

UNIDAD 14. LAS MACROS

14.1 Introducción

Las macros son un método sencillo para llevar a cabo una o varias tareas básicas como abrir y cerrar formularios, mostrar u ocultar barras de herramientas, ejecutar informes, etc.

También sirven para crear métodos abreviados de teclado y para que se ejecuten tareas automáticamente cada vez que se inicie la base de datos.

Si guardamos la Macro con el nombre de AutoExec, cada vez que se inicie la base de datos, se ejecutará automáticamente. Esto es debido a que Access al arrancar busca una macro con ese nombre, si la encuentra será el primer objeto que se ejecute antes de lanzar cualquier otro.

Esta opción es muy socorrida a la hora de efectuar comprobaciones o lanzar procesos antes de que el usuario empiece a trabajar con la base de datos.

La configuración por defecto de Access, nos impedirá ejecutar ciertas acciones de macro si la base de datos no se encuentra en una ubicación de confianza, para evitar acciones malintencionadas.

Para ejecutar correctamente las macros de bases de datos que consideremos fiables, podemos añadir la ubicación en el Centro de confianza, tal como vimos en la Unidad 10 al realizar consultas de acción.

14.2 Crear una macro

Para definir una macro, indicaremos una acción o conjunto de acciones que automatizarán un proceso. Cuando ejecutemos una Macro, el proceso se realizará automáticamente sin necesidad, en principio, de interacción por nuestra parte.

Por ejemplo, podríamos definir una macro que abra un formulario cuando el usuario haga clic en un botón, o una macro que abra una consulta para subir un diez por cien el precio de nuestros productos.

Crear una macro es relativamente fácil, sólo tienes que hacer clic en el botón Macro  de la pestaña Crear y se abrirá la ventana con la nueva macro, así como sus correspondientes Herramientas de macros, englobadas en la pestaña Diseño.

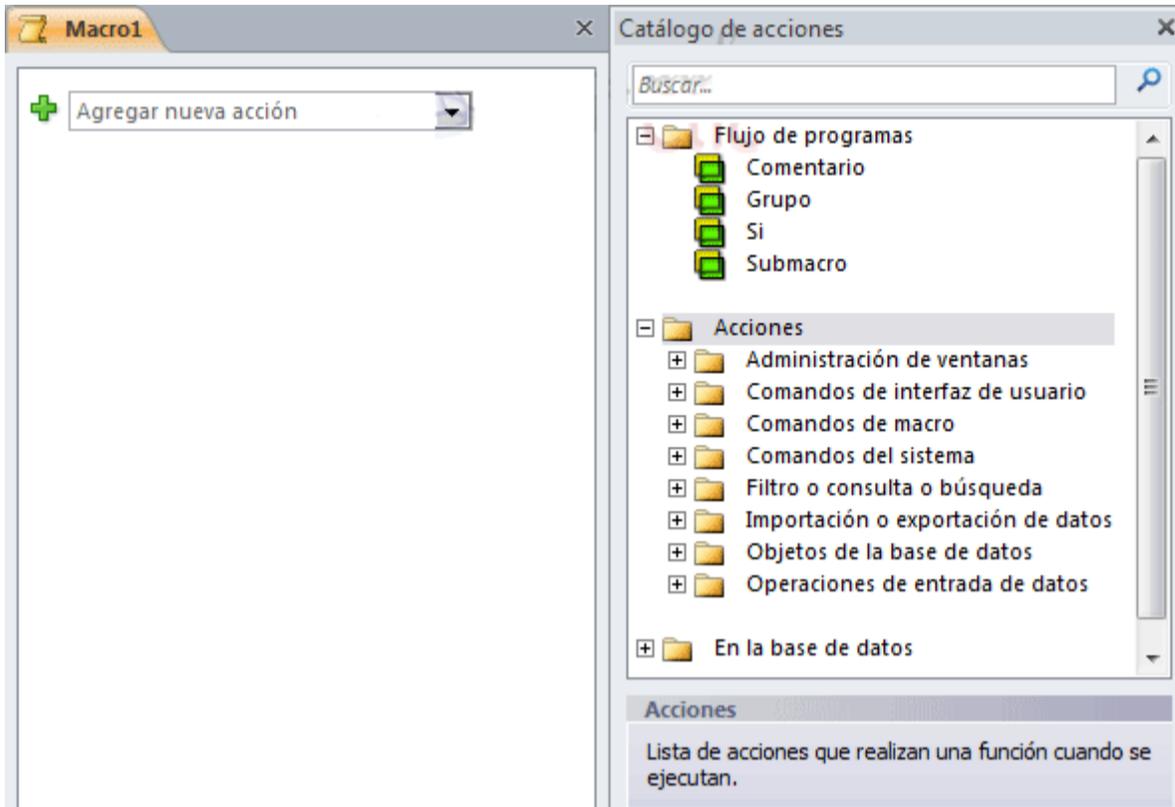


Si habías utilizado versiones antiguas de Access, notarás que en la actual versión se ha modificado un tanto la forma de trabajar con las macros.

