Cada uno de los argumentos como hemos visto en los ejemplos del tema anterior: **posición, texto y botones a mostrar**.

También las distintas acciones según cuál botón el usuario haya presionado o según el valor ingresado en el cuadro de texto.



**Ejemplo:**

```

Sub opcion_InputBox()
Dim opcion
opcion = InputBox("Ingrese opción: V=Ventas, C=Compras, ", "Tareas")
If opcion = "V" Then
    MsgBox "Trabjará con Ventas"
    '.... otras instrucciones
Elseif opcion = "C" then
    MsgBox "Trabjará con Compras"
    '.... otras instrucciones
End If
End Sub
    
```

**Importante:**

En esta rutina se utiliza la expresi3n **Elseif** para consultar por un segundo dato y no dejar librado a cualquier valor.

Es decir que si no es V, evaluar si es C. Por cualquier otro valor ingresado el proceso finaliza sin ejecutar instrucci3n alguna.

### 3.4. CANCELAR UN INPUTBOX: CÓMO DETECTARLO

Como utilizamos los cuadros **InputBox** para solicitar al usuario el ingreso de algún dato (fecha, contraseña, criterio para filtro, etc), es **muy importante** evaluar si ha presionado el botón **Cancelar** o el botón de **Cierre**.

Podemos observar **2 tipos de errores** frecuentes en la **programación** de estos cuadros:

1- Si la variable fue definida, por ej como *Integer*, el presionar el botón Cancelar o Cerrar devolverá un valor vacío. Esto se traducirá en un **error** del tipo **13- No coinciden los tipos**.

Para evitar la interrupción del proceso por este error, debemos agregar la línea de control de error. Según el ejemplo anterior las instrucciones quedarán de esta manera:

```
Sub error1_InputBox()  
Dim valor as Integer  
On error goto Salir  
valor = InputBox("Ingrese un valor entre 1 y 4", "Ejemplo")  
If valor > 0 And valor <= 4 Then  
    MsgBox "Correcto"  
    '..... otras instrucciones  
Else  
    MsgBox "Incorrecto"  
    '.... otras instrucciones  
End If  
Salir:  
End Sub
```

2- Volvamos al ejemplo anterior e imaginemos que la variable no ha sido definida o se ha dejado como Variant. Si presionamos el botón **Cancelar** o **Cerrar** sin ingresar valores, la respuesta será siempre el segundo mensaje y el proceso seguirá . Pero **no es lo mismo** ingresar un valor incorrecto que cancelar el proceso . Lo correcto entonces es evaluar primero si se ha presionado el botón **Cancelar**: En ese caso el proceso finaliza sin ejecutar ninguna de las instrucciones de la rutina.

```
Sub error2_InputBox()  
Dim valor  
valor = InputBox("Ingrese un valor entre 1 y 3", "Ejemplo")  
If valor <> "" Then  
    If valor > 0 And valor < 4 Then  
        MsgBox "Correcto"  
    Else  
        MsgBox "Incorrecto"  
    End If  
Else  
    MsgBox "El proceso se canceló"  
End If  
End Sub
```

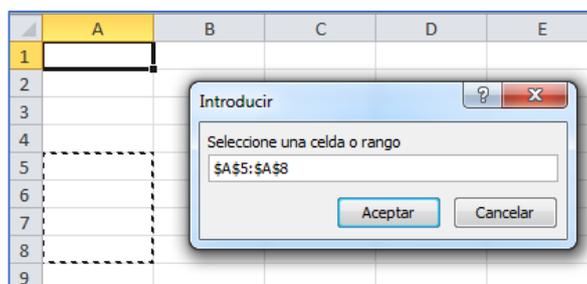
## 3.5. SELECCIONAR RANGO DESDE UN INPUTBOX

Al solicitar el ingreso de un rango desde un **InputBox** se nos presenta el problema de poder equivocarnos en el tipeado de la cadena.

Para evitarlo, podemos permitir que desde un **InputBox** con **Type:=8**, se pueda seleccionar manualmente el rango.

Para ello asignamos a una variable establecida con la instrucción **Set** el rango seleccionado desde la hoja, en lugar de escribirlo en el cuadro.

Al ejecutar la macro se presentará la siguiente ventana. Se selecciona el rango y se presiona el botón **Aceptar**.

**Sub solicitaRango()**

*Dim rango As Range*

*'controlamos posibles errores*

*On Error Resume Next*

**Set rango = Application.InputBox("Seleccione una celda o rango", Type:=8)**

*'si el rango no está vacío lo seleccionamos*

**If Not rango Is Nothing Then**

**rango.Select**

*'guardamos su dirección*

**dire = rango.Address(False, False)**

*MsgBox dire*

*'o guardamos el valor si hemos seleccionado 1 sola celda y mostramos su contenido*

**valor = rango.Value**

*MsgBox valor*

**End If**

*'se devuelve el control de errores a su estado normal*

*On Error GoTo 0*

**End Sub**

**Nota:** la expresión **Type:=8** en la construcción de un **InputBox**, indica que el dato ingresado será de tipo **Rango**

Al escribir: **rango.address(false, false)** nos devuelve una referencia del tipo **B5**, en cambio solo con **rango.address** la referencia será absoluta, del tipo **\$B\$5**

