

Programas de Aplicación III

Tema 4. Otros controles

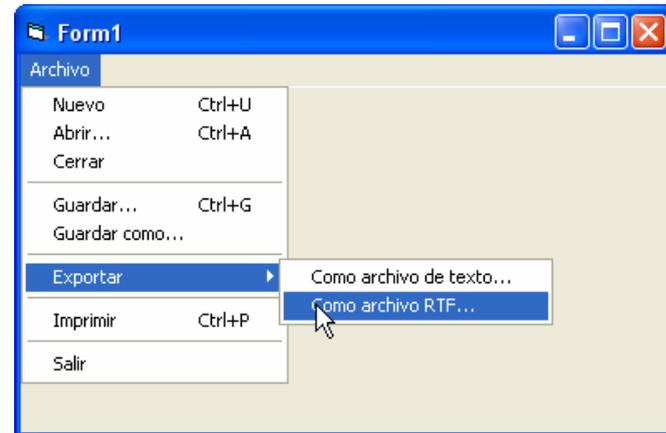
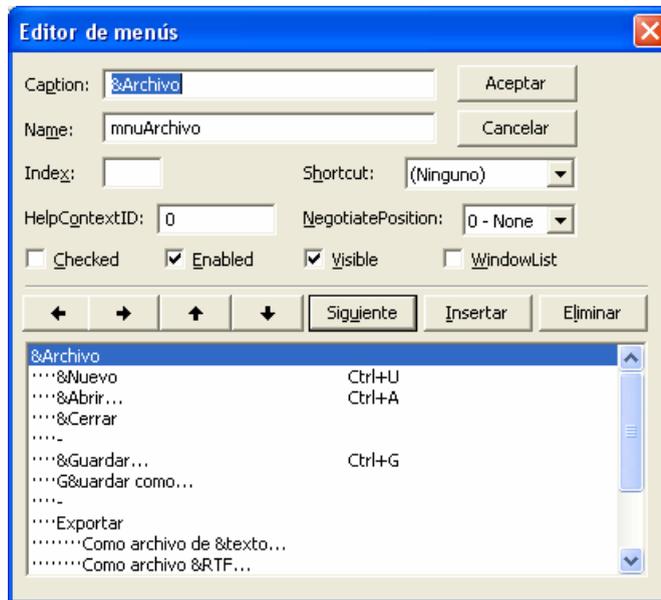
Luis Rodríguez Baena y María Dorrego Luxán

Universidad Pontificia de Salamanca (campus Madrid)

Facultad de Informática

Menús (I)

- ❑ Control intrínseco que no aparece en la barra de herramientas.
 - Para utilizar el editor de menús: Ctrl+E o .
- ❑ El editor de menús.



Menús (II)

❑ Propiedades

Name	Nombre del control. Es necesario introducirlo, no hay un nombre por omisión.
Caption	Título del menú.
Checked	Valor lógico. Indica si la opción de menú muestra o no la marca de verificación.
Enabled	Valor lógico. Indica si la opción de menú está o no habilitada.
NegotiatePosition	Determina la forma en que los menús de nivel superior aparecen cuando en el formulario existe un objeto incrustado (OLE). <ul style="list-style-type: none">● 0. El menú no se muestra cuando hay un objeto OLE activo.● 1 (izquierda). El menú se muestra a la izquierda del menú del objeto OLE.● 2 (medio). El menú se muestra en el centro del menú del objeto OLE.● 3 (derecha) El menú se muestra a la derecha del menú del objeto OLE.
Shortcut	Establece la tecla de método abreviado a un objeto menú. Sólo en tiempo de diseño.
Visible	Valor lógico. Indica si la opción de menú es o no visible.
WindowList	En aplicaciones MDI saca los títulos de los formularios abiertos.

Menús (III)

❑ Notas sobre la propiedad Name.

- El prefijo recomendado es mnu.
- El nombre de un objeto menú debe estar relacionado con su jerarquía.

❑ Notas sobre la propiedad Caption.

- Las líneas horizontales aparecen si en la propiedad Caption se introduce un menú.
- La tecla de acceso directo se determina anteponiendo el carácter &.
- Cuando alguna opción abre un cuadro de diálogo se debe seguir de puntos suspensivos.

