

Comencemos a programar con
VBA - Access

Entrega **16**

Código vs. Macros
Objeto DoCmd

Código frente a macros

Access posee un potente juego de macros que permiten un alto grado de automatización en una serie de tareas.

La mayor parte de los cursos de Access impartidos en los diversos centros de formación, incluyen en su última parte una introducción al manejo de las macros.

La cruda realidad es que la mayoría de las veces apenas si se llega a mostrar su manejo, si no es en un curso de los llamados “avanzados”.

En este momento no pretendo explicar cómo se manejan las Macros, pero ya que podemos vernos en la necesidad de transformar macros en código haré una pequeña introducción.

Si vamos a la ayuda de Access veremos que una macro se define como un conjunto de acciones creadas para automatizar algunas de las tareas comunes.

Veamos en la práctica cómo se manejan.

Vamos a crear un formulario y en él pondremos una etiqueta grande con un texto en rojo, por ejemplo **Formulario abierto**.

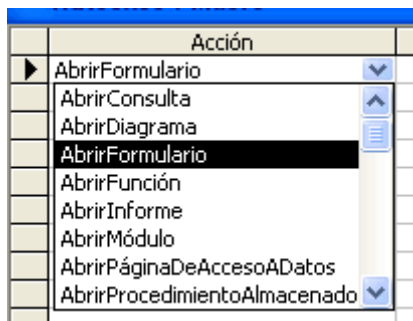
Guardamos el formulario con el nombre **PruebaMacro**.

Abrimos la ventana macros, pinchando en la pestaña macros y en el botón **[Nuevo]**.

Una vez dentro podremos ver la columna **Acción**, y si activáramos la opción **Condiciones** del menú **Ver** nos aparecerá la columna **Condición**.

En la primera fila de columna Acción seleccionamos la acción **[AbrirFormulario]**.

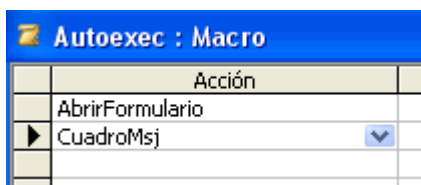
Inmediatamente después de hacerlo, nos aparece en la parte inferior unos cuadros en los que podemos introducir los **argumentos** de la macro.



Nombre del formulario	PruebaMacro
Vista	Formulario
Nombre del filtro	
Condición WHERE	
Modo de datos	
Modo de la ventana	Normal

Si nos ponemos en el cuadro **Nombre del formulario**, vemos que nos aparece una flecha hacia abajo que nos permite seleccionar un formulario.

Si la base de datos era nueva nos aparecerá únicamente el formulario **PruebaMacro** que acabamos de guardar. Seleccionaremos ese formulario.



Mensaje	Formulario abierto
Bip	Sí
Tipo	Información
Título	

Vamos a la fila siguiente de la columna Acción, y seleccionamos la acción **[CuadroMsj]**, y en los cuadros de abajo ponemos:

En Mensaje: **Formulario abierto**, y en Tipo: **Información**.

A continuación presionamos en el botón guardar, asignándole como nombre **Autoexec**. Cerramos la ventana de macros, e incluso Access.

Volvemos a abrir el fichero mdb y veremos que ahora se abre el formulario PruebaMacro inmediatamente después de cargar el fichero. y nos muestra una cuadro de mensaje tipo a los que ya hemos visto con la función **MsgBox** en la Entrega 12.

Hemos comprobado lo siguiente: si a una macro, le ponemos como nombre **Autoexec**, se ejecutarán sus acciones en el momento de abrir el fichero Access.

Este es un comportamiento interesante al que en algún momento podremos sacar partido.

¿Y a cuento de qué viene ahora el hablar de las macros?. ¿No es ir hacia atrás?

Las macros ha sido una herramienta muy utilizada por “usuarios avanzados de Access” que no tenían conocimientos de programación y que querían dotar a sus aplicativos de cierto grado de automatismo.

Lo que quizás algunos de esos usuarios no sabían es que Access permite realizar una conversión directa de las macros a código VBA.

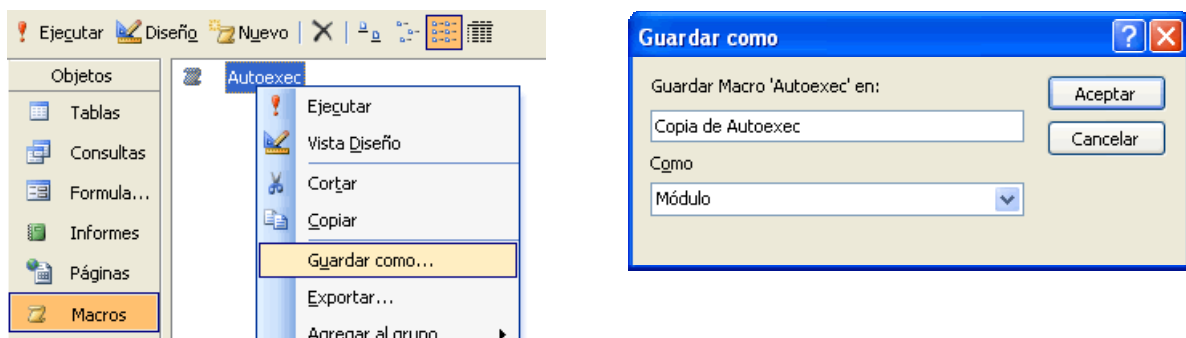
Las funciones y procedimientos de VBA tienen más flexibilidad y posibilidades que las macros. Además las macros no pueden realizar tareas como el control de errores, o el acceso a un único registro con la misma simplicidad que con VBA.

Vamos a la pestaña de Macros, colocamos el cursor encima del nombre de la macro Autoexec, y presionamos el botón derecho del ratón.

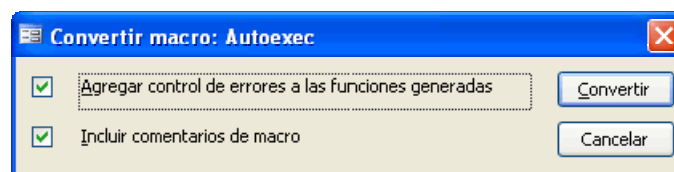
Nos aparecerá un menú contextual en el que seleccionaremos la opción [Guardar como...].

Tras esto nos aparece una nueva ventana en la que se nos propone guardar la macro con el nombre Copia de Autoexec.

En el segundo cuadro seleccionamos [Módulo] y presionamos el botón [Aceptar].



Tras ello nos aparece una nueva ventana en la que nos sugiere que la exportación a un módulo la va a efectuar con control de errores y comentarios.



No cambiamos esos valores y presionamos el botón [Convertir].

Si ahora nos vamos a la ventana de Módulos veremos que aparece un nuevo módulo llamado **Macro convertida - Autoexec**.

El proceso de conversión de macros se puede también realizar siguiendo estos pasos

1. Seleccione la macro a convertir en la ventana de macros.
2. Seleccionamos la opción de menú **[Herramientas]** y dentro de ella **[Macro]**
3. Del menú que se despliega seleccionamos **[Convertir macros a Visual Basic]**
4. Nos aparece el cuadro Convertir macro del punto anterior.

Se pueden convertir tanto las macros generales como las macros asociadas a formularios o informes.

Si abrimos el módulo generado, podremos ver que contiene el siguiente código.

```
Option Compare Database
```

```
Option Explicit
```

```
'-----
' Autoexec
'
'-----

Function Autoexec()
On Error GoTo Autoexec_Err

    DoCmd.OpenForm "PruebaMacro", acNormal, "", "", , acNormal
    Beep
    MsgBox "Formulario abierto", vbInformation, ""

Autoexec_Exit:
    Exit Function

Autoexec_Err:
    MsgBox Error$
    Resume Autoexec_Exit

End Function
```

Vemos que la macro ha sido cambiada por un módulo que contiene la función **Autoexec**.

*Quiero suponer que en vez de un procedimiento **sub**, genera una función que no devuelve explícitamente ningún valor, para que pueda ser utilizada directamente desde la pestaña eventos de la ventana de propiedades de un formulario, informe o control. Ya hablaremos más adelante de este tema...*

Esta función tiene dos partes que se corresponden con las dos acciones de la macro.

La acción **AbrirFormulario** se sustituye por el método **OpenForm** del objeto **DoCmd**.

La acción **CuadroMsj** se sustituye por la función **MsgBox** que ya hemos visto.

Haré mención aparte a la instrucción **Beep**.

Esta instrucción en teoría sirve para que el equipo emita un sonido de advertencia.

Dependiendo de la tarjeta de sonido que se tenga, y la configuración del software, se oirá un pitido, un sonido o incluso puede que no se oiga absolutamente nada.

Las mismas acciones que hemos puesto en la macro **Autoexec**, podríamos haberlas puesto en un mismo fichero de Macro, como dos macros independientes.

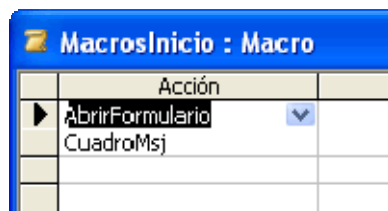
Para ello vamos a hacer lo siguiente:

Desde la ventana de Macros, ponemos el cursor del ratón encima de la macro Autoexec y presionamos el botón derecho del ratón (el izquierdo para los zurdos que así lo tengan configurado).

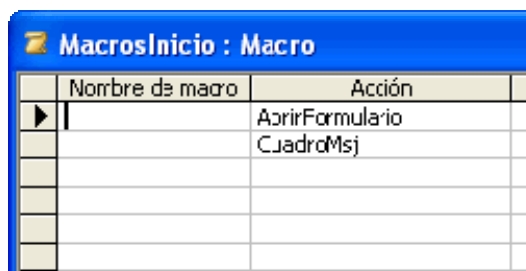
Nos aparecerá un menú contextual, una de cuyas opciones es **Cambiar nombre**.