**NOTA:** ver ejemplos en libro **Llamada-Eventos UF** que se adjunta con este manual.

## 4. USO DE CONTROLES

### 4.1. TIPOS DE CONTROLES

### 4.2. COMO COLOCAR CONTROLES EN UNA HOJA EXCEL

### 4.3. BARRA ACTIVEX

### 4.4. BARRA FORMULARIOS

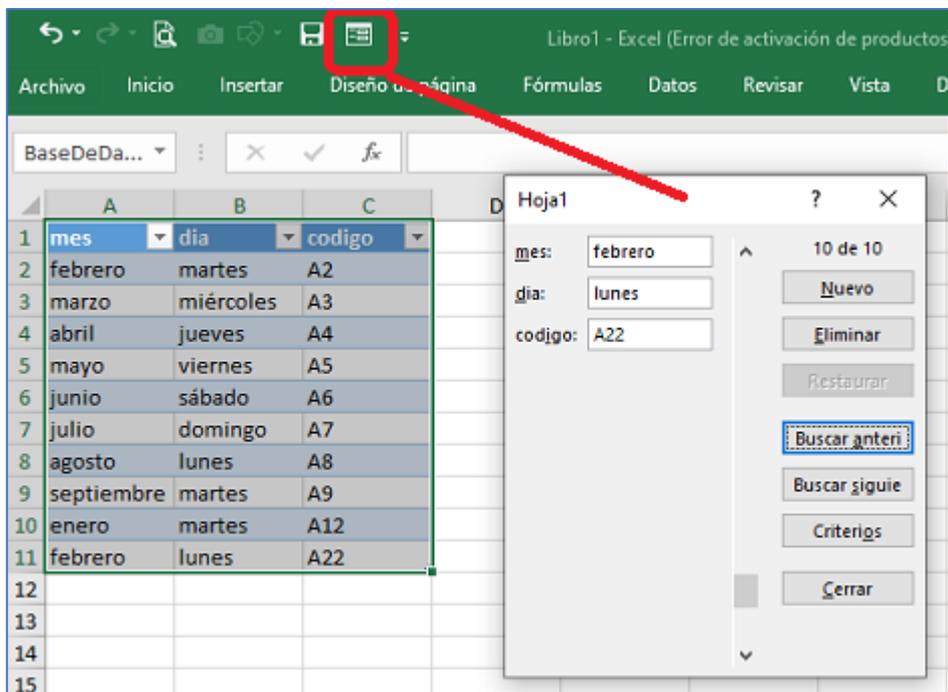
### 4.5. CÓMO RECONOCER SI SE TRATA DE UN CONTROL 'ACTIVEX' O DE 'FORMULARIO'?

### 4.6. CONOCER RUTINA ASOCIADA A UN CONTROL

## 5. TIPOS DE FORMULARIOS

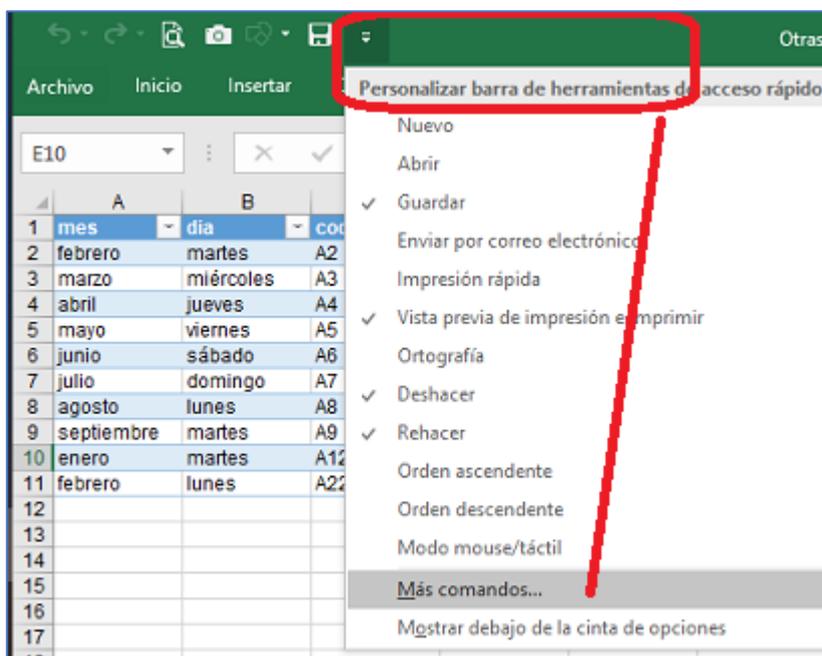
### 5.1. HERRAMIENTA 'FORMULARIO' DE EXCEL

Se trata de un formulario básico, sin necesidad de programación. Que podría ser útil a la hora de crear tablas simples, sin condiciones. Y para realizar búsquedas rápidas.

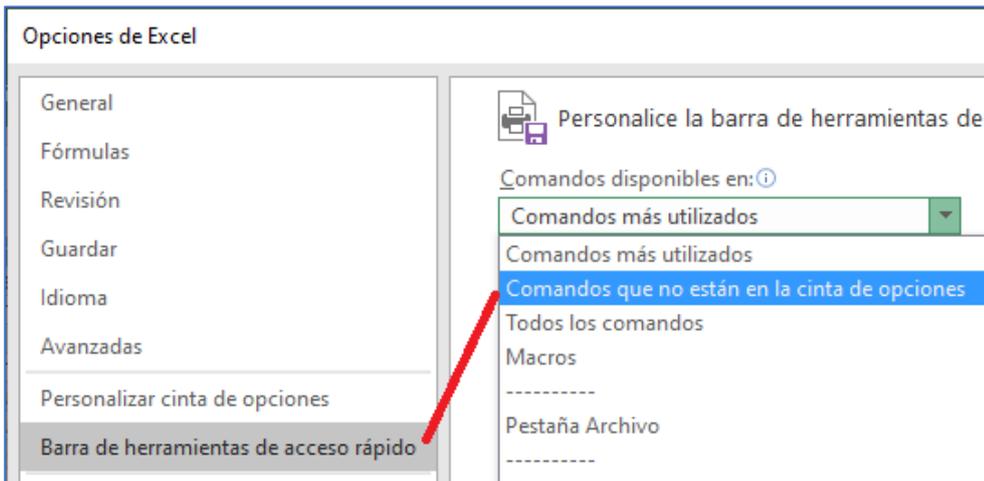


Para utilizar la herramienta **Formulario** habrá que habilitarla previamente.

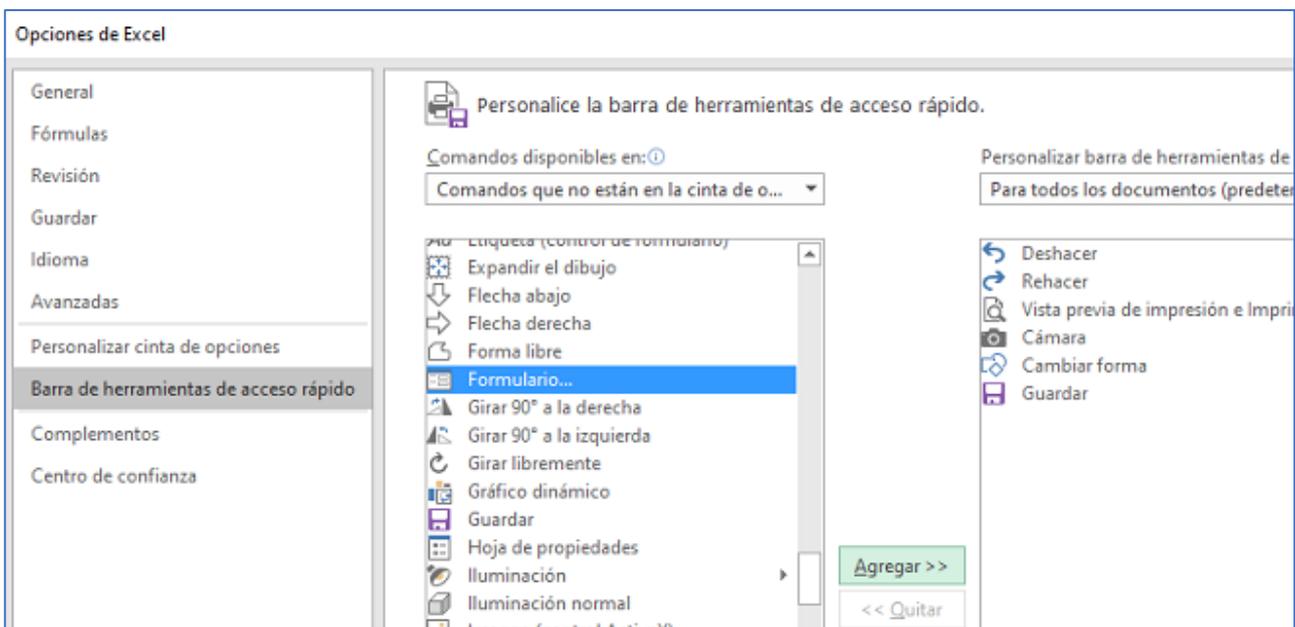
Con clic derecho en botón **Personalizar barra de herramientas de acceso rápido**, opción **Más Comandos**