❑ Eventos: Click.

Menús (IV)

□ Jerarquías de menús.

- Se pueden anidar hasta cinco niveles de submenú.
- La estructura de los menús se consigue en el editor de menús.
- En el código fuente del formulario aparece la jerarquía diseñada.

```
Begin VB.Menu mnuArchivo
    Caption          = "&Archivo"
    Begin VB.Menu mnuArchivoNuevo
        Caption      = "&Nuevo"
        Shortcut     = ^U
    End
    ...
    Begin VB.Menu mnuArchivoExportar
        Caption      = "Exportar"
        Begin VB.Menu mnuArchivoExportarTexto
            Caption   = "Como archivo de &texto..."
        End
    End
    ...
```

Menús emergentes (I)

❑ Método PopupMenu del objeto Form.

[objeto.]PopupMenu *nombreMenu*, [*flags*], [*x*], [*y*], [*predeterminado*]

- nombreMenu. Puede ser cualquier objeto menú definido.
 - ✓ Se pueden hacer menús emergentes que no aparezcan en el menú principal haciendo su propiedad visible a false.
- Flags. Establece las características del menú.
 - ✓ 0 - vbPopupMenuLeftAlign. El lado izquierdo del menú emergente se sitúa en *x*.
 - ✓ 4 – vbPopupMenuCenterAlign. El menú emergente se centra en *x*.
 - ✓ 8 - vbPopupMenuRightAlign. El lado derecho del menú emergente se sitúa en *x*.
 - ✓ 0 – vbPopupMenuLeftButton. Sólo reacciona a los clics cuando se usa el botón derecho.
 - ✓ 2 – vbPopupMenuRightButton. Reacciona a los dos botones.
- X, Y coordenadas donde aparecerá el botón.
- Predeterminado. Opción predeterminada del menú emergente.

Menús emergentes (II)

□ Utilización.

- Se utiliza en combinación con el evento `MouseDown` de un control.

```
Private Sub objeto_MouseDown(botón As Integer, mayús As Integer, x As Single, y As Single)
```

✓ botón.

- × 1 - `vbLeftButton`. Está presionado el botón primario.
- × 2 - `vbRightButton`. Está presionado el botón secundario.
- × 3 - `vbMiddleButton`. Está presionado el botón central.

✓ mayús.

- × 1 - `vbShiftMask`. Está presionada la tecla MAYÚS.
- × 2 - `vbCtrlMask`. Está presionada la tecla CTRL.
- × 3 - `vbAltMask`. Está presionada la tecla ALT.

- ✓ X, Y. Coordenadas del punto donde se ha pulsado el botón respecto al control.

Menús emergentes (II)

□ Ejemplos.

```
Private Sub Form_MouseDown(Button As Integer, Shift As Integer, X As Single,
                            Y As Single)
    If Button = vbRightButton Then
        PopupMenu mnuArchivo, , , , mnuArchivoGuardar
    End If
End Sub
```

```
'Evita que aparezca el menú contextual por omisión del control Text1
Private Sub Text1_MouseDown(Button As Integer, Shift As Integer, X As Single,
                            Y As Single)
    If Button = vbRightButton Then
        Text1.Enabled = False
        PopupMenu mnuArchivo, , , , mnuArchivoGuardar
        Text1.Enabled = True
    End If
End Sub
```

Arrays de menús

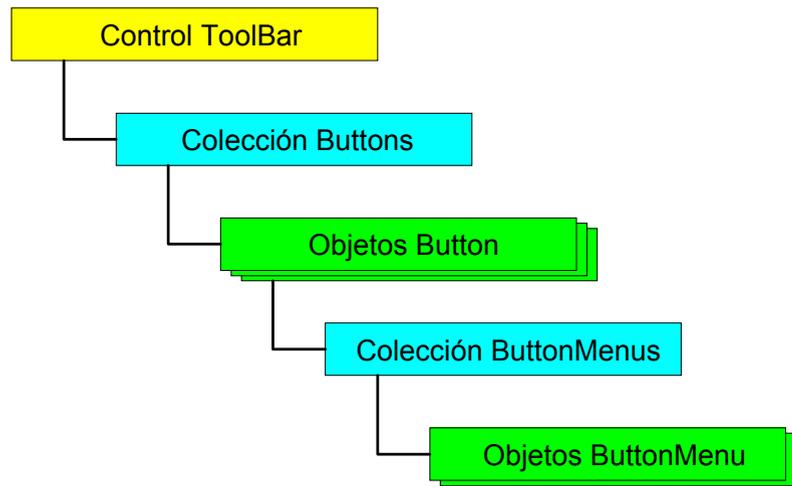
❑ Limitaciones.