Como podrás observar, la ventana principal consta de una lista desplegable que te permite elegir la Acción para la macro. En el panel de la izquierda encontrarás estas mismas acciones agrupadas por categorías según su tipo y con un útil buscador en la zona superior, de forma



que te sea más sencillo localizar la que deseas aplicar. Si no te aparece el panel Catálogo de acciones puedes sacarlo con el botón Catálogo de acciones  de la pestaña Diseño.



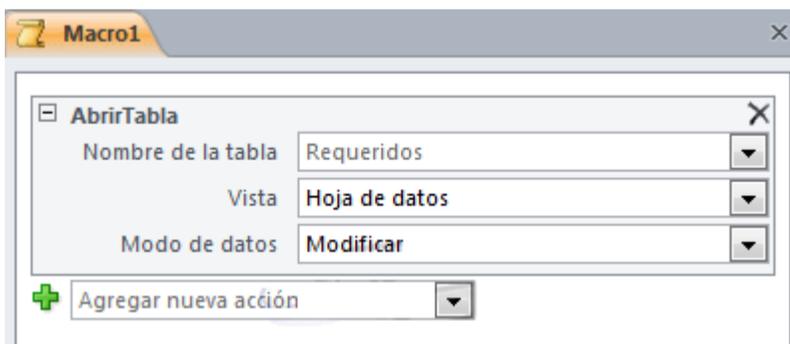
Podemos añadir tantas acciones como queramos, ya que al elegir una opción en el desplegable aparecerá otro inmediatamente debajo del primero, y así sucesivamente. Simplemente deberemos tener presente que se ejecutarán en el orden en que se encuentren. Es una cuestión de lógica, se ejecuta de forma lineal, de forma que no tendría sentido tratar de Cerrar ventana si aún no la hemos abierto, por ejemplo.

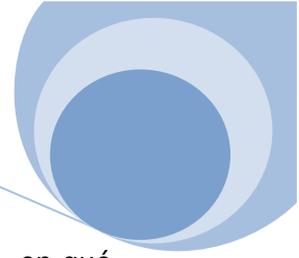
Para cambiar el orden en el que se encuentren las acciones puedes arrastrarlas con el ratón hasta la posición correcta o bien utilizar los botones de la acción, que aparecerán al pasar el cursor sobre ella. Con ellos podrás subir o bajar un nivel la acción por cada pulsación o eliminarla.



Obviamente estos botones sólo están disponibles si hay más de una acción. La última sólo podrá ascender, la primera sólo podrá descender y si sólo hay una acción únicamente dispondrá del icono eliminar situado a la derecha.

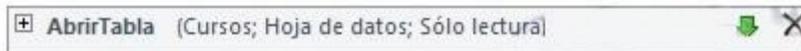
En función de la acción que seleccionemos aparecerá un panel con un aspecto u otro, en el que podremos especificar los detalles necesarios.





Por ejemplo, para la acción AbrirTabla, necesitaríamos saber el Nombre de la tabla, en qué Vista queremos que se muestre y si los datos se podrán modificar o no una vez abierta (Modo de datos). No siempre será obligatorio rellenar todos los campos, algunos son requeridos, el resto tendrá un valor por defecto (como en este caso Vista: Hoja de datos) o que simplemente sean opcionales.

Cuando tengas muchas acciones en una macro, es posible que te interese ocultar los detalles para ver la lista de acciones una bajo otra. En ese caso, podrás expandir  y contraer  la información desde el botón de la esquina superior izquierda de la acción. Cuando se ocultan los detalles, la información relevante se muestra toda en una fila, como puedes observar en la siguiente imagen.



Otra forma de contraer y expandir es desde su correspondiente grupo en la pestaña Diseño.