En la próxima ventana, desde opción **Comandos más utilizados** seleccionar el que diga **Comandos que no se encuentran en la cinta**.



Buscar el que se llama **Formulario**. Presionar **Agregar** y luego **Aceptar**.



Ya tenemos la herramienta disponible en nuestra barra de herramientas.

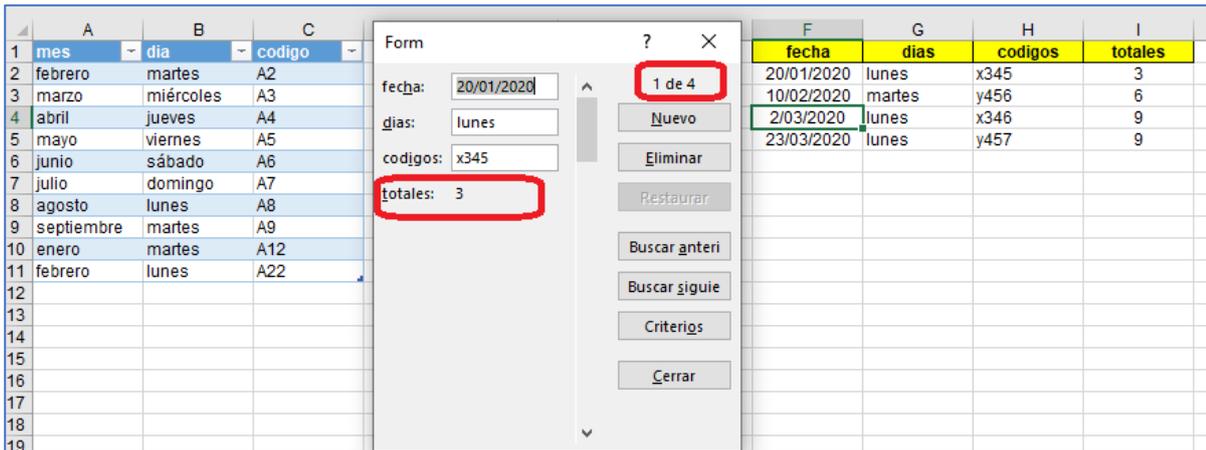
Procederemos a preparar la hoja.

En la imagen se observan 2 tablas de datos: una con formato Tabla y la otra no. Con cualquiera de estos modelos el uso de la herramienta es el mismo.

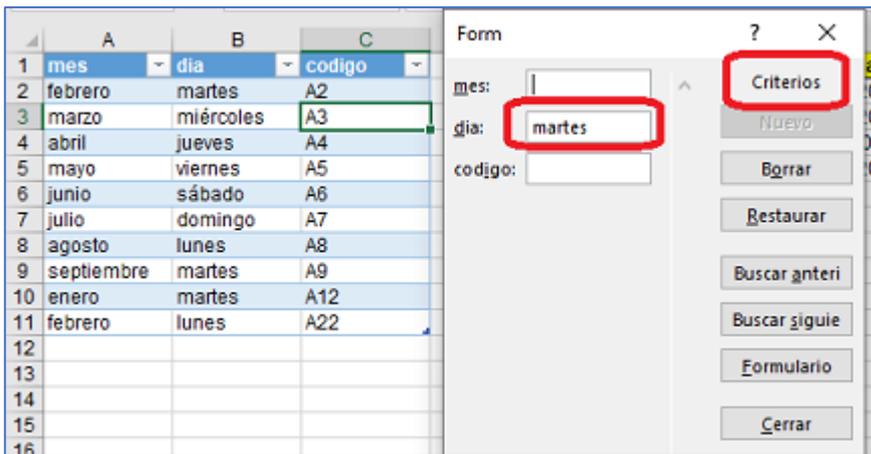
Se colocan los títulos y las fórmulas si las hubiera. Y estando seleccionada alguna celda de la tabla, al llamar a la herramienta Formulario ya se verá el primer elemento de la tabla y el número total de registros.

Los campos con fórmulas solo se observan pero no se permite ingreso en ellos. Tal como se observa en el ejemplo del rango de celdas (no Tabla).

**IMPORTANTE: en este tipo de formularios el avance solo debe hacerse con TAB. Con ENTER equivale a terminar el proceso.**



Para realizar una **búsqueda** presionar el botón **Criterios** e ingresar el dato en su campo correspondiente. Los botones **Buscar anterior** y **Buscar siguiente** permiten navegar por la tabla de datos según esos criterios.



**NOTA:** este tipo de formulario es básico. No permite colocar condiciones, control de contenidos o desplegables para seleccionar datos comunes como nombres de meses o días. Pero puede ser muy útil al momento de realizar una búsqueda rápida.

## 5.2. FORMULARIO DISEÑADO EN HOJA EXCEL.

Este tipo de formulario es ideal para los que no conocen tanto de programación como para diseñar un Userform, donde se requiere conocer los tipos de controles, sus eventos, sus propiedades y suficiente programación como para realizar controles de contenidos.

En este modelo se pueden utilizar algunas de las siguientes herramientas:

- Validación de datos
- Fórmulas
- Desplegables
- Eventualmente algunos controles como botones de Guardar, Imprimir, etc.

**NOTA:** El modelo de formulario de la imagen siguiente se desarrolla completo en capítulo 14.

Copiar Fact

Ver Nota 2 para evitar que el botón salga en la impresión.