- Los elementos de un array de menús deben ser adyacentes y pertenecer al mismo nivel.
- Los elementos aparecerán en orden ascendente del índice del array (aunque no tienen que ser correlativos).
- Creación dinámica de nuevas opciones.
 - ✓ Sólo en el mismo nivel.
 - ✓ Sólo en posiciones bien definidas (donde se hayan dejado huecos en el array).

```
Private Sub Command3_Click()  
    Dim i As Integer  
    i = mnuOtraOpcion.Count  
    Load mnuOtraOpcion(i)  
    mnuOtraOpcion(i).Caption = "Opción " & i  
End Sub  
  
Private Sub mnuOtraOpcion_Click(Index As Integer)  
    MsgBox "Ha pulsado la opcion " & Index  
End Sub
```

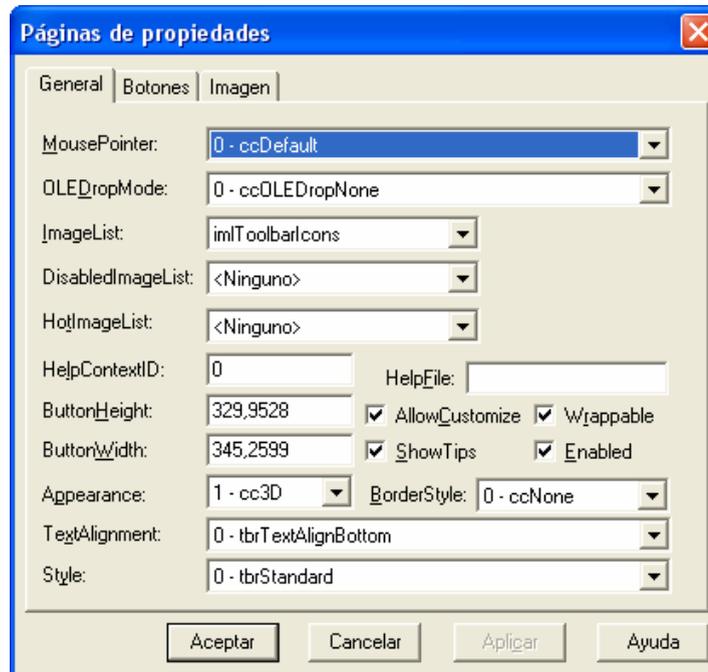
Control ToolBar (I)

- ❑ Hay que cargar el componente "Microsoft Windows Common Controls 6.0".
- ❑ Prefijo recomendado: tbr.
- ❑ Estructura del objeto.



Control ToolBar (II)

- ❑ Propiedades generales (personalizado).

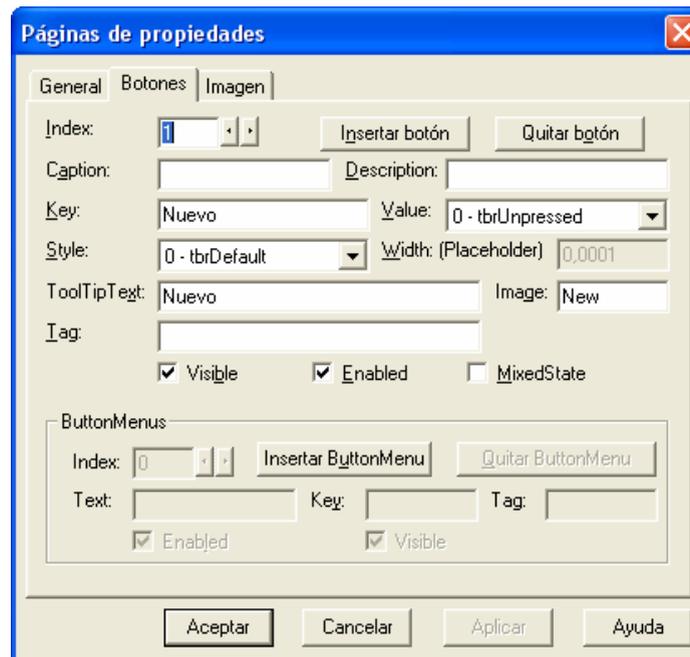


Control ToolBar (III)