Cambiamos el nombre de la macro **Autoexec**, por el de **MacroInicio**.

A continuación pulsamos el botón de Diseño (el la escuadra, la regla y el lápiz) y se nos abrirá la macro en vista diseño.



MacroInicio : Macro	
	Acción
▶	AbrirFormulario
	CuadroMsj



MacroInicio : Macro	
Nombre de macro	Acción
▶	AbrirFormulario
	CuadroMsj

Vamos a activar, si no lo estuviera, una columna a la izquierda que contendrá el nombre de la macro.

Recordemos que hay que hacer visible, en el editor de macros, la columna correspondiente al nombre de macro. Para ello, desde el editor de Macros, en la opción de menú **[Ver]** activamos la opción **[Nombres de macro]**.

Tras esto nos aparecerá una columna vacía a la izquierda.

Por pura cuestión de orden vamos a separar las dos acciones, que se convertirán en macros independientes.

Para ello ponemos el cursor a la izquierda de la celda donde aparece CuadroMsj, y pulsamos dos veces en la opción de menú **[Insertar] [Filas]**.

Ahora nos ponemos a la izquierda de la primera Acción (AbrirFormulario) y escribimos como nombre para esa macro **Autoexec**.

En la celda que está a la izquierda de la acción CuadroMsj escribimos como nombre de la macro algo tan original como **Mensaje**.

Con esto ya tenemos dos macros diferentes escritas en el mismo fichero de macros.

En el caso anterior teníamos una única macro llamada Autoexec que ejecutaba 2 acciones diferentes.

Ahora, al arrancar el fichero de access se ejecutará la macro **Autoexec**, que nos abrirá el formulario, pero no se ejecutará la acción de la macro **Mensaje**. Puedes comprobarlo cerrando el fichero access y volviéndolo a abrir.

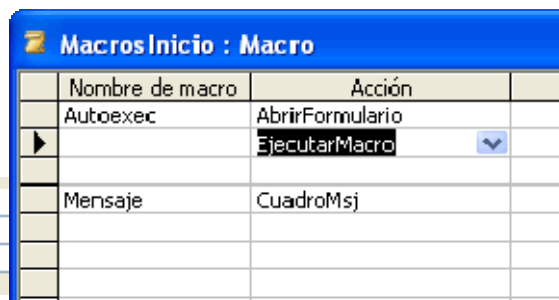
Para solventar este problema, podemos llamar a la macro Mensaje desde la macro Autoexec. Lo podemos conseguir mediante la Acción EjecutarMacro.

En la columna Acción, nos ponemos debajo de la celda en la que está escrita la acción AbrirFormulario y seleccionamos la acción EjecutarMacro.

A continuación seleccionamos como parámetro la macro **MacroInicio.Mensaje**.

Nombre de macro
Número de repeticiones
Expresión de repetición

MacroInicio.Mensaje



MacroInicio : Macro	
Nombre de macro	Acción
▶	EjecutarMacro
	CuadroMsj
	CuadroMsj

Guardamos todo y vemos que ahora sí que se muestra el cuadro de mensaje para indicarnos la apertura del formulario.

¿Cómo afectaría este cambio en las macros al código que se generaría con el conversor de macros a código VBA?

La respuesta la podemos obtener de forma inmediata.

Guardamos la macro, si es que no lo habíamos hecho, y desde la ventana macro, seleccionamos nuestra flamante **MacroInicio** y activamos la opción de menú **[Herramientas] [Macro] [Convertir Macros a Visual Basic]**

Se nos genera un nuevo módulo, esta vez con el nombre

Macro convertida- MacroInicio.

El código que contiene ese módulo es el siguiente

```
Option Compare Database
Option Explicit

'-----
' MacroInicio_Autoexec
'
'-----

Function MacroInicio_Autoexec()
On Error GoTo MacroInicio_Autoexec_Err

    DoCmd.OpenForm "PruebaMacro", acNormal, "", "", , acNormal
    DoCmd.RunMacro "MacroInicio.Mensaje", , ""

MacroInicio_Autoexec_Exit:
    Exit Function

MacroInicio_Autoexec_Err:
    MsgBox Error$
    Resume MacroInicio_Autoexec_Exit

End Function

'-----
' MacroInicio_Mensaje
'
'-----

Function MacroInicio_Mensaje()
On Error GoTo MacroInicio_Mensaje_Err
```

```
Beep
MsgBox "Formulario abierto", vbOKOnly, ""
```

```
MacrosInicio_Mensaje_Exit:
```

```
Exit Function
```

```
MacrosInicio_Mensaje_Err:
```

```
MsgBox Error$
```

```
Resume MacrosInicio_Mensaje_Exit
```

```
End Function
```

Así como la primera vez nos creó un módulo con una única función, al tener ahora dos macros en un mismo fichero de macros, nos crea un módulo con dos funciones.

¿Cuándo usar Macros y cuándo código VBA?

Las macros permiten automatizar, de una forma simple, tareas como Abrir un formulario, cerrar un informe o desplazarnos entre los registros de una tabla.

Si acudimos a la ayuda de Access, en el apartado con título semejante vemos que nos recomienda el uso de macros fundamentalmente para realizar asignaciones globales de teclas. Incluso para la realización de determinadas acciones al arrancar la base de datos, comenta la posibilidad de utilizar la opción Inicio del menú Herramientas.

Para el resto de las acciones recomienda la utilización de VBA.

La mayor parte de las acciones ejecutables mediante macros, pueden ser realizadas ventajosamente usando VBA mediante los métodos del objeto **DoCmd**, que las implementa.

Como recordatorio indicaré que si por ejemplo quisiéramos que al abrir una base de datos se realice una determinada acción podemos usar una macro llamada Autoexec.

Como también hemos comentado, ésta no es la única opción, ya que podríamos hacer que aparezca un formulario concreto al arrancar la aplicación, mediante la opción **[Inicio]** del menú **[Herramientas]** y efectuar las llamadas a los procedimientos que nos interesen desde el código de ese formulario.

El objeto DoCmd

El objeto **DoCmd** es un objeto específico de Access, creado para sustituir a las acciones de las Macros. De hecho sustituye con ventaja a casi todas ellas.

Hasta Access 97, no existía ese objeto, pero sí existía el procedimiento **DoCmd**.

Por tanto el objeto **DoCmd** empezó a existir con Access 97.

Los argumentos de la acción serán ahora los argumentos del método de **DoCmd**.

En la acción **AbrirFormulario**, poníamos como Nombre del formulario **PruebaMacro**, como vista **Formulario** y como modo de la ventana **Normal**.

Al método **OpenForm**, que abre un formulario, le pasamos como nombre del formulario (**FormName**) **"PruebaMacro"**, como vista (**View**) **acNormal**, y como tipo de ventana (**WindowMode**) **acNormal**.

Las constante **acNormal** está definida como una constante enumerada miembro de Access.AcFormView. Su valor numérico es 0.

Ya comentamos las **Constantes Enumeradas** en la entrega 12. Este tipo de constantes van a ir apareciendo con mucha frecuencia conforme vayamos avanzando en VBA.

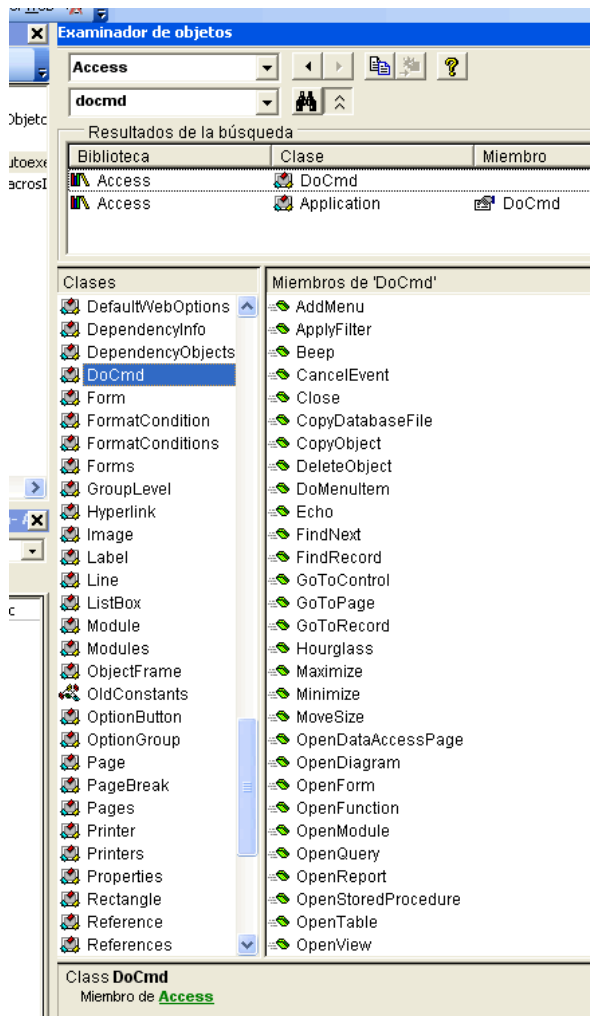
Como habrás podido ver hay un paralelismo total entre la acción de macro y el método correspondiente de **DoCmd**.

Pero no todas las Acciones de las macros están implementadas en los métodos de **DoCmd**.

Por ejemplo, en la conversión de macros a código VBA hemos visto que **CuadroMsj** se sustituye por la función **MsgBox**.