Cliente: Tiendas XX

Dirección Av.Bs.As.

Ciudad Bs.As.

Cod Postal 1000

Nro.Cta. 10005

### FACTURA

Fecha: 28/02/2020

Otros datos: -----

Otros datos: -----

Nro Factura 3

**COMENTARIOS**

- Celda Validada
- Celdas con fórmulas

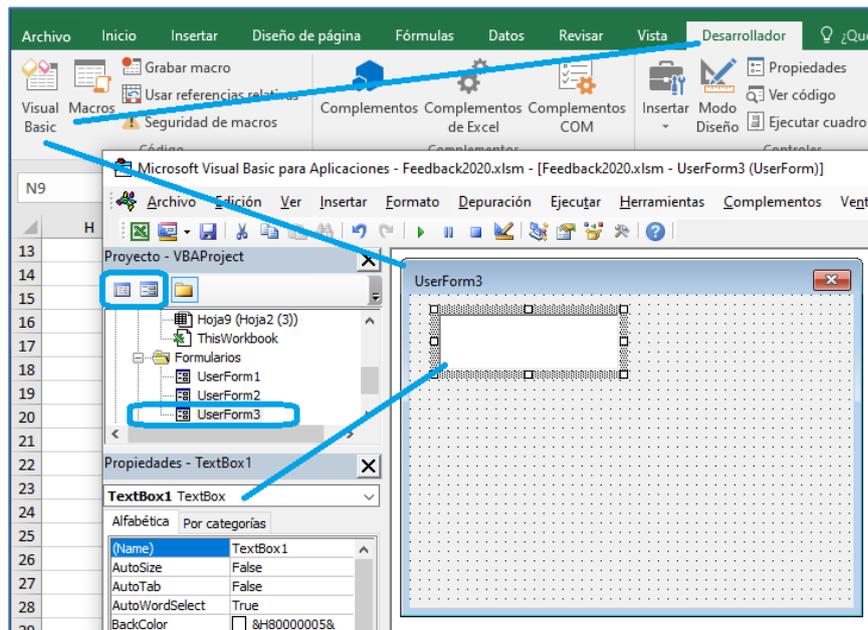
ITEM	CANT	COD.PROD	PRODUCTO	DESCRIPCIÓN	Precio Unit	Precio Tot
1	4	102	pantalón	jean hombres	\$ 60,00	\$ 240,00
2	3	105	pollera corta	pollera tela estampada	\$ 35,00	\$ 105,00
3	1	101	camisa blanca	camisa c/bolsillos		
			#N/A			
			#N/A			
			#N/A			
			#N/A			
			#N/A			
<b>Subtotal</b>					\$	345,00
<b>IVA 11%</b>						37,95
<b>TOTAL:</b>					\$	382,95

Ver Nota1 para evitar error #N/A

### 5.3. USERFORMS.

Los **Userforms** son los formularios que se desarrollan desde el **Editor de Macros**.

Se dibujan los controles necesarios, se establecen sus propiedades y se programa cada uno de los eventos necesarios para su ejecución.



El tema **USERFORMS**, que es la estrella de este manual, se desarrolla a partir del capítulo siguiente.

## 6. LOS USERFORMS

### 6.1. TIPOS DE USERFORMS

### 6.2. CREAR UN USERFORM

### 6.3. PROPIEDADES Y EVENTOS

### 6.4. DIBUJAR CONTROLES EN UN USERFORM

#### 6.4.1. Asignar iguales propiedades a controles

#### 6.4.2. Asignar nombre a los controles

#### 6.4.3. Asignar orden de entrada a los controles.

### 6.5. CÓMO EJECUTAR O LLAMAR A UN USERFORM

### 6.6. CÓMO PROGRAMAR EL FORMULARIO Y SUS CONTROLES

### 6.7. PROPIEDADES DEL USERFORM

#### 6.7.1. Lista detallada de Propiedades

### 6.8. EVENTOS DEL USERFORM

### 6.9. UBICACIÓN DEL USERFORM EN LA PANTALLA.

### 6.10. USERFORM SIEMPRE CENTRADO

### 6.11. INMOVILIZAR UN USERFORM

6.12. OCULTAR UN USERFORM

6.13. CERRAR UN USERFORM

6.14. ACCEDER AL LIBRO DESDE UN USERFORM ABIERTO

6.15. INHABILITAR BOTÓN DE CERRAR

6.16. QUITAR BOTÓN DE CERRAR (X)

6.17. AGREGAR BOTONES DE MINIMIZAR Y MAXIMIZAR EN UN USERFORM

## 7. EL CONTROL LABEL O ETIQUETA

7.1. CARACTERÍSTICAS DE UN CONTROL LABEL

7.2. PROPIEDADES Y EVENTOS DE UN LABEL

7.3. INICIALIZANDO UN CONTROL LABEL

7.4. USO DE LABEL COMO CONTADOR

7.5. EVENTO MOUSEMOVE: MOSTRAR MENSAJE AL PASAR EL MOUSE POR EL CONTROL

7.6. LIMPIEZA DE LABEL

7.7. COLOR PARPADEANTE EN UN LABEL

## 8. EL CONTROL TEXTBOX O CUADROS DE TEXTO

### 8.1. CARACTERISTICAS DE UN CONTROL TEXTBOX

### 8.2. PROPIEDADES Y EVENTOS DE UN TEXTBOX

### 8.3. INICIALIZANDO UN CONTROL TEXTBOX

### 8.4. TEXTBOX CON DATOS DE OTRO USERFORM

### 8.5. ASIGNANDO ORDEN O FOCO A LOS CONTROLES

### 8.6. TEXTBOX PARA CONTRASEÑAS

### 8.7. CÓMO Y CUÁNDO CONTROLAR DATOS DE INGRESO

#### 8.7.1. Longitud máxima para un campo

### 8.8. EVENTO KEYPRESS PARA CONTROLAR CARACTERES INGRESADOS.

#### 8.8.1. Caracteres numéricos.

#### 8.8.2. Caracteres especiales para números.

#### 8.8.3. Caracteres alfabéticos.

### 8.9. FUNCIONES DE VALIDACIÓN

#### 8.9.1. Controlar campos numéricos

#### 8.9.2. Controlar caracteres o texto

8.10. VALIDAR FORMATOS FECHA

8.11. INGRESAR FECHAS SIN SEPARADORES

8.12. FUNCIONES DE CONVERSIÓN. PASAR DATOS A HOJA

8.12.1. Convertir texto en mayúsculas (o minúsculas)

8.12.2. Convertir dato numérico a formato moneda.

8.12.3. Evitar campos vacíos al momento de guardar

8.13. SELECCIONAR RANGOS CON REFEDIT

8.14. LIMPIEZA DE TEXTBOX

8.14.1. ¿Cómo limpiar todos los controles sin repetir las instrucciones?