Align	vbAlignNone (0), vbAlignTop (1), vbAlignBottom (2), vbAlignLeft (3), vbAlignRight (4).
AllowCustomize	Valor lógico que determina si el usuario puede personalizar la barra.
ShowTips	Valor lógico que determina si se visualizan los TipText.
Wrappable	Valor lógico que determina si la barra se ajusta al ancho del formulario.
ButtonWidth, ButtonHeight	Ancho y alto de los botones.
Style	Determina la apariencia del control: tbrStandard (0) o tbrFlat (1).
ImageList	Nombre del control ImageList que contiene los iconos de los botones. Los gráficos de los botones deben almacenarse en un control ImageList en el que se añade el gráfico y se determina una clave.

Control ToolBar (IV)

- ❑ Propiedades de los botones.
 - Por cada botón de la lista hay que añadir un objeto Button a la colección Buttons de la barra.



Control ToolBar (V)

Caption	Texto que aparece con cada botón
Descripción	Texto que aparecerá en las operaciones de configuración.
Key	Clave por la que se identificará el control.
ToITipText	Texto que aparecerá cuando esté activada la propiedad ShowTips
Image	Clave que identifica el icono dentro de la colección de imágenes del control ImageList.
Value	Estado del botón: tbrUnpressed (0), tbrPressed (1).
Style	Apariencia del botón: <ul style="list-style-type: none">● tbrDefault (0). Botón normal.● tbrCheck (1). Permanece pulsado cuando se presiona.● tbrButtonGroup (2). Grupo de botones, sólo uno puede aparecer pulsado.● tbrSeparator (3). Separador con el ancho estándar de un botón.● tbrPlaceHolder (4). Separador con la anchura especificada en la propiedad Width.● tbrDropDown (5). Aparece una flecha al lado del botón que permite elegir alguno de los objetos ButtonMenu asociados.

Control ToolBar (VI)

❑ Evento ButtonClick

```
Private Sub Toolbar1_ButtonClick(ByVal Button As  
MSComCtlLib.Button)
```

- Se produce cuando se pulsa un botón. El argumento Button es el objeto Button sobre el que se ha pulsado.

```
Select Case Button.Key
```

...

❑ Evento ButtonMenuClick

```
Private Sub Toolbar1_ButtonMenuClick(ByVal ButtonMenu As  
MSComCtlLib.Button)
```

Cuadros de diálogo comunes (I)

- ❑ El control CommonDialog ejecuta cuadros de diálogo de Windows.
 - Para utilizarlo hay que añadir el componente "Microsoft Common Dialog Control 6.0". 
- ❑ Propiedad name. Prefijo dlg.
- ❑ Métodos Show...
 - ShowOpen.
 - ShowSave.
 - ShowFont.
 - ShowColor.
 - ShowPrint.

Cuadro de diálogo Color

- Método ShowFont.
- Propiedad Color.
 - Devuelve el color seleccionado por el usuario.
- Propiedad Flags.

cdlCCFullOpen (&H2)	Presenta el cuadro de diálogo incluyendo la sección Definir colores personalizados.
cdlCCHelpButton (&H8)	El cuadro presenta un botón de Ayuda
cdlCCPreventFullOpen (&H4)	Desactiva el botón "Definir colores personalizados" evitando que el usuario defina su color personalizado
cdlCCRGBInit (&H1)	Establece el valor de color inicial del cuadro de diálogo.

```
Private Sub Command1_Click()  
    dlg.Flags = cdlCCRGBInit  
    dlg.ShowColor  
    BackColor = dlg.Color  
End Sub
```

Cuadro de diálogo Fuente (I)

- Método ShowFont.
- Propiedades relacionadas con las fuentes.