Cuando la Macro está terminada, puede guardarse , ejecutarse  y cerrarse. Más tarde podremos llamarla desde un control Botón, o ejecutarla directamente desde la ventana de la base de datos haciendo clic en Ejecutar o bien haciendo doble clic directamente sobre ella.

14.3 Ejecutar una macro

Aunque aún no hayamos aprendido mucho sobre ellas, es importante que tengamos claro para qué sirven exactamente las macros y cuándo se ejecutan.

Desde luego, siempre podemos abrir el diseño de la macro y pulsar el botón Ejecutar  en la cinta de opciones, para ejecutarla de forma manual. También podríamos hacer doble clic sobre ella en el Panel de navegación. Pero estas no son las prácticas más utilizadas.

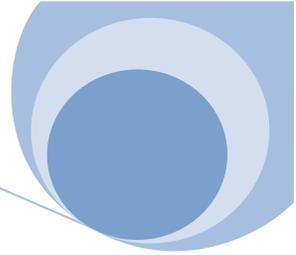
La mayoría de veces, las macros serán acciones que se ejecutan en segundo plano, sin la plena conciencia del usuario de la base de datos. El usuario que se encarga de actualizar el inventario o dar de alta pacientes no tiene por qué saber cómo se llaman las tablas y qué acciones concretas ejecuta cada macro. Normalmente, el usuario en realidad trabaja con formularios amigables, con botones y otros controles, que utiliza de forma intuitiva.

Somos nosotros, quienes creamos la base de datos, los encargados de asignar a cada control la macro conveniente. Por lo tanto, lo que debemos hacer es asignar una macro que programe qué acción se ejecutará al interactuar con un determinado control u objeto. Y para ello trabajaremos con sus Eventos.

Un evento es una acción que el usuario realiza, normalmente de forma activa. Por ejemplo hacer clic o doble clic sobre un botón, cambiar de un registro a otro en un formulario, modificar un determinado campo de un registro, cerrar la base de datos, etc.

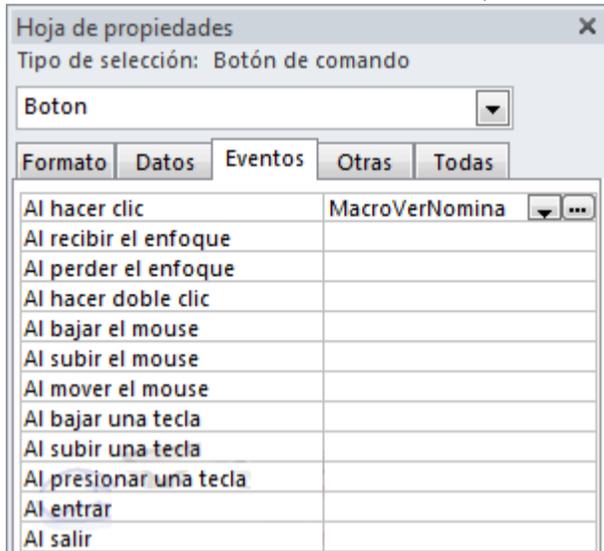
Deberemos reflexionar sobre en qué momento nos interesa que se ejecute la macro, para aprender a elegir qué evento o qué control la desencadenará.





Para asociar la macro a un control:

En la vista diseño de formulario, seleccionamos el control o el propio formulario.



Luego abrimos su Hoja de propiedades, si no está ya abierta, y nos situamos en la pestaña Eventos.

Entre los posibles eventos, elegimos el que nos conviene según en qué momento queremos que se ejecute la macro. Al hacer clic en él aparecerán dos botones:

- El primero nos permitirá desplegar la lista de macros que tengamos en la base de datos. Ahí es donde deberemos indicar qué macro ejecutar.
- El segundo botón nos permite elegir el tipo de generador entre los generadores de macros, expresiones y código. No vamos a entrar en detalle en él.

En el ejemplo de la imagen hemos asignado al evento Al hacer clic de un Botón de comando una macro que se encarga de mostrar la nómina del empleado actual. De forma que si el usuario está viendo los registros de empleados en un formulario y pulsa el botón, se abrirá una ventana con el formulario que contiene los datos de su última nómina. La macro MacroVerNomina la tendremos que haber definido previamente.

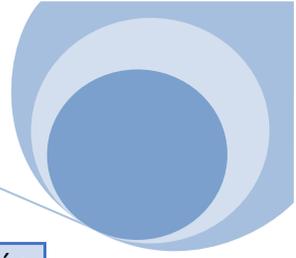
14.4 Acciones más utilizadas

En este apartado veremos las acciones más utilizadas en las Macros. Siempre puedes recurrir a la ayuda de Access para obtener información sobre acciones que aquí no tratamos. * Algunas de estas acciones no se muestran si no está pulsado el icono Mostrar todas las

acciones , en la pestaña Diseño.