8.15. SELECCIÓN DEL TEXTO COMPLETO EN UN TEXTBOX

8.16. SIMULAR MARCAS DE AGUA EN UN TEXTBOX

## 9. LOS OBJETOS COMBOBOX Y LISTBOX

### 9.1. CARACTERISTICAS DE LOS COMBOBOX/LISTBOX

### 9.2. PROPIEDADES Y EVENTOS DE LOS COMBOBOX Y LISTBOX

### 9.3. EJEMPLOS DE PRINCIPALES PROPIEDADES DE COMBOBOX/LISTBOX

9.3.1. Asignar rango a un Combo o Lista

9.3.2. Desplegar la lista de un Combobox

9.3.3. Celda asociada al elemento seleccionado

9.3.4. Autocompletar datos en un control

9.3.5. Permitir el ingreso de datos que no están en la lista

9.3.6. Seleccionar varios registros de una lista (solo para LISTBOX)

9.3.7. Mostrar lista con botones de opción (solo para LISTBOX)

9.3.8. Mover elementos desde un Listbox a un Textbox

### 9.4. ACTUALIZAR RANGO ASOCIADO A UN COMBOBOX/LISTBOX

9.4.1. Rango de datos fijo.

9.4.2. Rango dinámico (4 casos)

### 9.5. LLENAR UN COMBOBOX SIN DATOS DUPLICADOS.

### 9.6. CÓMO LLENAR COMBOBOX/LISTBOX DE VARIAS COL

### 9.7. CONTROLES DEPENDIENTES (COMBOS Y/O LISTAS)

9.8. TRABAJAR CON LISTAS O CUADROS DE VARIAS COLUMNAS

9.8.1. Cantidad de columnas

9.8.2. ¿Qué valor guardar o mostrar en controles de varias columnas?

9.8.3. Ancho de columnas

9.8.4. Títulos de columnas

9.8.5. ListBox con más de 10 columnas.

9.9. TRABAJAR CON 2 (O MÁS) LISTAS PARALELAS

9.10. MOSTRAR ELEMENTO SELECCIONADO DE LISTA

9.11. SELECCIONAR VARIOS REGISTROS DE UN LISTBOX

9.12. QUITAR ELEMENTOS DE UNA LISTA

9.13. REORDENAR ELEMENTOS DE UNA LISTA

9.14. CONTAR ELEMENTOS DE UNA LISTA

9.15. PASAR DATOS DE UN COMBOBOX O LISTA A LA HOJA

9.16. MOSTRAR DATOS FILTRADOS EN UN CONTROL COMBOBOX O LISTBOX

9.17. LIMPIEZA DE COMBOBOX/LISTBOX

9.18. TRAZAR LÍNEA PARA SEPARAR ELEMENTOS DE UN LISTBOX .

## 10. EL OBJETO FRAME O MARCO

### 10.1. CARACTERISTICAS DE UN CONTROL FRAME

Un control **FRAME o Marco** se utiliza para **agrupar controles**.

Utilizaremos un control **Frame** para crear un grupo de opciones exclusivas (optionButton), controles no excluyentes (checkbox) o controles con contenido estrechamente relacionado, como pueden ser los tipos de datos para crear registros en una base de Clientes, Proveedores o de otra índole.

En la siguiente imagen vemos 3 controles Frame utilizados para diferenciar las distintas tareas:

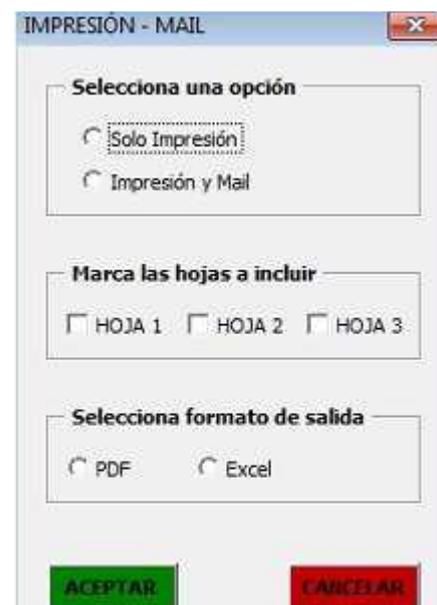
- a- Para agrupar un par de botones de opción.
- b- Para contener los controles relacionados con la elección del primer OptionButton.
- c- Para contener los controles relacionados con la elección del segundo botón.



La imagen a derecha nos muestra un modelo de formulario que será llamado antes de la impresión. Desde allí se establecen los diferentes criterios y según eso serán las macros a ejecutar.

Utilizamos la propiedad **Caption** del Frame si deseamos dejar un texto aclaratorio del contenido de los marcos.

O podemos dejarlos sin texto como en la siguiente imagen:



**NOTA: El código se desarrolla en el capítulo 12 (CheckBox) en el tema 12.7 : combinar option y checkbox.**

## 10.2. PROPIEDADES Y EVENTOS DE UN CONTROL FRAME

Generalmente utilizamos un control **Frame** solo a efectos de agrupar o separar controles dentro de un formulario. Por lo que aquí veremos solo algunas propiedades. La lista completa se encuentra detallada en el último capítulo de este manual.

En el ejemplo de la imagen anterior se utilizó la propiedad **CAPTION** para indicar el contenido de los controles que contienen los 2 marcos o **Frame** mayores, pero puede dejarse esa propiedad en blanco y solo utilizar el marco para separar o agrupar los elementos como en el caso de los botones de opción.

Hemos visto que la propiedad **TABINDEX** indica el orden de **foco** de cada control. En un Userform con **Frame**, el que recibe el orden es el marco, siguiendo con el resto de los controles del UF.

Dentro del **Frame**, el orden de los controles que allí se encuentran empieza nuevamente en 0.

El **evento** predeterminado de un control **Frame** es **Click**.