Color	Color de la fuente seleccionada. Precisa el Flag cdICFEffects.
FontName	Nombre de la fuente seleccionada.
FontSize	Tamaño de la fuente seleccionada.
FontBold	Determina si se ha seleccionado o no negrita.
FontItalic	Determina si se ha seleccionado o no cursiva.
FontStrikethru	Determina si se ha seleccionado o no tachado.
FontUnderline	Determina si se ha seleccionado o no subrayado.

Cuadro de diálogo Fuente (II)

- ❑ Propiedad Flags (es necesario establecer alguna de las cuatro primeros valores).

cdlCFScreenFonts	Muestra sólo la lista de fuentes de pantalla.
cdlCFPrinterFonts	Muestra sólo la lista de fuentes de impresora.
cdlCFBoth	Muestra las fuentes tanto de pantalla como de impresora.
cdlCFWYSIWIG	Muestra sólo las fuentes que estén disponibles tanto para la pantalla como para la impresora.
cdlCFEffects	Habilita los campos tachado, subrayado y color.

```
Private Sub Command1_Click()  
    dlg.Flags = cdlCFBoth + cdlCFEffects  
    dlg.ShowFont  
    With Label1  
        .FontName = dlg.FontName  
        .FontSize = dlg.FontSize  
        .FontBold = dlg.FontBold  
        .FontItalic = dlg.FontItalic  
        .FontUnderline = dlg.FontUnderline  
        .FontStrikethru = dlg.FontStrikethru  
    End With  
End Sub
```

Cuadro de diálogo Imprimir (I)

- Método ShowPrinter
- Propiedades.

Copies	Número de copias seleccionado.
FromPage, ToPage	Valor de la selección de los valores De y A del cuadro de diálogo.
Orientation	Determina el valor de la orientación: cdIPortait (1) o cdLandscape (2).
PrinterDefault	Determina si las selecciones efectuadas afectan a la impresora predeterminada del sistema.

Cuadro de diálogo Imprimir (II)

❑ Flags.

cdPDPrintSetup	Muestra el cuadro de diálogo configurar impresión en lugar del cuadro imprimir.
cdIPDHidePrintToFile	Oculto la casilla de verificación Imprimir en archivo.
cdIPDDisablePrintToFile	Desactiva la casilla de verificación Imprimir en archivo.
cdIPDNoPageNumbers	Desactiva el botón de opción Páginas.
cdIPDNoSelection	Desactiva el botón de opción Selección
cdIReturnDC	Devuelve el contexto de dispositivo de la impresora seleccionada. Dicho contexto se puede recoger con la propiedad hDc del control y ser utilizado por las API de Windows.
cdIReturnDC	Devuelve la información del contexto de dispositivo de la impresora seleccionada. Dicha información se puede recoger con la propiedad hDc del control y ser utilizado por las API de Windows.

Cuadro de diálogo Imprimir (III)

- ❑ Flags: verificar el estado de la selección.

cdIPDPrintToFile	Recoge el estado de la casilla de verificación Imprimir en archivo.
cdIPDAIIPages	Recoge el estado del botón de opción Todas las páginas.
cdIPDPageNums	Recoge el estado del botón de opción Páginas.
cdIPDSelection	Recoge el estado del botón de opción Selección.

- El estado de los flags se puede recoger después del método ShowPrinter

```
dlg.ShowPrinter
If dlg.Flags And cdIPDSelection Then
    MsgBox "Se ha pulsado selección"
Else
    'No se ha pulsado
End If
```

Cuadro de diálogo Imprimir (IV)

□ Imprimir: el objeto Printer.

- Método Print: coloca texto en el objeto Printer.

✓ `Printer.Print "Hola".`

- Método NewPage: envía un salto de página.
- Método KillDoc: anula la impresión.
- Método EndDoc: envía el texto a la cola de impresión.

```
Private Sub Command1_Click()  
    dlg.ShowPrinter  
    Dim i As Integer  
    For i = dlg.FromPage To dlg.ToPage  
        Printer.Print Text1  
        Printer.NewPage  
    Next  
    Printer.EndDoc  
End Sub
```

Cuadro de diálogo Abrir (I)

- ❑ Método ShowOpen.
- ❑ Propiedades.