Acción	Descripción
AbrirConsulta	Esta acción abre una consulta escogida entre las existentes en la base de datos.
AbrirFormulario	Del mismo modo que la acción anterior, abre un formulario.
AbrirInforme	Igual que las acciones anteriores, permite abrir un informe.
AbrirTabla	Esta acción permite abrir una tabla.



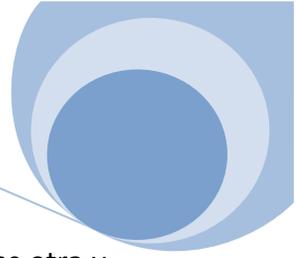


BuscarRegistro	Utilizaremos esta acción para buscar registros. Esta acción busca el primer registro que cumpla los criterios especificados. Puedes utilizar esta acción para avanzar en las búsquedas que realices.
BuscarRegistroSiguiente	Se posiciona en el siguiente registro que cumpla con los criterios indicados en la acción BuscarRegistro anterior. No tiene argumentos.
CancelarEvento	Esta acción cancela el evento que produjo la ejecución de la macro. No tiene argumentos.
CerrarVentana	Con esta acción podrás cerrar cualquier ventana que se encuentre abierta.
CuadroDeMensaje	Con las Macros incluso podremos mostrar mensajes para interactuar con el usuario.
DetenerMacro	Introduce esta acción en una Macro para detener su ejecución. No tiene argumentos. Verás su utilidad más adelante.
DetenerTodasMacros	Esta acción detendrá todas las Macros que se estén ejecutando en ese momento. No tiene argumentos.
*Eco	Esta acción es muy útil para ocultar al usuario las operaciones que se están realizando con una Macro. Permite la activación o desactivación de la visualización de las acciones en pantalla.
EjecutarComandoDeMenú	Utiliza esta acción para lanzar comandos que puedas encontrar en cualquier barra de herramientas.
*EstablecerValor	Una acción muy útil que te permitirá modificar los valores de los campos.
IrARegistro	Te permitirá saltar a un registro en particular dentro de un objeto.
MaximizarVentana	Esta acción maximiza la ventana activa para que ocupe todo el espacio de la ventana de Access.
MinimizarVentana	Al contrario que la anterior, esta acción minimiza la ventana activa convirtiéndola en una barra de título en la parte inferior de la ventana de Access.
SalirDeAccess	Esta acción hace que Access se cierre.

14.5 El flujo del programa

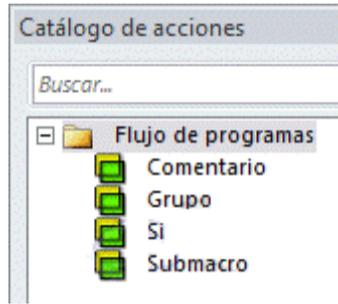
Como cualquier programa, una macro tiene un flujo que determina cómo se van a ejecutar las diferentes acciones que contiene.





Hemos dicho que, en principio, las acciones se ejecutan de forma secuencial, una tras otra y en el orden en que se han ido colocando. Pero esto no es siempre así. En ocasiones puede interesarnos que el programa fluya de una forma diferente, que ejecute una acción sólo si se cumple una determinada condición. Puede interesarnos dividir las acciones en bloques o crear una determinada jerarquía.

Access 2013 nos permite realizar todas estas acciones de forma muy sencilla e intuitiva. Si nos fijamos en el panel derecho Catálogo de acciones, encontraremos un apartado denominado Flujo de programas. En él encontraremos las opciones necesarias para estructurar la macro según nuestras necesidades.



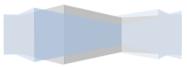
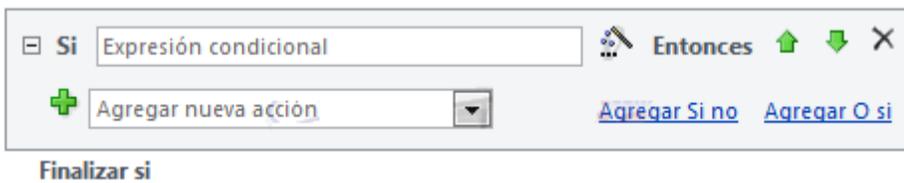
Para introducir un elemento sólo deberemos hacer doble clic sobre él. Se insertará en la pantalla principal que contiene las acciones y podremos modificarlo desde allí.