Otras acciones no implementadas en **DoCmd** son

Acción	Equivalencia en VBA
RunApp	Función Shell
RunCode	Llamada a la subrutina correspondiente
SendKeys	Instrucción SendKeys
SetValue	Operador = de asignación
StopAllMacros	Instrucciones Stop o End
StopMacro	Instrucciones Exit Sub o Exit Function



Podemos ver directamente los métodos implementados en la clase **DoCmd**, mediante la ventana del Examinador de objetos. Para activar esa ventana pulsaremos la tecla [F2] desde el editor de código.

Para localizar la clase **DoCmd** seleccionamos la biblioteca de **Access** y escribimos el nombre de la clase que queremos buscar.

Tras pulsar la tecla de búsqueda se nos posicionará en la clase **DoCmd**.

En la columna de la derecha podremos ver sus métodos.

Se puede comprobar que, al contrario de otras clases, no posee ni propiedades ni eventos.

Para la distinción entre Clase, Objeto, Propiedad, Evento y Método, os remito a las anteriores entregas en las que se daba una explicación a estos conceptos, así como a entregas posteriores en las que se profundizará sobre los mismos.

A continuación se describen los métodos implementados en **DoCmd**, junto con algunos pequeños ejemplos de código.

Si se requiriera una información más extensa se puede acudir a la ayuda de Access.

Método	Funcionalidad
AddMenu	<p>Permite crear barras de menú y menús contextuales.</p> <p>Se le pasa el <i>Nombre del menú</i>, el <i>Nombre de la macro de menú</i> y el <i>Texto de la barra de estado</i>.</p>
ApplyFilter	<p>Permite aplicar un filtro, una consulta o una cláusula "Tipo WHERE" de una instrucción SQL a una tabla, a un formulario o a un informe.</p> <p>Se le pasan como parámetros el <i>Nombre del filtro</i> y la <i>Condición</i> que aparecería después del Where.</p> <p>La siguiente línea de código hace que el formulario en donde está ubicada muestre sólo aquellos registros cuyo campo Marca comience por A, B ó C.</p> <pre>DoCmd.ApplyFilter , "Marca Like '[ABC]*'"</pre>
Beep	Se puede utilizar para emitir un sonido por los altavoces del PC
CancelEvent	Se utiliza para cancelar un evento. Sólo tiene efecto cuando se ejecuta como resultado de un evento. No usa parámetros.
Close	<p>Permite cerrar la ventana de un objeto.</p> <p>Hay que pasarle como parámetros el <i>Tipo</i> y el <i>Nombre del objeto</i>. También se le puede pasar un tercer argumento para indicar si se guardan o no los posibles cambios.</p> <p>Este código cierra el formulario actual</p> <pre>DoCmd.Close</pre> <p>Esta línea cierra el formulario PruebaMacro y guarda los posibles cambios sin preguntar si deben o no ser guardados.</p> <pre>DoCmd.Close acForm, " PruebaMacro", acSaveYes</pre>
CopyDatabaseFile	<p>Copia la base de datos conectada al proyecto actual en un archivo de base de datos de Microsoft SQL Server para la exportación.</p> <p>Los parámetros son <i>Nombre del Archivo Base De Datos</i>, <i>Sobrescribir un Archivo Existente</i> y <i>Desconectar Todos Los Usuarios</i>.</p> <p>Este ejemplo está sacado de la ayuda de Access.</p> <p>Copia la base de datos conectada al proyecto activo en un archivo de base de datos SQL Server.</p> <p>Si el archivo ya existiera, lo sobrescribe, y se desconectan todos los demás usuarios que estén conectados a la base de datos antes de realizar la copia.</p> <pre>DoCmd.CopySQLDatabaseFile _ DatabaseFileName:="C:\Export\Sales.mdf", _ OverwriteExistingFile:=True, _ DisconnectAllUsers:=True</pre> <p>Para pasar los parámetros utiliza parámetros "Con Nombre" mediante el operador de asignación :=.</p>
CopyObject	Permite copiar objetos de la base de datos, como Tablas,

Consultas, Formularios, Informes, Módulos, etc... en la base de datos actual o en otra que se especifique.

Permite incluso copiar un objeto cambiándole el nombre.

Como parámetros se pasa el *Nombre de la base de datos destino*, el *Nuevo nombre* que vaya a tener el objeto copiado, el *Tipo de objeto* a copiar y el *Nombre* del mismo.

Este código copia la tabla Marcas en la tabla CopiaMarcas

```
DoCmd.CopyObject, "CopiaMarcas", acTable, "Marcas"
```

Si se aplica sobre tablas vinculadas crea un nuevo vínculo.

DeleteObject

Elimina un objeto de una base de datos.

Hay que pasarle como parámetros el *Tipo* y el *Nombre* del objeto.

Este ejemplo borra la tabla CopiaMarcas.

```
DoCmd.DeleteObject acTable, "CopiaMarcas"
```

Si se aplica sobre tablas vinculadas elimina sólo el vínculo.

DoMenuItem

Este método es un método obsoleto, mantenido por compatibilidad con las versiones anteriores a Access 97.

Ha sido sustituido por [RunCommand](#).

Consulte la ayuda para mayor información

Echo

Se utiliza para mostrar u ocultar los resultados de la ejecución de una macro mientras se está ejecutando.

```
DoCmd.Echo True
```

Para desactivar **Eco** mostrando un mensaje

```
DoCmd.Echo False, "Ejecutándose código de VBA"
```

FindNext

Busca el siguiente registro que cumpla las condiciones definidas previamente mediante el método [FindRecord](#).

FindRecord

Busca un registro que cumpla determinados criterios.

Usa los mismos criterios que la opción de menú [Edición] [Buscar].

Los criterios son *Cadena a buscar*, *Coincidencia* de la cadena de búsqueda, *Distinguir mayúsculas*, *Sentido* a buscar, buscar en *Campo activo* o en *Todos*, *Buscar el primer registro* o en *Siguientes*, buscar en *Campos con formato*, en el *Campo activo* y buscar *Primero*

El parámetro *Coincidencia* toma una de las siguientes constantes del tipo AcFindMatch

```
acAnywhere
acEntire (valor predeterminado)
acStart
```

Como *Sentido* toma una de las constantes AcSearchDirection

```
acDown
acSearchAll (valor predeterminado)
acUp
```

Como *Campo activo* toma una de las constantes AcFindField

```
acAll
```

acCurrent (valor predeterminado)

La siguiente instrucción busca el primer registro que contenga el texto "Olaz" en cualquiera de sus campos, distinguiendo entre mayúsculas y minúsculas, considerando también los campos con formato.

```
DoCmd.FindRecord "Olaz", _
    acAnywhere, _
    True, _
    acSearchAll, _
    True
```

GoToControl

Desplaza el foco al campo o control especificado.

El siguiente código desplaza el foco al control txtFecha del formulario.

```
DoCmd.GoToControl "txtFecha"
```

GoToPage

Lleva el foco al primer control de la página especificada en un formulario en el que se han colocado saltos de página, o a una ficha determinada de un control ficha.

El parámetro *Número* de página es obligatorio.

Los parámetros opcionales posición *Horizontal* y posición *Vertical* son útiles si el formulario es mayor que el tamaño de la ventana de Windows. Este código traslada el foco al primer control de la página 3 del formulario.

```
DoCmd.GoToPage 3
```

GoToRecord

La acción GoToRecord (IrARegistro) convierte el registro especificado en el registro actual de un conjunto de resultados de una tabla, formulario o consulta.

Admite los siguientes parámetros:

Tipo de objeto, *Nombre*, *Registro* (Anterior, Siguiente, Primero, Último, Ir a o Nuevo), *Número* de registros a desplazar.

Como tipo de objeto se utiliza una constante AcDataObjectType

Esta constante puede tomar los valores

```
acActiveDataObject (Valor predeterminado)
acDataForm
acDataFunction
acDataQuery
acDataServerView
acDataStoredProcedure
acDataTable
```

El parámetro *Registro* puede tomar el valor de una constante del tipo AcRecord. Sus posibles valores son

```
acFirst
acGoTo
acLast
acNewRec
acNext (Valor predeterminado)
acPrevious
```

Este código activa el séptimo registro del formulario "Empleados".

```
DoCmd.GoToRecord acDataForm, "Empleados",
acGoTo, 7
```