Evento	Descripción
AddControl	Se produce cuando se inserta un control en el formulario, en tiempo de ejecución
BeforeDragOver	Se produce cuando se está ejecutando una operación de arrastrar y colocar.
BeforeDropOrPaste	Se produce cuando el usuario está a punto de colocar o pegar datos en un objeto.
Click	Se produce al hacer clic en algún sector del Frame, no al clic en algún control dentro de él.
DbIclic	Se produce al hacer doble clic en algún sector del Userform, no al doble clic en algún control
Enter	El evento Enter se produce antes de que un control reciba realmente el enfoque desde un control del mismo formulario.
Error	Se produce cuando un control detecta un error y no puede devolver información del error al programa que lo ha llamado.
Exit	El evento Exit se produce inmediatamente antes que un control pierda el enfoque en favor de otro control del mismo formulario
KeyDown	Se producen en secuencia cuando un usuario presiona y suelta una tecla. El evento KeyDown se produce cuando el usuario presiona una tecla.
KeyPress	Se produce cuando el usuario presiona una tecla ANSI.
KeyUp	El evento <b>KeyUp</b> se produce cuando el usuario suelta una tecla.
Layout	Se produce cuando cambia el tamaño de un formulario
MouseDown	Se produce cuando el usuario hace clic con un botón del mouse. MouseDown se produce cuando el usuario presiona el botón del mouse;
MouseMove	Se produce cuando el usuario mueve el mouse.
MouseUp	<b>MouseUp</b> se produce cuando el usuario suelta el botón del <i>mouse</i> .
RemoveControl	Este evento se produce cuando se elimina un control desde el formulario, no cuando un control se descarga debido a que el formulario se está cerrando
Scroll	Se produce cuando se vuelve a posicionar un cuadro de desplazamiento.
Zoom	Se produce cuando cambia el valor de la propiedad Zoom.

**Nota: el siguiente ejemplo y otros más con uso de Frames se encuentran desarrollados en libro Frame-Option-Check que se acompaña con este manual.**

### 10.3. AGRANDAR UN FRAME A MEDIDA QUE SE AGREGAN DATOS

A veces por razones de espacio, podemos tener un **Frame** con varios **Textbox** los cuales *solo* se verán si fuese necesario llenar datos en ellos.

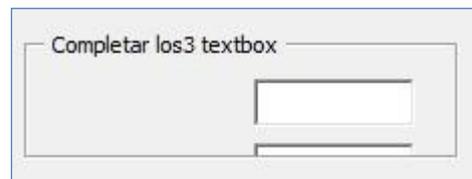
Es decir que necesitamos **agrandar** el frame a medida que vamos utilizando los textbox.

Una manera sería agregar código al evento **Change del ScrollBar** que tenga ese Frame, y otra la que veremos a continuación, que le va dando mayor altura al marco a medida que pasamos hacia los textbox de la línea siguiente:

```
Private Sub TextBox1_Enter()  
Frame1.Height = Frame1.Height + 20  
' el tamaño que necesites para que se vea el otro box  
End Sub
```

```
Private Sub TextBox2_Enter()  
Frame1.Height = Frame1.Height + 20  
' el tamaño que necesites para que se vea el otro box  
End Sub
```

```
Private Sub TextBox3_Enter()  
Frame1.Height = Frame1.Height + 20 ' el tamaño necesario para ver el otro box  
End Sub
```



Y además agregaremos al último Textbox este código que vuelve al Frame a su tamaño reducido:

```
Private Sub TextBox3_AfterUpdate()  
Frame1.Height = Frame1.Height - 60  
'porque incrementé 20 a cada uno de los 3 textbox  
End Sub
```

**NOTA:** Podríamos tener, además del código en el evento *AfterUpdate* del último control, un pequeño botón en el mismo Frame con estas instrucciones:

```
Private Sub CommandButton3_Click()  
If Frame3.Height > 54 Then Frame3.Height = 54  
End Sub
```

Como las instrucciones para reducir la altura del frame se coloca en el evento del último control, es posible que querramos llenar solo algunos más. Entonces para ocultarlos nuevamente utilizaremos este botón. La altura del frame en este modelo es de 54 y así s lo vuelve.

### 10.4. INHABILITAR TODOS LOS CONTROLES DENTRO DE UN FRAME.

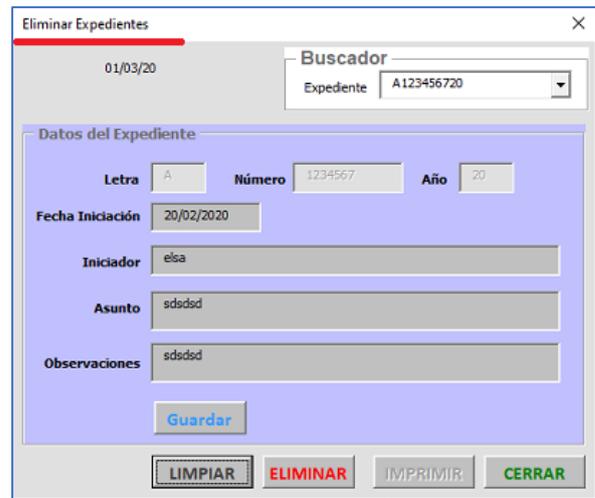
Este es un requisito frecuente: según cierta condición se requiere habilitar o no una serie de controles, que podremos tenerlos agrupados dentro de un marco o Frame.

En la siguiente imagen, se trata de un formulario utilizado tanto para crear, modificar, buscar y eliminar registros de una base. Por lo tanto, cuando se trata de eliminar solo se muestran los campos pero ese marco está inhabilitado de realizar cambios incluyendo el botón de Guardado. Para esto se establece la propiedad Enable del Frame en False.

**Ejemplo 1:**

```

If opc = 1 then
    Frame2.Enabled =
False
Else
    Frame2.Enabled = True
End If
    
```



Al activar el formulario se sabe quién lo llamó analizando una variable pública. Y según eso será cómo se muestran o habilitan los diferentes controles. Las instrucciones para inhabilitar controles pueden estar en diferentes eventos. En este caso se optó por el de Activate del Userform.

**Ejemplo 2:**

```

Private Sub UserForm_Activate()
Select Case llama 'variable pública
    Case Is = 1
        Frame2.Enabled = False 'se inhabilita el frame del buscador
        Me.Caption = "Registrar Expedientes"
    Case Is = 2
        Me.Caption = "Modificar Expedientes"
        CmdEliminar.Enabled = False 'se inhabilita el botón de Eliminar
    Case Is = 3
        Frame4.Enabled = False 'se inhabilita el frame con los datos y botón Guardar
        Me.Caption = "Eliminar Expedientes"
End Select
End Sub
    
```