- Los comentarios son texto aclarativo que no se ejecuta. Son como notas del diseñador/programador para documentar lo que va realizando y hacer más legible la macro. Es muy recomendable utilizarlos para ir indicando lo que se va haciendo en la macro porque al cabo de un tiempo puede que no te acuerdes de lo que hacía la macro y te vendrán muy bien esos comentarios. Aparece un recuadro para escribir el texto, tal cual y cuando finalizamos vemos que se ha englobado entre unas etiquetas /* Así */ y se ha coloreado en verde para resaltarlo. */* Esto es un comentario aclarativo */*
- El grupo tampoco se ejecuta, es una forma de englobar en un bloque un conjunto de acciones, de forma que quede mejor estructurado.
- La opción Si permite que se ejecuten determinadas acciones únicamente si se cumple la condición especificada.
- Y la submacro es un conjunto de acciones que no entran en el flujo de ejecución directamente, sino que se han de activar de forma explícita mediante las acciones EjecutarMacro o AIOcurrirError.

14.6 Acciones condicionadas

Vamos a profundizar un poco en la opción condicional SI del flujo del programa.

Cuando introducimos una condición haciendo doble clic sobre la opción en el Catálogo de acciones, se introduce en la página principal como si se tratase de una acción más, con la salvedad de que nos permite establecer una condición de forma manual o mediante el Generador de expresiones (desde el botón en forma de varita).





Si esta condición resulta cierta, Entonces se efectuará la acción que indiquemos a continuación.

Podemos utilizar Agregar Si no para indicar las acciones que queremos que se ejecuten en caso de que la condición no se cumpla.

También podemos introducir otra condición, que ejecutará unas acciones distintas si no se cumple la condición del bloque anterior y sí la condición introducida al pulsar Agregar O si (para los que sepan programar es equivalente a un elseif).

Con estas simples herramientas podemos crear condiciones de lo más complejas.

La ventaja principal de la nueva interfaz para crear macros es la legibilidad del código, que queda perfectamente estructurado y es sencillo de interpretar a simple vista.

```
Si [Formularios]![Cursos tabular]![Nombre Curso]="Internet" Y [Formularios]![Cursos tabular]![Nº horas]<30 Entonces
  ! EstablecerValor ([Formularios]![Cursos tabular]![Nombre Curso]; "Diseño Web")
  ! EstablecerValor ([Formularios]![Cursos tabular]![Nº horas]; [Formularios]![Cursos tabular]![Nº horas]+5)
  ! CerrarVentana (Formulario; Cursos tabular; Si)
  CuadroDeMensaje (Los campos 'Nombre Curso' y 'nºhoras' se han actualizado correctamente.; No; Aviso; !; Actua...
  DetenerMacro
```

Finalizar si

Para sacar el máximo partido a las acciones condicionadas lo único que hay que tener claro es qué condición se tiene que cumplir y qué efectos queremos que tenga (las acciones). Para ello es recomendable hacer un análisis previo con el fin de tener bien claras las ideas antes de empezar.

Un ejemplo de condición simple sería [Formularios]![Cursos]![Código Curso] = 4 que únicamente desencadenaría las acciones si el contenido del control Código Curso del formulario Cursos es igual a 4.

Pero también podemos utilizar una expresión más compleja, por ejemplo:

- **[Formularios]![Cursos]![Fecha Inicio] Entre #2-Mar-2006# Y #9-Nov-2010#** para indicar si la fecha se encuentra en un intervalo.
- **EsNulo([Formularios]![Cursos]![nºhoras])** para realizar la acción si no hay nada guardado en el campo nºhoras.

O incluso unir varias condiciones utilizando los operadores Y y O como vimos en el tema de consultas.

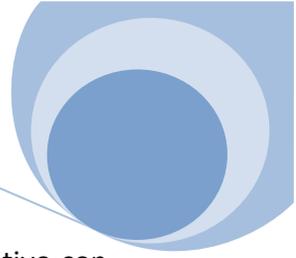
14.7 Grupos de Macros

Es posible que no le encuentres la utilidad en macros sencillas, pero en macros complejas puede ser conveniente agrupar las acciones. Para agrupar correctamente sólo hemos de pensar de forma lógica.