DialogTitle	Título del cuadro de diálogo.
FileName	Nombre del archivo seleccionado.
InitDir	Directorio de inicio del cuadro de diálogo.
Filter	Lista que aparecerá en el combo Tipo de archivo. Es una expresión de cadena en la que por cada elemento debe aparecer una pareja formada por la descripción y el filtro separados por la barra vertical () <code>dlg.Filter = "Todos *.* JPGs *.jpg"</code>
FilterIndex	Índice del tipo de archivo por omisión de la lista Tipo.
DefaultExt	Cadena que indica la extensión por omisión que tendrá el archivo.
MaxFileSize	Tamaño máximo en bytes del nombre del archivo. Por omisión es de 256. Es posible que este tamaño no sea suficiente si se eligen múltiples archivos.

Cuadro de diálogo Abrir (II)

❑ Flags.

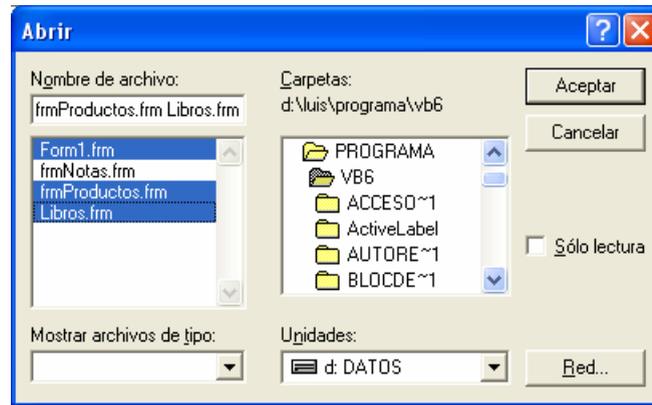
cdIOFNReadOnly	Estado de la casilla de verificación Abrir como sólo lectura.
cdIOFNHideReadOnly	Oculto la casilla de verificación Abrir como sólo lectura.
cdIOFNNoChangeDir	No modifica el directorio por omisión.
cdIOFNAllowMultiselect	Habilita la posibilidad de elegir múltiples archivos.
cdIOFNPathMustExist	Rechaza nombres de archivo con rutas de acceso inexistentes.
cdIOFNFileMustExist	Rechaza seleccionar archivos que no existan.
cdIOFNCreatePrompt	Si el archivo no existe pregunta si se desea crear uno nuevo.
cldOFNExplorer	Utiliza un cuadro de diálogo similar al explorador de Windows cuando se utiliza multiselección (en multiselección el cuadro de diálogo mostrado es el utilizado en aplicaciones antiguas de Windows).
cldOFNNoLongNames	Impide el empleo de nombres largos de archivo.

Cuadro de diálogo Abrir (III)

❑ Multiselección.

- Cuando se utiliza multiselección los nombres de los archivos seleccionados se devuelven en una única cadena en la propiedad FileName. El primer elemento sería la ruta de acceso.
 - ✓ Puede ser interesante aumentar el espacio reservado para almacenar los nombres de archivo (propiedad MaxFileSize).
- Cuando no se utiliza el flag `cdIOFNEexplorer` la ruta y los archivos se separan por un espacio en blanco.
- Cuando se utiliza el flag `cdIOFNEexplorer`, la ruta y los archivos se separan por el carácter null (`vbNullChar`).

Cuadro de diálogo Abrir (III)



```
dlg.Flags = cdlOFNAllowMultiselect
dlg.ShowOpen
Dim archivos() As String
'Los nombres de archivo aparecen separados por un espacio en blanco
'La función Split devuelve un array de elementos delimitados por un
'separador.
archivos() = Split(dlg.FileName, " ")
Dim i As Integer
For i = 0 To UBound(archivos)
    Debug.Print archivos(i)
Next
```

Cuadro de diálogo Guardar

- Método ShowSave.
- Funciona igual que el cuadro de diálogo ShowOpen.
- Flags.
 - `cdIOFNOOverWritePrompt`. Muestra un mensaje antes de sobrescribir un archivo existente.

El botón Cancelar de los CommonDialog

❑ Propiedad CancelError.