Hourglass	<p>Hace aparecer un reloj de arena en vez del cursor normal del ratón.</p> <p>Si el parámetro es False, vuelve a colocar el cursor por defecto.</p> <pre>DoCmd.Hourglass True</pre>
Maximize	<p>Maximiza la ventana actual hasta ocupar totalmente la ventana de Windows. No tiene parámetros.</p> <pre>DoCmd.Maximize</pre>
Minimize	<p>Minimiza la ventana actual y la coloca en la parte inferior de la ventana de Microsoft Access.</p> <pre>DoCmd.Minimize</pre>
MoveSize	<p>Permite desplazar y cambiar el tamaño de la ventana activa. Se corresponde con la macro DesplazarTamaño.</p> <p>Sus parámetros son: <i>PosiciónHorizontal</i> y <i>Vertical</i> de la esquina superior izquierda, más la <i>Anchura</i> y <i>Altura</i> nuevos de la ventana. Los cuatro parámetros son opcionales.</p> <p>Este código cambia la posición izquierda de la ventana actual, así como su altura.</p> <pre>DoCmd.MoveSize 3000, , , 4000</pre>
OpenDataAccessPage	<p>Abre una página de acceso a datos.</p> <p>Sus parámetros son el <i>Nombre</i> de la página y el <i>Modo de apertura</i>.</p> <p>En el parámetro <i>Modo de apertura</i> se utiliza una de las constantes del tipo <code>AcDataAccessPageView</code>.</p> <p>Sus valores pueden ser</p> <pre>acDataAccessPageBrowse (predeterminado) acDataAccessPageDesign (Vista Diseño)</pre> <p>El siguiente código abre la página Alumnos en modo Examinar.</p> <pre>DoCmd.OpenDataAccessPage "Alumnos", _ acDataAccessPageBrowse</pre>
OpenDiagram	<p>Este método sirve, en un fichero de Proyecto (.adp) conectado a una base de datos SQLServer, para abrir en modo diseño un diagrama de relaciones definido en la base de datos.</p> <p>Como parámetro se introduce el <i>Nombre</i> del diagrama.</p> <p>En esta línea se abre el diagrama guardado previamente como <code>ModeloDeDatos</code>.</p> <pre>DoCmd.OpenDiagram "ModeloDedatos"</pre>
OpenForm	<p>Abre el formulario especificado</p> <p>Se pueden pasar hasta 7 parámetros que controlarán el <i>Nombre</i> del formulario a abrir, el <i>Tipo</i> de vista como se <i>Mostrará</i> el formulario, el <i>Filtro</i> que se aplicará a los datos, una <i>Cláusula Where</i> que define las características que deben cumplir los datos, el <i>Modo como se editarán</i> (sólo lectura, etc.) y el <i>Modo como se mostrará</i> la ventana. Adicionalmente se le puede incluir los <i>Argumentos de apertura</i>.</p> <p>Como <i>Tipo</i> de vista se puede usar una de las siguientes constantes <code>AcFormView</code></p>

acDesign
acFormDS
acFormPivotChart
acFormPivotTable
acNormal (predeterminado)
acPreview

El Modo de edición toma una constante AcFormOpenDataMode

acFormAdd Se puede agregar registros nuevos pero no se pueden modificar los existentes.

acFormEdit Se pueden modificar los registros existentes y agregar registros nuevos.

acFormPropertySettings (predeterminado)

acFormReadOnly Los registros sólo se pueden ver.

El *Modo como se mostrará* la ventana toma uno de los valores AcWindowMode

acDialog El formulario será Modal y Emergente.

acHidden El formulario estará oculto.

acIcon Se abre minimizado.

acWindowNormal valor predeterminado

La siguiente línea abre el formulario Alumnos, en modo de sólo lectura, con los datos de los alumnos residentes en Pamplona.

```
DoCmd.OpenForm "Alumnos", , , _
"Poblacion = 'Pamplona'", acFormReadOnly
```

OpenFunction

Abre una función definida por el usuario en una base de datos de Microsoft SQL Server para verla desde Microsoft Access.

Los parámetros son el nombre de la función, la *Vista* como se va a mostrar la función, y el *Modo* de visualización o edición de los datos.

La Vista puede ser una constante del tipo AcView.

acViewDesign Abre la función en la vista Diseño.

acViewNormal (predeterminado).

acViewPivotChart Vista Gráfico dinámico.

acViewPivotTable Vista Tabla dinámica.

acViewPreview Abre la función en la Vista preliminar.

El modo de visualización es del tipo AcOpenDataMode.

acAdd Abre la función para la inserción de datos.

acEdit (predeterminado). Abre la función para actualizar los datos existentes.

acReadOnly Abre la función en Sólo lectura.

La siguiente línea abre la función Ajuste en vista normal y en el modo edición de datos.

```
DoCmd.OpenFunction "Ajuste", , acEdit
```

OpenModule

Abre un procedimiento dentro de un módulo en el modo diseño.

Como parámetros se pueden pasar el *Nombre del módulo* y el del *Procedimiento* a abrir.

Si sólo se indica el nombre del módulo se abre en el primer procedimiento del mismo.

Si sólo se indica el nombre del procedimiento, lo busca entre

todos los módulos y si lo encuentra lo abre en el mismo.

Si no existiera el módulo generará el error 2516.

Si no existiera el procedimiento generará el error 2517.

Esta línea abre el módulo Utilidades en la función Edad.

```
DoCmd.OpenModule "Utilidades", "Edad"
```

OpenQuery

Se puede utilizar para abrir una consulta de selección o una consulta de referencias cruzadas en la *Vista* Hoja de datos, diseño o en vista preliminar.

Esta acción ejecuta una consulta de acción. También se puede seleccionar un modo de entrada de datos para la consulta.

Los parámetros a introducir son nombre de la consulta, modo de presentación y *Modo de edición* de los datos.

El parámetro *Vista* es del tipo AcFormView ya visto en **OpenForm**.

El *Modo de edición* de los datos es del tipo AcOpenDataMode

acAdd

acEdit (predeterminado)

acReadOnly

El siguiente código ejecutará una consulta de acción llamada ctaActualizarPrecios.

```
DoCmd.OpenQuery "ctaActualizarPrecios"
```

Este código abrirá una consulta llamada ctaPrecios en modo Vista Previa.

```
DoCmd.OpenQuery "ctaPrecios ", acViewPreview
```

Este código abrirá la misma consulta pero en modo edición.

```
DoCmd.OpenQuery "ctaPrecios ", , acEdit
```

OpenReport

Abre el informe especificado

Los parámetros controlarán el nombre del informe, la vista preliminar o en modo diseño, el filtro que se aplicará a los datos, y la condición Where que deben cumplir los datos, el modo de presentación de la ventana del informe, pudiéndose incluir argumentos de apertura.

La siguiente línea abre el Informe infAlumnos, en vista previa, con los datos de los alumnos residentes en Pamplona.

```
DoCmd.OpenReport "infAlumnos", _
    acViewPreview, , _
    "Poblacion = 'Pamplona'
```

OpenStoredProcedure

Abre un procedimiento almacenado de una base de datos SQLServer desde un archivo de proyecto .adp

Como argumentos se pasan el Nombre del procedimiento, el *Modo de la Vista* del tipo **AcView** visto en **OpenFunction** y el *Modo de edición* de los datos del tipo AcOpenDataMode visto en **OpenQuery**.

El siguiente código abre el procedimiento almacenado ControlAlumnos en el modo Normal pero de sólo lectura.

```
DoCmd.OpenStoredProcedure "ControlAlumnos", _
    acViewNormal", _
    acReadOnly
```

OpenTable

Abre una tabla en vista hoja de datos, diseño o vista previa, y permite, en su caso, especificar el *Modo de la Vista* (tipo AcView) y el del *Modo de edición* de los datos (tipo AcOpenDataMode).

El siguiente código abre la tabla Alumnos en el modo Normal pero de sólo lectura.

```
DoCmd.OpenTable "Alumnos", _
    acViewNormal", _
    acReadOnly
```

OpenView

Abre una vista de una base de datos SQLServer desde un archivo de proyecto .adp

La vista puede estar en modo hoja de datos, diseño o vista previa.

Como parámetros se pasan el *Nombre* de la vista, el *Modo como se va a ver* (AcView) y el *Modo de edición* de los datos (tipo AcOpenDataMode).

El siguiente código abre la vista AlumnosMatriculados en el modo tabla dinámica de sólo lectura.

```
DoCmd.OpenView "AlumnosMatriculados", _
    acViewPivotTable, _
    acReadOnly
```

OutputTo

Este método permite presentar los datos de un objeto (tabla, consulta, formulario, informe, módulo, página de acceso a datos, vista, procedimiento almacenado o función) en formato HTML (*.htm; *.html), texto (*.txt), Microsoft Active Server Pages (*.asp), Excel (*.xls), Microsoft Internet Information Server IIS (*.htx, *.idc), texto enriquecido (*.rtf), Páginas de acceso a datos (*.htm; *.html) o datos XML (*.xml).

Como página de acceso a datos sólo se pueden presentar los formularios e informes.

Las páginas de acceso a datos sólo se pueden presentar en formato HTML.

Los formatos de IIS y ASP sólo están disponibles para tablas, consultas y formularios.

Los argumentos del método son

Tipo de objeto, Nombre del objeto, Formato de salida, Archivo de salida, Autoinicio, Archivo de plantilla, Codificación.

Como *Tipo de objeto* se utiliza una constante AcOutputObjectType

```
acOutputDataAccessPage (No permitido)
acOutputForm
acOutputFunction
acOutputModule
acOutputQuery
acOutputReport
acOutputServerView
acOutputStoredProcedure
acOutputTable
```

El siguiente código guarda la tabla Alumnos en el fichero **Alumnos.rtf** ubicado en la carpeta **C:\Datos** e inmediatamente después lo abre con Word para su posible edición.

```
DoCmd.OutputTo acOutputTable, _
                "Alumnos", _
                acFormatRTF, _
                "C:\Datos\Alumnos.rtf", _
                True
```

Consulte la ayuda de Access para completar esta información.

PrintOut

Imprime el objeto activo (hoja de datos, informe, formulario, página de acceso a datos o módulo).

Se puede especificar el *Rango* a imprimir, páginas *Desde* y *Hasta* las que imprimir, Calidad de la impresión, *Número de copias*, e *Intercalar copias*.

Como *Rango* se utiliza una constante del tipo AcPrintRange.

```
acPages
acPrintAll (Valor predeterminado)
acSelection
```

Como *Calidad* se utiliza una constante del tipo AcPrintQuality.

```
acDraft
acHigh (Valor predeterminado)
acLow
acMedium
```

El siguiente código imprimirá 4 copias de todo el objeto actual en alta calidad, intercalando las páginas.

```
DoCmd.PrintOut acPrintAll, , , _
                acHigh , _
                4, _
                True
```

Quit

Cierra todas las ventanas de Access y sale del programa.