Por ejemplo, si en una macro tenemos las acciones:

- Abrir tabla de empleados
- Buscar registro del empleado con mayores ventas este mes
- Actualizar registro incluyendo un plus del 5% de sus ventas.
- Guardar registro
- Cerrar tabla de empleados

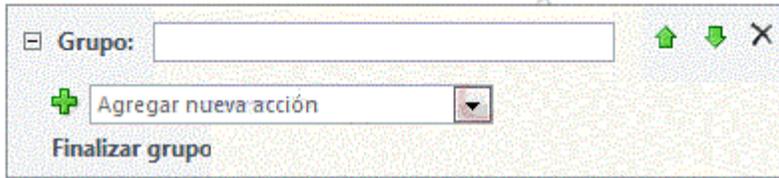




Podríamos perfectamente crear un grupo llamado Plus por ventas, que sería descriptivo con respecto a las acciones que ejecuta.

Al hacer doble clic sobre Grupo en el flujo de programas del Catálogo de acciones, veremos el siguiente panel en la página principal:

☰ Grupo:



Es tan sencillo como darle un nombre e ir incorporando las acciones en su interior.

Si ya habías definido las acciones y creas el grupo posteriormente, no te preocupes. Al igual que las acciones se pueden mover para variar su orden, también se pueden mover a distinto nivel, dentro de grupos, condiciones o submacros. Simplemente deberás arrastrarlas hasta su lugar de destino.

Los grupos en definitiva sirven para delimitar un conjunto de acciones que guardan cierta relación.

14.8 Submacros

Las submacros son agrupaciones de acciones, al igual que los grupos, pero con una finalidad y funcionalidad diferentes.

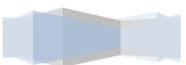
Al contrario que los grupos, la submacro no se ejecuta en el orden lógico del programa. Al ejecutar la macro que la contiene, esta es ignorada como si se tratara de un comentario. Únicamente podremos ejecutarla incorporando una de las dos acciones capaces de ejecutar las submacros: EjecutarMacro y AIOcurrirError.

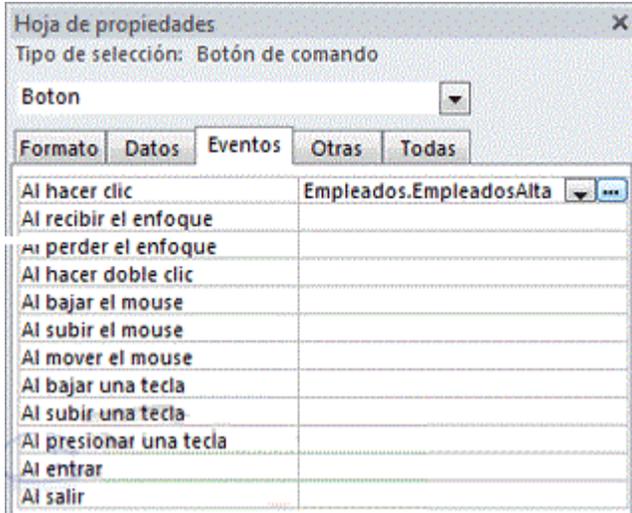
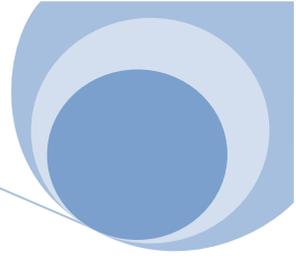
Las **aplicaciones prácticas más habituales** serán:

- **AIOcurrirError**: Para mostrar mensajes informativos al usuario del error ocurrido.
- **EjecutarMacro**: Para romper la estructura de ejecución lineal del código de la macro, con estructuras de repetición que veremos a continuación.

O bien para ejecutar la submacro de forma explícita desde el evento de un control. Veamos un ejemplo:

Imagina que tenemos una macro guardada con el nombre **Empleados**. En ella hemos creado las submacros **EmpleadosAlta**, **EmpleadosBaja** y **EmpleadosActualizar**. Rara vez ejecutaremos la macro **Empleados** completa, sino que nos referiremos a una de sus agrupaciones. Para ejecutar un alta, deberemos referirnos a ella como **Empleados.EmpleadosAlta**. Es decir, en el evento del control de formulario que ejecute la macro, deberemos escribir **NombreDeMacro.NombreDeSubmacro**.





14.9 Crear bucles de repetición

Hemos comentado que las submacros permiten crear estructuras de repetición utilizando la acción EjecutarMacro. Pues bien, ahora aprenderemos cómo implementarlas.