## 11. EL OBJETO OPTIONBUTTON O BOTONES DE OPCIÓN

### 11.1. CARACTERÍSTICAS DE LOS OPTIONBUTTON

### 11.2. PROPIEDADES Y EVENTOS DE LOS OPTIONBUTTON

### 11.3. RECONOCER EL OPTIONBUTTON SELECCIONADO

### 11.4. EJECUTAR RUTINA AL SELECCIONAR UN BOTÓN DE OPCIÓN

### 11.5. ATAJO DE TECLADO PARA SELECCIONAR UN BOTÓN DE OPCIÓN

### 11.6. LIMPIEZA DE OPTIONBUTTONS

## 12. EL OBJETO CHECKBOX O CASILLAS DE VERIFICACIÓN

### 12.1. CARACTERÍSTICAS DE LOS CHECKBOX

### 12.2. PROPIEDADES Y EVENTOS DE UN CHECKBOX

### 12.3. RECONOCER LOS CHECKBOX SELECCIONADOS

### 12.4. CÓDIGO PARA SELECCIÓN DE CHECKBOX

### 12.5. ATAJO DE TECLADO PARA TILDAR UN CHECKBOX (ACCELERATOR)

### 12.6. LIMPIEZA DE CHECKBOX

### 12.7. COMBINAR OPTION Y CHECKBOX

## 13. EL OBJETO COMMANDBUTTON O BOTÓN DE COMANDO

### 13.1. CARACTERÍSTICAS DE UN BOTÓN DE COMANDO

### 13.2. PROPIEDADES Y EVENTOS DE LOS COMMANDBUTTON

### 13.3. EJEMPLOS DEL USO DE UN BOTÓN DE COMANDO

### 13.4. BOTONES CON IMÁGENES

### 13.5. ATAJOS DE TECLADO PARA EJECUTAR BOTONES.

### 13.6. INHABILITAR BOTONES SEGÚN CONDICIÓN

## 14. FORMULARIOS EN HOJA EXCEL

14.1. CARACTERÍSTICAS DE ESTOS FORMULARIOS

14.2. USAR SOLO CELDAS HABILITADAS

14.3. CÓMO VOLCAR DATOS A UNA HOJA RESUMEN

14.4. CÓMO LIMPIAR UN FORMULARIO EN HOJA

14.5. USO DE FÓRMULAS

14.6. USO DE CELDAS CON VALIDACIÓN

14.7. PROGRAMANDO CONTROLES

## 15. UF: APERTURA CON CONTRASEÑA

## 15.1. EJEMPLO DE USERFORM DE INGRESO

En este **Modelo** confeccionaremos un Userform para solicitar una **clave de ingreso** según la tarea a realizar. **(ver en libro Ejemplos\_Userforms el modelo nº 1)**

## 15.2. ASIGNANDO PROPIEDADES

Una vez dibujados los controles en el formulario, será momento de asignarle algunas **propiedades**. **Nombre** del control, **texto** a mostrar, **tamaño**, **posición**, **color**, **orden**, **visible** o no, **habilitado** o no, son algunas de las **propiedades** que estableceremos desde esta ventana:

Los **controles** utilizados y **sus propiedades** asignadas para este modelo, se indican en la siguiente tabla.

Control	Propiedad	Valor predeterminado	Nombre del Control
Userform1	Caption	FORMULARIO DE INGRESO	Userform1
Label1	Caption	Bienvenidos al Sistema de Facturación	Label1
	ForeColor	&H00C00000& (azul)	
Frame1	Caption	Elija su tarea:	Frame1
	TabIndex	0	
OptionButton1	Name	OB_Clie	OB_Clie
	Acelerator	C	
	Caption	Clientes	
	Height	25	
	Top	15	
OptionButton2	Name	OB_Prod	OB_Prod
	Acelerator	P	
	Caption	Productos	
	Height	25	
	Top	15	

OptionButton3	Name	OB_Fact	OB_Fact
	Accelerator	F	
	Caption	Facturas	
	Height	25	
	Top	15	
Label2	Caption	Ingrese su contraseña:	Label2
TextBox1	Name	txtPass	txtPass
	PassWordChar	*	
	TabIndex	1	
CommandButton1	Name	cmdAcepta	cmdAcepta
	Caption	ACEPTAR	
	Font	Tahoma, Negrita, 10	
	TabIndex	2	

Luego, durante la ejecución del formulario, podremos modificar algunas de estas propiedades, como veremos en los ejemplos que se presentan en cada uno de los capítulos siguientes.

#### Texto que mostrará el control:

La propiedad **Caption** es el texto que muestran algunos controles como **Label, Frame, OptionButton, CommandButton** y también el **Userform**. No son valores que el usuario modifique en el uso del formulario.

En cambio en los **Textbox** se utiliza la propiedad **Value o Text**.

#### Nombre del control:

El **nombre** del control (**Name**) es una propiedad importante a la hora de programarlos. Será más fácil de ubicar de qué control se trata si los llamamos haciendo referencia a su contenido. OB\_Clie, OB\_Prod son nombres más apropiados que dejar los OptionButton predeterminados.

#### Orden de selección o foco:

La propiedad **TabIndex** nos permite decidir el **orden** en que los controles recibirán el foco al presionar Tab o Enter.

Los controles **Label** no reciben el foco salvo que modifiquemos su propiedad **TabStop** colocándola en True. Los controles **OptionButton** tienen su propio orden dentro del **frame** que los contiene.

#### Acceso rápido al control:

En este ejemplo también se aplicó la propiedad **Accelerator**, la cual sirve para configurar el acceso rápido a los controles de opción, acción que se realiza mediante la combinación teclas "ALT + Letra ingresada.

#### Propiedad 'password':

Al control **Textbox** se le asignó '\*' (asterisco) en la propiedad **PassWordChar** para impedir se vea la clave ingresada.

#### Propiedades de tamaño:

**Height** = Alto del control

**Width** = ancho del control

#### Propiedades de posición:

**Left** = posición margen izquierdo

**Top** = posición margen superior

En el ejemplo que se desarrolla en este capítulo, las propiedades de **tamaño y ubicación** se asignaron automáticamente a medida que dibujamos y ubicamos el control manualmente. Luego se ajustaron algunas de ellas desde la ventana de propiedades.

Por ejemplo, a los 3 controles `OptionButton` se les asignó la misma propiedad **Top** para que aparezcan alineados.

Para **asignar las mismas propiedades** a varios controles, como los `OptionButton` del ejemplo, seleccionar los 3 controles manteniendo presionada la tecla **Ctrl**. Luego ingresar valores en **Height, Top, Font**, y el resto de las propiedades comunes.

### 15.3. PROGRAMANDO UN USERFORM DE INGRESO

*Qué tenemos que tener en cuenta en este ejercicio? Qué se espera del usuario al ejecutarse el form de inicio?*

- 1-Que se haya seleccionado alguna opción del Frame
- 2-Que la clave ingresada sea correcta, es decir que coincida tarea+clave
- 3-Acciones a seguir en caso de haber pasado los 2 controles anteriores. En este caso se habilitarán las hojas según la tarea seleccionada en los botones de opción.