Se puede introducir como parámetro una *Opción de salida*, constante del tipo AcQuitOption para que pregunte si tiene que guardar los cambios, los guarde directamente o no tenga en cuenta los cambios.

```
acQuitPrompt Muestra un cuadro de diálogo que le
pregunta si desea guardar cualquier objeto de base de
datos que se haya modificado pero no se haya guardado.
acQuitSaveAll (Predeterminado) Guarda todos los
objetos sin mostrar un cuadro de diálogo.
acQuitSaveNone Sale de Microsoft Access sin guardar
ningún objeto
```

La siguiente línea hace que se salga sin guardar los posibles cambios

```
DoCmd.Quit acQuitSaveNone
```

Rename

Sirve para cambiar el nombre a un objeto de la base de datos

Sus argumentos son: el *Nuevo nombre* del objeto, el *Tipo* de objeto y el *Nombre anterior* del mismo

Como *Tipo* de objeto se usa una constante del tipo `AcObjectType`

<code>acDataAccessPage</code>	<code>acDefault</code> (predeterminado)
<code>acDiagram</code>	<code>acForm</code>
<code>acFunction</code>	<code>acMacro</code>
<code>acModule</code>	<code>acQuery</code>
<code>acReport</code>	<code>acServerView</code>
<code>acStoredProcedure</code>	<code>acTable</code>

Esta línea de código cambia el nombre del formulario

"tmpCuentas" a "Cuentas".

```
DoCmd.Rename "Cuentas" , acForm, "tmpCuentas"
```

RepaintObject

Se usa para finalizar las actualizaciones pendientes de pantalla para un objeto de base de datos o para el objeto de base de datos activo, si no se especifica ninguno.

Incluyen los recálculos pendientes de los controles del objeto.

Como argumentos opcionales se pasan el *Tipo* de objeto del tipo `AcObjectType` (ver el método anterior) y el *Nombre* del mismo.

```
DoCmd.RepaintObject acForm, "Cuentas"
```

Requery

Actualiza los datos de un objeto, al volver a ejecutar la consulta origen del mismo.

Como parámetro utiliza el nombre del objeto. Se puede aplicar a formularios, subformularios, cuadros de lista y combinados, objetos ole y controles con funciones agregadas de dominio.

```
DoCmd.Requery "lstClientes"
```

Restore

Vuelve a mostrar, a su tamaño normal, una ventana de Windows que pudiera estar minimizada ó maximizada.

No utiliza parámetros, y se ejecuta en el objeto que lo llama.

```
DoCmd.Restore
```

RunCommand

El método `RunCommand` ejecuta un comando de menú integrado de barra de herramientas.

Reemplaza al método `DoMenuItem` de `DoCmd`.

Como parámetro se pasa el comando a ejecutar, representado por una de las constantes `AcCommand`.

El método `RunCommand` no se puede usar para ejecutar un comando en un menú o barra de herramientas personalizada.

Se usa con menús y barras de herramientas integradas.

La siguiente línea muestra el cuadro de diálogo:

```
[Acerca de Microsoft Office Access]
```

```
DoCmd.RunCommand acCmdAboutMicrosoftAccess
```

Hay una gran cantidad de constantes para llamar a cada uno de los posibles comandos de menú o de barra de herramientas de Microsoft Access.

Las constantes `AcCommand` son:

acCmdAboutMicrosoftAccess	acCmdChangeToLabel	acCmdDelete
acCmdAddInManager	acCmdChangeToListBox	acCmdDeleteGroup
acCmdAddToNewGroup	acCmdChangeToOptionButton	acCmdDeletePage
acCmdAddWatch	acCmdChangeToTextBox	acCmdDeleteQueryColumn
acCmdAdvancedFilterSort	acCmdChangeToToggleButton	acCmdDeleteRecord
acCmdAlignBottom	acCmdChartSortAscByTotal	acCmdDeleteRows
acCmdAlignCenter	acCmdChartSortDescByTotal	acCmdDeleteTab
acCmdAlignLeft	acCmdClearAll	acCmdDeleteTable
acCmdAlignmentAndSizing	acCmdClearAllBreakPoints	acCmdDeleteTableColumn
acCmdAlignMiddle	acCmdClearGrid	acCmdDeleteWatch
acCmdAlignRight	acCmdClearHyperlink	acCmdDemote
acCmdAlignToGrid	acCmdClearItemDefaults	acCmdDesignView
acCmdAlignTop	acCmdClose	acCmdDiagramAddRelatedTables
acCmdAlignToShortest	acCmdCloseWindow	acCmdDiagramAutoSizeSelectedTables
acCmdAlignToTallest	acCmdColumnWidth	acCmdDiagramDeleteRelationship
acCmdAnalyzePerformance	acCmdCompactDatabase	acCmdDiagramLayoutDiagram
acCmdAnalyzeTable	acCmdCompileAllModules	acCmdDiagramLayoutSelection
acCmdAnswerWizard	acCmdCompileAndSaveAllModules	acCmdDiagramModifyUserDefinedView
acCmdApplyDefault	acCmdCompileLoadedModules	acCmdDiagramNewLabel
acCmdApplyFilterSort	acCmdCompleteWord	acCmdDiagramNewTable
acCmdAppMaximize	acCmdConditionalFormatting	acCmdDiagramRecalculatePageBreaks
acCmdAppMinimize	acCmdConnection	acCmdDiagramShowRelationshipLabels
acCmdAppMove	acCmdControlWizardsToggle	acCmdDiagramViewPageBreaks
acCmdAppRestore	acCmdConvertDatabase	acCmdDocMaximize
acCmdAppSize	acCmdConvertMacrosToVisualBasic	acCmdDocMinimize
acCmdArrangelconsAuto	acCmdCopy	acCmdDocMove
acCmdArrangelconsByCreated	acCmdCopyDatabaseFile	acCmdDocRestore
acCmdArrangelconsByModified	acCmdCopyHyperlink	acCmdDocSize
acCmdArrangelconsByName	acCmdCreateMenuFromMacro	acCmdDocumenter
acCmdArrangelconsByType	acCmdCreateRelationship	acCmdDropSQLDatabase
acCmdAutoCorrect	acCmdCreateReplica	acCmdDuplicate
acCmdAutoDial	acCmdCreateShortcut	acCmdEditHyperlink
acCmdAutoFormat	acCmdCreateShortcutMenuFromMacro	acCmdEditingAllowed
acCmdBackgroundPicture	acCmdCreateToolbarFromMacro	acCmdEditRelationship
acCmdBackgroundSound	acCmdCut	acCmdEditTriggers
acCmdBackup	acCmdDataAccessPageAddToPage	acCmdEditWatch
acCmdBookmarksClearAll	acCmdDataAccessPageBrowse	acCmdEncryptDecryptDatabase
acCmdBookmarksNext	acCmdDataAccessPageDesignView	acCmdEnd
acCmdBookmarksPrevious	acCmdDataAccessPageFieldListRefresh	acCmdExit
acCmdBookmarksToggle	acCmdDatabaseProperties	acCmdExport
acCmdBringToFront	acCmdDatabaseSplitter	acCmdFavoritesAddTo
acCmdCallStack	acCmdDataEntry	acCmdFavoritesOpen
acCmdChangeToCheckBox	acCmdDataOutline	acCmdFieldList
acCmdChangeToComboBox	acCmdDatasheetView	acCmdFilterByForm
acCmdChangeToCommandButton	acCmdDateAndTime	acCmdFilterBySelection
acCmdChangeToImage	acCmdDebugWindow	acCmdFilterExcludingSelection

acCmdFind	acCmdJoinProperties	acCmdOpenStartPage
acCmdFindNext	acCmdLastPosition	acCmdOpenTable
acCmdFindNextWordUnderCursor	acCmdLayoutPreview	acCmdOpenURL
acCmdFindPrevious	acCmdLineUpIcons	acCmdOptions
acCmdFindPrevWordUnderCursor	acCmdLinkedTableManager	acCmdOutdent
acCmdFitToWindow	acCmdLinkTables	acCmdOutputToExcel
acCmdFont	acCmdListConstants	acCmdOutputToRTF
acCmdFormatCells	acCmdLoadFromQuery	acCmdOutputToText
acCmdFormHdrFtr	acCmdMacroConditions	acCmdPageHdrFtr
acCmdFormView	acCmdMacroNames	acCmdPageNumber
acCmdFreezeColumn	acCmdMakeMDEFile	acCmdPageProperties
acCmdGoBack	acCmdMaximumRecords	acCmdPageSetup
acCmdGoContinue	acCmdMicrosoftAccessHelpTopics	acCmdParameterInfo
acCmdGoForward	acCmdMicrosoftOnTheWeb	acCmdPartialReplicaWizard
acCmdGroupByTable	acCmdMicrosoftScriptEditor	acCmdPaste
acCmdGroupControls	acCmdMoreWindows	acCmdPasteAppend
acCmdHideColumns	acCmdNewDatabase	acCmdPasteAsHyperlink
acCmdHidePane	acCmdNewGroup	acCmdPasteSpecial
acCmdHideTable	acCmdNewObjectAutoForm	acCmdPivotAutoAverage
acCmdHorizontalSpacingDecrease	acCmdNewObjectAutoReport	acCmdPivotAutoCount
acCmdHorizontalSpacingIncrease	acCmdNewObjectClassModule	acCmdPivotAutoFilter
acCmdHorizontalSpacingMakeEqual	acCmdNewObjectDataAccessPage	acCmdPivotAutoMax
acCmdHyperlinkDisplayText	acCmdNewObjectDiagram	acCmdPivotAutoMin
acCmdImport	acCmdNewObjectForm	acCmdPivotAutoStdDev
acCmdIndent	acCmdNewObjectFunction	acCmdPivotAutoStdDevP
acCmdIndexes	acCmdNewObjectMacro	acCmdPivotAutoSum
acCmdInsertActiveXControl	acCmdNewObjectModule	acCmdPivotAutoVar
acCmdInsertChart	acCmdNewObjectQuery	acCmdPivotAutoVarP
acCmdInsertFile	acCmdNewObjectReport	acCmdPivotChartByRowByColumn
acCmdInsertFileIntoModule	acCmdNewObjectStoredProcedure	acCmdPivotChartDrillInto
acCmdInsertHyperlink	acCmdNewObjectTable	acCmdPivotChartDrillOut
acCmdInsertLookupColumn	acCmdNewObjectView	acCmdPivotChartMultiplePlots
acCmdInsertLookupField	acCmdObjBrwFindWholeWordOnly	acCmdPivotChartMultiplePlotsUnifiedScale
acCmdInsertMovieFromFile	acCmdObjBrwGroupMembers	acCmdPivotChartShowLegend
acCmdInsertObject	acCmdObjBrwHelp	acCmdPivotChartType
acCmdInsertPage	acCmdObjBrwShowHiddenMembers	acCmdPivotChartUndo
acCmdInsertPicture	acCmdObjBrwViewDefinition	acCmdPivotChartView
acCmdInsertPivotTable	acCmdObjectBrowser	acCmdPivotCollapse
acCmdInsertProcedure	acCmdOfficeClipboard	acCmdPivotDelete
acCmdInsertQueryColumn	acCmdOLEDELinks	acCmdPivotDropAreas
acCmdInsertRows	acCmdOLEObjectConvert	acCmdPivotExpand
acCmdInsertSpreadsheet	acCmdOLEObjectDefaultVerb	acCmdPivotRefresh
acCmdInsertSubdatasheet	acCmdOpenDatabase	acCmdPivotShowAll
acCmdInsertTableColumn	acCmdOpenHyperlink	acCmdPivotShowBottom1
acCmdInsertUnboundSection	acCmdOpenNewHyperlink	acCmdPivotShowBottom10
acCmdInvokeBuilder	acCmdOpenSearchPage	acCmdPivotShowBottom10Percent