EjecutarMacro: Nos da la posibilidad de llamar a una macro (o submacro) y repetir su ejecución, definiendo así un bucle repetitivo.

Para ello deberemos especificar el Nombre de macro que queremos ejecutar, ya sea una macro completa (como Empleados) o una submacro (como Empleados.EmpleadosAlta).

Podemos indicar que la Macro se ejecute más de una vez escribiendo en Número de repeticiones un número que indique cuántas veces queremos que se repita la ejecución de la macro (el número deberá ser mayor que 1).

Y en Expresión de repetición podremos introducir una expresión condicional para que se repita la ejecución de la macro mientras la condición se cumpla (sea verdadera). Para componerla se nos dará la opción de utilizar el Generador de Expresiones. La condición se evaluará antes de ejecutar la macro así que si al empezar, la condición no se cumple, la macro no se ejecutará.

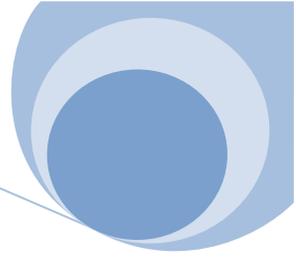
¡Ojo con los bucles infinitos! si utilizas el argumento Expresión de repetición la condición que pongas deberá depender de un parámetro que cambie dentro de la macro que se repite, sino podrás entrar en un bucle que no se acabe nunca.

Ahora veremos una estructura que ilustrará el uso de la combinación de varias técnicas: grupos, submacros y bucles. No se trata de un ejemplo práctico real, pero servirá para entender mejor los conceptos. Macro con Acción en submacro y un bucle que se encarga de ejecutarla tres veces

Como puedes ver en la imagen, hemos creado un grupo Bucle y una submacro Accion dentro de la macro PruebaBucle:

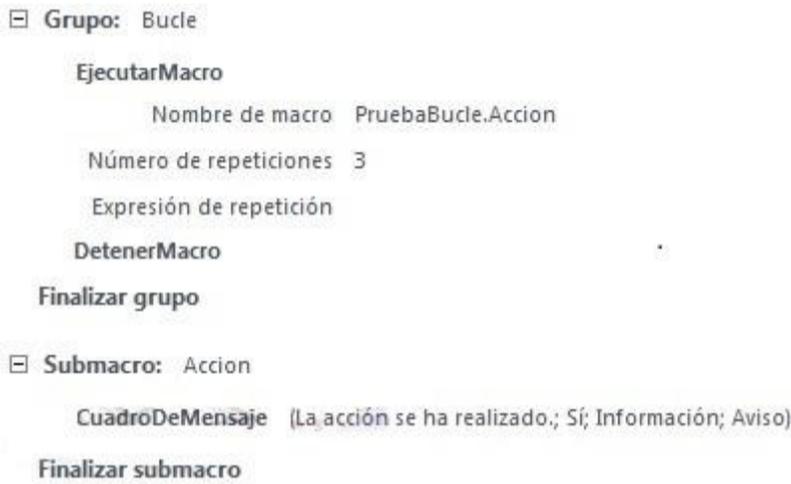
- por una parte tenemos una submacro Accion que en este caso muestra un mensaje,
- y por otra tenemos el bucle, que lo que indica es que se ejecute la macro de acción tres veces.





Podríamos trabajar de dos formas distintas con la submacro:

- Ejecutar la macro completa: Ejecutaría el código completo de forma secuencial. Es decir, entraría en el bucle que llama a Accion y lo repetería tres veces, mostrando tres mensajes de aviso.
- Ejecutar la submacro: Si en un control decidiéramos llamar en uno de sus eventos directamente a la submacro (PruebaBucle.Accion), se mostraría un solo mensaje de aviso ya que la submacro se ejecutaría únicamente una vez, ya que el resto de la macro, que incluye el bucle de repetición, no se leería.