Con *doble click* entonces sobre el control `CommandButton` pasaremos a la hoja de código donde encontramos estas líneas:

```
Private Sub cmdAcepta_Click()
```

```
End Sub
```

La rutina completa quedará así:

```
Private Sub cmdAcepta_Click()  
'controla si se hizo selección de tarea  
If OBCLie.Value = False And OBProd.Value = False And OBFact.Value = False Then  
    MsgBox "Debe seleccionar alguna tarea"  
    Exit Sub  
End If  
'controla si tarea y usuario corresponden  
If txtPass = "usuario1" And OBCLie.Value = True Then  
    'habilita la hoja  
    Sheets("Clientes").Visible = True  
    Sheets("Clientes").Select  
Elseif txtPass = "usuario2" And OBProd.Value = True Then  
    Sheets("Productos").Visible = True  
    Sheets("Productos").Select  
Elseif txtPass = "usuario3" And OBFact.Value = True Then  
    Sheets("Facturas").Visible = True  
    Sheets("Facturas").Select  
Else  
    'no encontró correspondencia entre tarea y clave  
    MsgBox "Contraseña inválida para la tarea seleccionada"  
    txtPass.Value = ""  
    Exit Sub
```

```
End If
'en caso correcto se cierra el formulario
Unload Me
End Sub
```

**Nota:** Este modelo de formulario podemos utilizarlo como pantalla inicial donde cada usuario luego de ingresar su tarea y su contraseña acceda a las hojas autorizadas.

#### 15.4. LLAMANDO AL USERFORM DE INGRESO

Ya **diseñamos** nuestro primer Userform: dibujamos controles, le asignamos algunas propiedades y **programamos** el evento para un botón de comando.

Ahora es tiempo de poder **ejecutarlo** y esto es lo que veremos:

##### ¿Cómo ejecutar o llamar un Userform?

**1- Método 1:** Como se trata de un modelo que será el inicio de una aplicación tendrá que ir en el **evento Open** del libro.

En el Editor de macros, objeto ThisWorkbook (o Este Libro) ingresar estas instrucciones:

```
Private Sub Workbook_Open()
UF_Ingreso.Show
End Sub
```

**2- Método 2:** En el libro de **Ejemplos**, se colocó un **botón de llamada** en la Hoja1. Para esto se dibuja un botón de comando con la barra '**Cuadro de controles**', clic secundario sobre el control, opción '**Ver código**' y al entrar al Editor de macros, se ingresó este código:

```
Private Sub CommandButton1_Click()
UF_Ingreso.Show
End Sub
```

**ACLARACIÓN:** Este **2do.método** para llamar a un Userform **es el que se utiliza de aquí en más** en cada uno de los ejemplos presentados en los **libros de Ejemplos** que se adjuntan con el manual.

## 16. UF: APERTURA CON VALORES PREDETERMINADOS

### 16.1. EJEMPLO DE USERFORM CON VALORES INICIALES

### 16.2. ASIGNAR VALORES DE INICIO A CONTROLES

### 16.3. LLAMANDO AL USERFORM CON VALORES INICIALES

## 17. UF: INGRESO DE DATOS CON VALIDACIÓN

### 17.1. DISEÑO FORMULARIO CON VALIDACIÓN

### 17.2. VUELCO DE DATOS VALIDADOS O CONVERTIDOS, A HOJA

### 17.3. VOLCAR DATOS COMO HIPERVÍNCULOS. BUSCAR VINCULACIÓN.

**18. UF: ALTAS, BAJAS Y CAMBIOS A BASE DE DATOS**

18.1. FORMULARIO PARA ALTAS, BAJAS Y MODIFICACIONES

18.2. PERMITIR INGRESO DE DATOS EN COMBOBOX

18.3. BOTONES PARA ALTAS, BAJAS Y CAMBIOS

18.4. BOTONES PARA NAVEGAR POR LA BASE

## 19. UF: CON CONTROL MULTIPAGINA o 'MULTIPAGE'

### 19.1. CONSIDERACIONES EN FORMULARIOS MULTIPAGE

### 19.2. PROPIEDADES DE UN MULTIPAGE Y SUS *PAGES*

### 19.3. EVENTOS DE UN MULTIPAGE

### 19.4. DISEÑO DE UN FORMULARIO MULTIPAGE

### 19.5. AÑADIR, QUITAR O MOVER PESTAÑAS

### 19.6. PROGRAMANDO BOTONES COMUNES

### 19.7. PROGRAMANDO BOTONES DE AVANCE

### 19.8. MULTIPAGE DE COLORES

## 20. UF: FORMULARIO DE AYUDA

### 20.1. DISEÑO DE UN FORMULARIO DE AYUDA

### 20.2. CONFIGURANDO EL FORMULARIO DE AYUDA

### 20.3. PROGRAMANDO UN FORMULARIO DE AYUDA

## 21. CONTROLES ESPECIALES

### 21.1. CONTROL SPINBUTTON O BOTÓN DE NRO

21.1.1. Propiedades y Eventos de un SpinButton

21.1.2. Ejemplo de código para un SpinButton

### 21.2. CONTROL SCROLLBAR O BARRA DE DESPLAZAMIENTO

21.2.1. Propiedades y Eventos de un ScrollBar

21.2.2. Ejemplo de código para un ScrollBar

### 21.3. CONTROL IMAGE O DE IMAGEN

21.3.1. Propiedades y Eventos de un control Image

21.3.2. Ejemplo de código para un control Image

21.3.3. Control con Imagen animada

21.3.4. Gráficos- Tabla Dinámica –Tabla de datos como imagen

### 21.4. CONTROL REFEDIT

21.4.1. Propiedades y Eventos de un control RefEdit

21.4.2. Ejemplo de código para un control RefEdit

## 21.5. CONTROL TOGGLEBUTTON O BOTÓN DE ALTERNAR

### 21.5.1. Propiedades y Eventos de un control ToggleButton

### 21.5.2. Ejemplo de código para un control ToggleButton

## 21.6. CONTROL TABSTRIP O BARRA DE TABULACIONES

### 21.6.1. Propiedades y Eventos de un TablStrip

### 21.6.2. Ejemplo de código para un control TabStrip

## 21.7. LOS CONTROLES CALENDAR Y DATAPICKER

### 21.7.1. Propiedades y Eventos de controles DTPicker y Calendar

### 21.7.2. Ejemplos de código para controles Calendar y DTPicker:

## 22. PROPIEDADES DE LOS CONTROLES

### 22.1. PROPIEDADES DE USERFORM Y CONTROLES

-----FIN DEL MANUAL-----