acCmdPivotShowBottom1Percent	acCmdPreviewOnePage	acCmdRepairDatabase
acCmdPivotShowBottom2	acCmdPreviewTwelvePages	acCmdReplace
acCmdPivotShowBottom25	acCmdPreviewTwoPages	acCmdReportHdrFtr
acCmdPivotShowBottom25Percent	acCmdPrimaryKey	acCmdReset
acCmdPivotShowBottom2Percent	acCmdPrint	acCmdResolveConflicts
acCmdPivotShowBottom5	acCmdPrintPreview	acCmdRestore
acCmdPivotShowBottom5Percent	acCmdPrintRelationships	acCmdRowHeight
acCmdPivotShowBottomOther	acCmdProcedureDefinition	acCmdRun
acCmdPivotShowTop1	acCmdPromote	acCmdRunMacro
acCmdPivotShowTop10	acCmdProperties	acCmdRunOpenMacro
acCmdPivotShowTop10Percent	acCmdPublish	acCmdSave
acCmdPivotShowTop1Percent	acCmdPublishDefaults	acCmdSaveAllModules
acCmdPivotShowTop2	acCmdQueryAddToOutput	acCmdSaveAllRecords
acCmdPivotShowTop25	acCmdQueryGroupBy	acCmdSaveAs
acCmdPivotShowTop25Percent	acCmdQueryParameters	acCmdSaveAsASP
acCmdPivotShowTop2Percent	acCmdQueryTotals	acCmdSaveAsDataAccessPage
acCmdPivotShowTop5	acCmdQueryTypeAppend	acCmdSaveAsHTML
acCmdPivotShowTop5Percent	acCmdQueryTypeCrosstab	acCmdSaveAsIDC
acCmdPivotShowTopOther	acCmdQueryTypeDelete	acCmdSaveAsQuery
acCmdPivotTableClearCustomOrdering	acCmdQueryTypeMakeTable	acCmdSaveAsReport
acCmdPivotTableCreateCalcField	acCmdQueryTypeSelect	acCmdSaveLayout
acCmdPivotTableCreateCalcTotal	acCmdQueryTypeSQLDataDefinition	acCmdSaveModuleAsText
acCmdPivotTableDemote	acCmdQueryTypeSQLPassThrough	acCmdSaveRecord
acCmdPivotTableExpandIndicators	acCmdQueryTypeSQLUnion	acCmdSelectAll
acCmdPivotTableExportToExcel	acCmdQueryTypeUpdate	acCmdSelectAllRecords
acCmdPivotTableFilterBySelection	acCmdQuickInfo	acCmdSelectDataAccessPage
acCmdPivotTableGroupItems	acCmdQuickPrint	acCmdSelectForm
acCmdPivotTableHideDetails	acCmdQuickWatch	acCmdSelectRecord
acCmdPivotTableMoveToColumnArea	acCmdRecordsGoToFirst	acCmdSelectReport
acCmdPivotTableMoveToDetailArea	acCmdRecordsGoToLast	acCmdSend
acCmdPivotTableMoveToFilterArea	acCmdRecordsGoToNew	acCmdSendToBack
acCmdPivotTableMoveToRowArea	acCmdRecordsGoToNext	acCmdServerFilterByForm
acCmdPivotTablePercentColumnTotal	acCmdRecordsGoToPrevious	acCmdServerProperties
acCmdPivotTablePercentGrandTotal	acCmdRecoverDesignMaster	acCmdSetControlDefaults
acCmdPivotTablePercentParentColumnItem	acCmdRedo	acCmdSetDatabasePassword
acCmdPivotTablePercentParentRowItem	acCmdReferences	acCmdSetNextStatement
acCmdPivotTablePercentRowTotal	acCmdRefresh	acCmdShowAllRelationships
acCmdPivotTablePromote	acCmdRefreshPage	acCmdShowDirectRelationships
acCmdPivotTableRemove	acCmdRegisterActiveXControls	acCmdShowEnvelope
acCmdPivotTableShowAsNormal	acCmdRelationships	acCmdShowMembers
acCmdPivotTableShowDetails	acCmdRemove	acCmdShowNextStatement
acCmdPivotTableSubtotal	acCmdRemoveFilterSort	acCmdShowOnlyWebToolbar
acCmdPivotTableUngroupItems	acCmdRemoveTable	acCmdShowTable
acCmdPivotTableView	acCmdRename	acCmdSingleStep
acCmdPreviewEightPages	acCmdRenameColumn	acCmdSizeToFit
acCmdPreviewFourPages	acCmdRenameGroup	acCmdSizeToFitForm

acCmdSizeToGrid	acCmdToggleBreakPoint	acCmdViewShowPaneGrid
acCmdSizeToNarrowest	acCmdToggleFilter	acCmdViewShowPaneSQL
acCmdSizeToWidest	acCmdToolbarControlProperties	acCmdViewSmallIcons
acCmdSnapToGrid	acCmdToolbarsCustomize	acCmdViewStoredProcedures
acCmdSortAscending	acCmdTransferSQLDatabase	acCmdViewTableColumnNames
acCmdSortDescending	acCmdTransparentBackground	acCmdViewTableColumnProperties
acCmdSortingAndGrouping	acCmdTransparentBorder	acCmdViewTableKeys
acCmdSpeech	acCmdUndo	acCmdViewTableNameOnly
acCmdSpelling	acCmdUndoAllRecords	acCmdViewTables
acCmdSQLView	acCmdUnfreezeAllColumns	acCmdViewTableUserView
acCmdStartupProperties	acCmdUngroupControls	acCmdViewToolbox
acCmdStepInto	acCmdUnhideColumns	acCmdViewVerifySQL
acCmdStepOut	acCmdUpsizingWizard	acCmdViewViews
acCmdStepOver	acCmdUserAndGroupAccounts	acCmdVisualBasicEditor
acCmdStepToCursor	acCmdUserAndGroupPermissions	acCmdWebPagePreview
acCmdStopLoadingPage	acCmdUserLevelSecurityWizard	acCmdWebPageProperties
acCmdSubdatasheetCollapseAll	acCmdVerticalSpacingDecrease	acCmdWebTheme
acCmdSubdatasheetExpandAll	acCmdVerticalSpacingIncrease	acCmdWindowArrangelcons
acCmdSubdatasheetRemove	acCmdVerticalSpacingMakeEqual	acCmdWindowCascade
acCmdSubformDatasheet	acCmdViewCode	acCmdWindowHide
acCmdSubformDatasheetView	acCmdViewDataAccessPages	acCmdWindowSplit
acCmdSubformFormView	acCmdViewDetails	acCmdWindowUnhide
acCmdSubformInNewWindow	acCmdViewDiagrams	acCmdWordMailMerge
acCmdSubformPivotChartView	acCmdViewFieldList	acCmdWorkgroupAdministrator
acCmdSubformPivotTableView	acCmdViewForms	acCmdZoom10
acCmdSwitchboardManager	acCmdViewFunctions	acCmdZoom100
acCmdSynchronizeNow	acCmdViewGrid	acCmdZoom1000
acCmdTabControlPageOrder	acCmdViewLargelcons	acCmdZoom150
acCmdTableAddTable	acCmdViewList	acCmdZoom200
acCmdTableCustomView	acCmdViewMacros	acCmdZoom25
acCmdTableNames	acCmdViewModules	acCmdZoom50
acCmdTabOrder	acCmdViewQueries	acCmdZoom500
acCmdTestValidationRules	acCmdViewReports	acCmdZoom75
acCmdTileHorizontally	acCmdViewRuler	acCmdZoomBox
acCmdTileVertically	acCmdViewShowPaneDiagram	acCmdZoomSelection

RunMacro

Sirve para ejecutar una Macro grabada.