- Común a todos los cuadros de diálogo.
- Produce un error en tiempo de ejecución cuando en un cuadro de diálogo se pulsa el botón cancelar (error 32755, cdlCancel).
- Capturando el error es posible detectar la pulsación del botón.

```
On Error Resume Next
dlg.ShowOpen
If Err.Number = cdlCancel then
    'Acciones a realizar al pulsar el botón Cancelar
Else
    'Acciones a realizar al pulsar el botón Aceptar
End if
```

El objeto Clipboard (I)

- ❑ Proporciona acceso al portapapeles de Windows.
- ❑ Puede contener varios datos, siempre y cuando sean de distinto formato.
 - Método GetFormat(*formato*)
 - ✓ Devuelve un valor lógico `true` si el contenido del portapapeles contiene alguno de los formatos que se pasan como parámetro.
 - ✓ El valor de *formato* puede ser:

Constante	Valor	Descripción
vbCFLink	&HBF00	Información de conversación DDE
vbCFText	1	Texto
VbCFBitmap	2	Mapa de bits(archivos .bmp)
vbCFMetafile	3	Metarchivo(archivos .wmf).
vbCFDIB	8	Mapa de bits independiente del dispositivo (DIB)
vbCFPalette	9	Paleta de colores

El objeto Clipboard (II)

- ❑ Función que devuelve el tipo de contenido del portapapeles.

```
Private Function ContenidoPortapapeles() As Integer
    'Devuelve 0 (vacío),1 (sólo texto),2 (sólo mapa de bits),
    '3 (texto y mapa de bits) o 4 (sólo texto RTF)
    Dim ClpFmt
    If Clipboard.GetFormat(vbCFText) Then ClpFmt = ClpFmt + 1
    If Clipboard.GetFormat(vbCFBitmap) Then ClpFmt = ClpFmt + 2
    If Clipboard.GetFormat(vbCFDIB) Then ClpFmt = ClpFmt + 4
    If Clipboard.GetFormat(vbCFRTF) Then ClpFmt = ClpFmt + 8
    Select Case ClpFmt
        Case 1 'Sólo tiene texto
            ContenidoPortapapeles = 1
        Case 2, 4, 6 'Sólo tiene gráficos
            ContenidoPortapapeles = 2
        Case 3, 5, 7, 14 'Tiene texto y gráficos
            ContenidoPortapapeles = 3
        Case 8, 9 'Tiene texto RTF
            ContenidoPortapapeles = 4
        Case Else
            ContenidoPortapapeles = 0
    End Select
End Function
```

El objeto Clipboard (III)

❑ Obtener el contenido del portapapeles.

- Método `GetData ([formato])`.
 - ✓ Devuelve el un gráfico del portapapeles.
 - ✓ Si no se utiliza el argumento `formato`, elige el más adecuado.
 - ✓ Formato puede tener los valores `vbCFBitmap`, `vbCFMetafile`, `vbCFDIB` o `vbCFPalette`.
- Método `GetText ([formato])`.
 - ✓ Devuelve el contenido de de texto del portapapeles.
 - ✓ `formato` puede tener los valores `vbCFLink`, `vbCFText` (predeterminado) o `vbCFRTF`.

El objeto Clipboard (IV)

□ Introducir contenido en el portapapeles.

- Método `SetData datos, [formato]`.
 - ✓ Introduce una imagen en el objeto Clipboard en el formato especificado.
 - ✓ Si se omite el formato determina el formato adecuado.
 - ✓ La imagen a insertar se puede obtener con la propiedad `Picture` de los controles `Form`, `Picture` o `Image` o mediante la propiedad `SelRTF` del objeto `RichTextBox`.
- Método `SetText datos, [formato]`.
 - ✓ Introduce una cadena de texto en el objeto Clipboard en el formato especificado.
 - ✓ Si se omite el formato lo introduce en formato de texto simple.

El objeto Clipboard (V)

- ❑ Copiar el texto seleccionado en el portapapeles.

```
Clipboard.SetText Text1.Text
```

- ❑ Pegar el texto del portapapeles en un TextBox

```
Text1.Text = Clipboard.GetText(vbCFText)
```

- ❑ Cortar el texto seleccionado en el portapapeles.

```
Clipboard.SetText Text1.Text
```

```
Text1.SelectionStart = 0
```

- ❑ Copiar un objeto gráfico en el portapapeles.

```
Clipboard.SetData Picture1.Picture
```

- ❑ Pegar un objeto gráfico en un PictureBox.

```
Picture1.Picture = Clipboard.GetData()
```