La Macro puede estar guardada de forma individual, o estar integrada dentro de un grupo de macros.

Como parámetros se pasa el *Nombre de la macro* y opcionalmente el *Número de veces* que se va a repetir la macro y una *Expresión numérica* que se evalúa cada vez que se ejecuta la macro. Si esta expresión diera False (0), se detendría la ejecución de la macro.

La siguiente línea de código ejecuta 3 veces la macro **Mensaje** guardada en el grupo de macros **MacrosInicio**. Esta macro es la que habíamos creado en un punto anterior.

```
DoCmd.RunMacro "MacrosInicio.Mensaje", 3
```

RunSQL

La acción RunSQL (EjecutarSQL) ejecuta una consulta de acción utilizando una instrucción SQL correspondiente. También puede utilizarse para ejecutar una consulta de definición de datos.

Como consultas de acción se puede anexar, eliminar y actualizar registros e incluso guardar un conjunto de datos resultado de una consulta dentro de una nueva tabla.

Como consulta de definición de datos se pueden usar las siguientes instrucciones SQL

Para crear una tabla	Create Table
Para modificar una tabla	Alter Table
Para borrar una tabla	Drop Table
Para crear un índice	Create Index
Para borrar un índice	Drop Index

El siguiente ejemplo crea la tabla Alumnos, con diferentes tipos de campo, utilizando el método RunSQL.

```
Public Sub CrearTablaAlumnos()
    Dim strSQL As String

    strSQL = "Create Table Alumnos " _
        & "(IDAlumno Integer, " _
        & "Nombre Varchar(20), " _
        & "Apellido2 Varchar(20), " _
        & "Apellido1 Varchar(20), " _
        & "FechaNacimiento Date, " _
        & "Sexo Varchar(1), " _
        & "Activo bit)"

    DoCmd.RunSQL strSQL
End Sub
```

Save

Guarda el objeto especificado o, si no se indica, el objeto activo.

Como parámetros opcionales se pueden pasar el *Tipo* de objeto y el *Nombre* del mismo.

El parámetro *Tipo de objeto* puede ser cualquiera de las constantes AcObjectType, vistas en **RenameObject**.

```
DoCmd.Save acForm, "AlumnosActivos"
```

SelectObject

Seleccionar el objeto especificado de la base de datos.

Como parámetros obligatorios hay que usar el Tipo de objeto (constante del tipo AcObjectType descrita en **RenameObject**) y el *Nombre* del mismo. Opcionalmente se puede indicar si se *Selecciona* el objeto en la ventana Base de datos, mediante **True** ó **False**.

El siguiente ejemplo selecciona el formulario Alumnos en la ventana Base de datos:

```
DoCmd.SelectObject acForm, "Alumnos", True
```

SendObject

Se utiliza para incluir una hoja de datos, formulario, informe,

módulo o página de acceso a datos de Access en un mensaje de correo electrónico. En las aplicaciones de correo electrónico que admitan la interfaz estándar Microsoft MAPI, se puede incluir objetos con los formatos de Microsoft Excel (*.xls), Texto MS-DOS (*.txt), Texto enriquecido (*.rtf), o HTML (*.html).

Como parámetros se utilizan

Tipo de objeto: Tabla, Consulta, Formulario, Informe, Módulo, Página de acceso a datos, Vista de servidor o Procedimientos almacenados. Parámetros adicionales son *Nombre del objeto*, *Formato de salida*, *Enviar a*, *Enviar copia a*, *Enviar copia oculta a*, *Asunto del mensaje*, *Texto del mensaje*, *Abrir (sí ó no) el programa de correo* para editar el mensaje antes de enviarlo, *Plantilla* para un archivo HTML.

El tipo de objeto a enviar viene dado por una constante del tipo `AcSendObjectType`

Puede ser una de las siguientes

```
AcSendObjectType.  
acSendDataAccessPage  
acSendForm  
acSendModule  
acSendNoObject (valor predeterminado)  
acSendQuery  
acSendReport  
acSendTable
```

El tipo de objeto a enviar viene dado por una de las siguientes constantes

```
acFormatXLS  
acFormatTXT  
acFormatRTF  
acFormatHTML
```

El siguiente código envía los datos de la tabla `DatosNumericos` como datos adjuntos en formato Excel, al correo correspondiente a Eduardo Olaz, una copia a Javier Itóiz y una copia oculta a Edurne Goizueta.

El mensaje tiene como encabezado `Datos Test maquinaria` y como texto `"Datos resultado de las pruebas"`.

El mensaje se envía sin necesidad de abrir el programa de correo.

```
DoCmd.SendObject acSendTable, _  
    "DatosNumericos", _  
    acFormatXLS, _  
    "Eduardo Olaz", _  
    "Javier Itoiz", _  
    "Edurne Goizueta", _  
    "Datos Test maquinaria", _  
    "Datos resultado de las pruebas", _  
    False
```

-
- SetMenuItem** Establece el estado de los elementos (habilitado o deshabilitado, activado o desactivado) de las barras de menú y barras de menú generales personalizadas creadas con macros de barras de menús.
- Los parámetros son *Índice de menú*, *Índice de comando*, *Índice de sub-comando* e *Indicador*.
- El *Indicador* es una constante del tipo `AcMenuType` y toma alguno de los siguientes valores
- `acMenuCheck`
 - `acMenuGray`
 - `acMenuUncheck`
 - `acMenuUngray` (valor predeterminado)
- Los índices empiezan con el valor 0.
- Esta línea de código desactiva el tercer comando del segundo menú de la barra de menús personalizada para la ventana activa:
- ```
DoCmd.SetMenuItem 1, 2, , acMenuGray
```
- SetWarnings** Activa o desactiva los mensajes de advertencia del sistema.
- Como parámetro *Activar advertencias* se pasa `True` ó `False`.
- Este código desactiva los mensajes de advertencia
- ```
DoCmd.SetMenuItem False
```
- ShowAllRecords** Quita los filtros aplicados a una tabla, conjunto de resultados de una consulta o del formulario activo y muestra todos los registros.
- ```
DoCmd.ShowAllRecords
```
- ShowToolbar** Muestra u oculta una barra de herramientas integrada o una barra de herramientas personalizada.
- Sus parámetros son el *Nombre de la barra* de herramientas, y una constante del tipo `AcShowToolbar` que indica cómo se quiere *Mostrar la barra* de herramientas.
- Los valores de `AcShowToolbar` pueden ser
- `acToolbarNo`
  - `acToolbarWhereApprop`
  - `acToolbarYes` (valor predeterminado)
- La siguiente línea de código muestra la barra de herramientas personalizada denominada **MiBarra** en todas las ventanas Microsoft Access que se vuelvan activas:
- ```
DoCmd.ShowToolbar "MiBarra" acToolbarYes
```
- Para más información vea, en la ayuda de Access la información sobre la Acción **ShowToolbar** (`MostrarBarraDeHerramientas`)
- TransferDatabase** Importa o exporta datos entre la base de datos de Microsoft Access actual o el proyecto de Microsoft Access (.adp) actual y otra base de datos.
- También puede vincular una tabla a la base de datos de Access actual desde otra base de datos.

Como parámetros podemos usar

Tipo de transferencia una constante del tipo `AcDataTransferType` que puede tomar los valores

```
acExport
acImport (valor predeterminado)
acLink
```

El segundo parámetro indica el *Tipo de base de datos*. El tipo viene indicado por una expresión de cadena que puede tomar los siguientes valores

```
Microsoft Access (predeterminada)
Jet 2.x
Jet 3.x
dBase III
dBase IV
dBase 5.0
Paradox 3.x
Paradox 4.x
Paradox 5.x
Paradox 7.x
ODBC Database
WSS
```

El siguiente parámetro es el *Nombre de la base de datos*.

A continuación se indica el *Tipo de objeto* que se va a importar ó exportar. Éste se define mediante una constante del tipo puede ser cualquiera de las constantes **AcObjectType**, vistas en **RenameObject**.

El parámetro *Origen* define el nombre de la tabla, consulta de selección u objeto de Access que se desea importar, exportar o vincular. Si la tabla es un fichero, como en el caso de las tablas DBF, se debe indicar la extensión de ésta.

Destino define el nombre que tendrá el objeto, en la base de datos destino, una vez importado, exportado o vinculado.

El parámetro *Estructura solamente* especifica si se va a importar o exportar la estructura de una tabla sin los datos.

El último parámetro *StoreLogin* especifica si, para una tabla vinculada desde la base de datos, se almacenan en la cadena de conexión la identificación de inicio de sesión y la contraseña de una base de datos ODBC.

Los siguientes ejemplos están basados en los que aparecen en la ayuda de Access, correspondientes al Método `TransferDatabase`

Este siguiente ejemplo importa el informe `rptVentasDeAbril` desde la base de datos Access `Ventas.mdb` al informe `VentasDeAbril` en la base de datos activa:

```
DoCmd.TransferDatabase acImport, _
    "Microsoft Access", _
    "C:\Mis Documentos\ Ventas.mdb", _
    acReport, _
    "rptVentasDeAbril", _
    "VentasDeAbril"
```

El siguiente ejemplo vincula la tabla de la base de datos ODBC Autores a la base de datos activa:

```
DoCmd.TransferDatabase acLink, _
"ODBC Database", _
"ODBC;DSN=DataSource1;" _
& "UID=User2;PWD=www;" _
& "LANGUAGE=us_english;" _
& "DATABASE=pubs", _
acTable, _
" Autores", _
"dbo Autores"
```

En el siguiente ejemplo se exporta el contenido de la tabla Clientes a una lista nueva denominada Lista de clientes en el sitio Windows SharePoint Services "http://example/WSSSite".

```
DoCmd.TransferDatabase transfertype:=acExport, _
databasetype:="WSS", _
databasename:="http://example/WSSSite", _
objecttype:=acTable, _
Source:="Clientes", _
Destination:=" Lista de clientes", _
structureonly:=False
```

TransferSpreadsheet

Importa o exporta datos entre la base de datos Access o el proyecto de Access (.adp) actual y un archivo de hoja de cálculo.

Se puede vincular los datos de una hoja de cálculo de Microsoft Excel a la base de datos de Access actual.

Puede establecerse vínculos, de sólo lectura, a los datos de un archivo de hoja de cálculo Lotus 1-2-3.

Los parámetros son.

Tipo de transferencia, una constante del tipo AcDataTransferType vista en el método **TransferDatabase**.

Tipo de hoja de cálculo del tipo AcSpreadSheetType

Puede tomar los valores

```
acSpreadsheetTypeExcel3
acSpreadsheetTypeExcel4
acSpreadsheetTypeExcel5
acSpreadsheetTypeExcel7
acSpreadsheetTypeExcel8 (valor predeterminado)
acSpreadsheetTypeExcel9 (valor predeterminado)
acSpreadsheetTypeLotusWJ2 - sólo versión Japonesa
acSpreadsheetTypeLotusWK1
acSpreadsheetTypeLotusWK3
acSpreadsheetTypeLotusWK4
```

Nombre de la tabla a la que se van a importar, de la que se van a exportar o con la que se van a vincular datos de hoja de cálculo.

Nombre de archivo desde el que se va a importar, al que se va a exportar o con el que se va a establecer un vínculo

Contiene nombres de campo. Especifica si la primera fila de la hoja de cálculo contiene los nombres de los campos.

Rango. Rango de celdas que se van a importar o vincular.

El siguiente ejemplo importa los datos ubicados en el rango (A1:G12), de la hoja de cálculo Lotus **DatosEmpleados.wk3** a la tabla **Empleados**.

Los nombres de los campos están contenidos en la primera fila.

```
DoCmd.TransferSpreadsheet acImport, _
    acSpreadsheetTypeLotusWK3, _
    "Empleados", _
    "C:\Lotus\DatosEmpleados.wk3", _
    True, _
    "A1:G12"
```

TransferSQLDatabase Transfiere toda la base de datos del tipo Microsoft SQL Server especificada a otra base de datos SQL Server.

La sintaxis completa de uso es la siguiente

```
DoCmd.TransferSQLDatabase ( _
    Servidor, _
    BaseDeDatos, _
    UsarConexiónDeConfianza, _
    InicioDeSesión, _
    Contraseña, _
    TransferirCopiaDatos)
```

Como parámetros se incluyen

Servidor (server): nombre del servidor al que se va a transferir la base de datos.

BaseDeDatos (DataBase): nombre de la nueva base de datos en el servidor especificado.

UsarConexiónDeConfianza (UseTrustedConnection): indicando con True que la conexión activa utiliza un inicio de sesión con privilegios de administrador del sistema. Si tomara otro valor, se deberán especificar los Parámetros *InicioDeSesión* y *Contraseña*.

InicioDeSesión (Login): Nombre de un inicio de sesión del servidor de destino con privilegios de administrador de sistema

Contraseña (Password): La contraseña para el inicio de sesión especificado en *InicioDeSesión*.

TransferirCopiaDatos (TransferCopyData): **True** si todos los datos de la base de datos se transfieren a la base de datos de destino. En caso contrario sólo se transfiere el esquema de la base de datos.

Este código transfiere la base de datos SQL Server actual a la nueva base de datos SQL Server denominada **CopiaGestion** en el servidor **Principal**. (El usuario debe disponer de privilegios de administrador del sistema en **Principal**.)

Se copian los datos y el esquema de la base de datos.

```
DoCmd.TransferCompleteSQLDatabase _
    Server:="Principal", _
    Database:="CopiaGestion", _
    UseTrustedConnection:=True, _
    TransferCopyData:=True
```

TransferText Importar o exporta texto entre la base de datos Access o el proyecto de Access (.adp) actual y un archivo de texto.

Puede vincular los datos de un archivo de texto a la base de datos de Access actual.

También puede importar, exportar y establecer vínculos con una tabla o lista de un archivo HTML.

La sintaxis completa de uso es la siguiente

```
DoCmd. TransferText ( _
    TipoDeTransferencia, _
    NombreDeEspecificación, _
    NombreDeLaTabla, _
    NombreDelFichero, _
    ContieneNombresDeCampo, _
    NombreDeLaTablaHTML, _
    PáginaDeCódigos)
```

Los parámetros son los siguientes

TipoDeTransferencia (**TransferType**): Importar, exportar datos o establecer un vínculo con archivos de texto con datos delimitados o de ancho fijo y archivos HTML. También se puede exportar datos a un archivo de datos de combinación de correspondencia de Microsoft Word. Su valor es una constante del tipo **AcTextTransferType** que puede tomar los siguientes valores

```
acExportDelim
acExportFixed
acExportHTML
acExportMerge
acImportDelim (valor predeterminado)
acImportFixed
acImportHTML
acLinkDelim
acLinkFixed
acLinkHTML
```

NombreDeEspecificación (**SpecificationName**): nombre de una especificación de importación o exportación creada y guardada en la base de datos activa. Con un archivo de texto de ancho fijo, se debe especificar un argumento o utilizar un archivo `schema.ini`, guardado en la misma carpeta que el archivo de texto importado, vinculado o exportado.

NombreDeLaTabla (**TableName**): Nombre de la tabla Access a la que desea importar, de la que desea exportar o a la que desea vincular datos de texto, o nombre de la consulta de Access cuyos resultados se desean exportar a un archivo de texto.

NombreDelFichero (**FileName**): Nombre completo, incluso ruta de acceso, del archivo de texto del que desea importar, al que desea exportar o con el que desea crear un vínculo.

ContieneNombresDeCampo (**HasFieldNames**): Si el valor es **True** o **-1**, se indica que la primera fila del archivo de texto se va a usar como nombres de campos al importar, exportar o vincular.

NombreDeLaTablaHTML (**HTMLTableName**): Nombre de la tabla o lista del archivo HTML que desea importar o vincular. Este argumento se omite salvo que el argumento *TipoDeTransferencia* se establezca en **acImportHTML** o **acLinkHTML**.

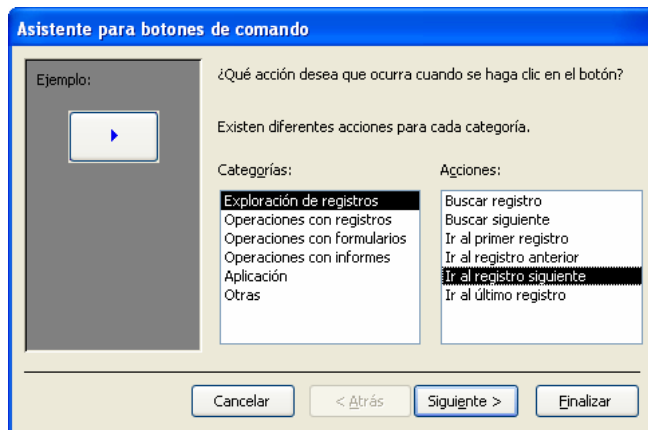
PáginaDeCódigos (CodePage): Valor Long que indica el conjunto de caracteres de la página de códigos

Uso de DoCmd en los Asistentes para controles

Los Asistentes para controles de Access, por ejemplo en los formularios, hacen un amplio uso del objeto DoCmd.



Si por ejemplo se añade un botón a un formulario, teniendo activado el asistente para controles, (botón señalado con la flecha roja) veremos que Access nos muestra un formulario en el que nos pide información acerca de la acción que queremos que ocurra cuando presionemos en el botón.



En nuestro caso vamos a crear un botón que haga que vayamos al registro siguiente respecto al registro actual.

Primero debemos decidir qué categoría vamos a utilizar, por lo que seleccionaremos en la izquierda la [Exploración de registros].

En la ventana de la derecha (acciones) seleccionaremos [Ir al registro siguiente].

Presionamos dos veces las sucesivas teclas [Siguiete] y ponemos como

nombre del botón cmdRegistroSiguiete, y presionamos la tecla [Finalizar].

Si hemos seguido estos pasos, nos mostrará en el formulario un botón con una flecha a la derecha.

Si abrimos el módulo de clase del formulario veremos que ha escrito el siguiente código

```
Private Sub cmdRegistroSiguiete_Click()
On Error GoTo Err_cmdRegistroSiguiete_Click

DoCmd.GoToRecord , , acNext

Exit_cmdRegistroSiguiete_Click:
Exit Sub

Err_cmdRegistroSiguiete_Click:
MsgBox Err.Description
Resume Exit_cmdRegistroSiguiete_Click

End Sub
```

Vemos que nos ha generado un procedimiento de evento que define qué debe pasar cuando se presione el botón [cmdRegistroSiguiete].

En concreto vemos que usa el objeto DoCmd con el método GotoRecord, y el parámetro acNext.

Esto hace que intente posicionarse en el registro siguiente. Incluso define un sistema para controlar posibles errores que se puedan generar.

Sugiero al lector que intente realizar un formulario, conectado a una tabla o consulta y que contenga botones para efectuar las siguientes operaciones:

1. Ir al primer registro
2. Ir al registro anterior
3. Ir al registro siguiente
4. Ir al último registro
5. Agregar nuevo registro
6. Guardar registro
7. Cerrar Formulario

Una vez realizadas estas operaciones analizar el código que se genera.

Así mismo sugiero que efectúe pruebas con diferentes métodos de DoCmd.