14.10 AutoKeys o atajos de teclado

Las macros también te permitirán crear métodos abreviados de teclado. Esto es, podremos asignar a una combinación de teclas una determinada acción, a través de una macro. Sólo tendrás que seguir los siguientes pasos:

1. Crea una nueva macro y llámala AutoKeys.
2. Crea una submacro por cada combinación de teclas que quieras implementar.
3. Como nombre, cada submacro deberá contener una combinación de teclas que siga la sintaxis de la siguiente tabla:

Sintaxis de la combinación	Combinación de Teclas
^C or ^1	CTRL+C ó CTRL+1
{F3}	F3
^{F3}	CTRL+F3
+{F3}	MAYUS+F3
{INSERT}	INSERT
^{INSERT}	CTRL+INSERT
+{INSERT}	MAYUS+INSERT





Esto no significa que sólo puedas utilizar esas combinaciones. Recuerda, se trata de la sintaxis, pero podrías también utilizar **{F2}**, **{F4}**, **^{F5}**, etc. Lo único que deberás tener en cuenta es que la tecla **CTRL** se escribirá como **^**, y la tecla **MAYÚS** como **+**.

Las teclas que contengan más de una letra deberán ir encerradas entre llaves **{ }** para que no se confundan con combinaciones. Por ejemplo, END sería el equivalente a la combinación de teclas E+N+D, mientras que **{END}** equivaldría a pulsar la tecla FIN.

4. Incluye tantas acciones como desees que se ejecuten al pulsar las teclas.

Es recomendable que trates de comprobar que la combinación no exista ya por defecto en Access, porque le dará prioridad a tu macro. Por ejemplo, F1 ejecuta la ayuda de Access, si asignas una macro a F1 ya no dispondrás de ese práctico atajo.

Veamos un ejemplo:



En la imagen vemos cuatro submacros que nos permitirán que:

- Al pulsar F2 se abra la tabla de alumnos.
- Al pulsar F3 se abra la tabla de cursos.
- Al pulsar CTRL+F2 se abra el formulario de alumnos.
- Al pulsar CTRL+F3 se abra el formulario de cursos.

14.11. Depuración de errores

Cuando creamos una macro con diferentes condiciones y saltos de ejecución puede que al final nos resulte difícil comprobar que actúa correctamente.

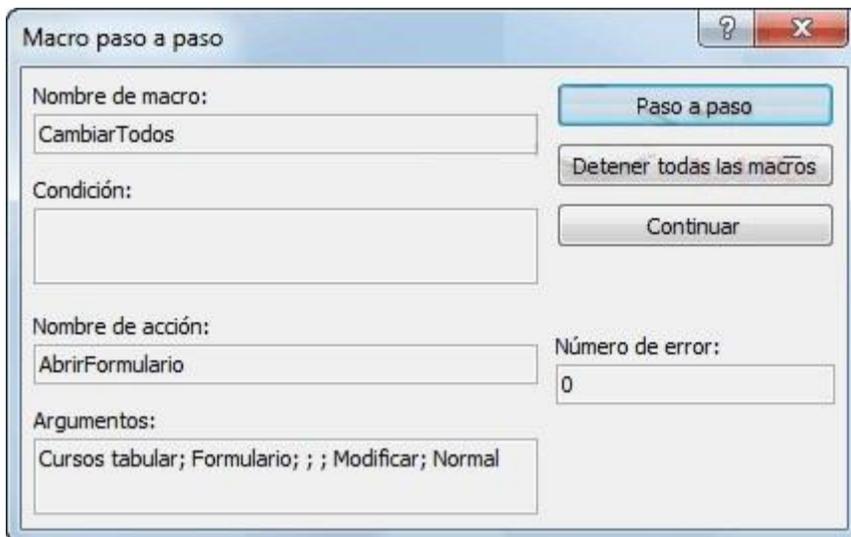
Para facilitar esta tarea, Access incorpora una herramienta que permite seguir la ejecución de la Macro paso a paso pudiendo en cada momento ver qué procesos se están llevando a cabo y los resultados que recibe de la base de datos.

Para activar esta opción solamente deberemos de hacer clic sobre el botón de Paso a paso icono paso a pa





en la pestaña Diseño. Una vez activada, cada vez que se ejecute una macro en el sistema lo hará en modo Paso a paso. Esto quiere decir que cada acción que se efectúe mostrará un cuadro de diálogo como el siguiente:



En él podrás ver:

- el **Nombre de macro** que se está ejecutando,
- la **Condición**, si es que tiene alguna definida, junto con su estado actual (verdadero o falso),
- y el **Nombre de la acción** que se va a ejecutar, junto con sus **Argumentos**.
- En caso de que haya ocurrido algún error, podrás ver cuál en Número de error. Este número te puede servir de referencia para buscar información sobre él en la documentación de Access o en internet.

Para **continuar con la reproducción Normal** de la macro pulsa **Continuar**. Si pulsas **Paso a Paso** seguirás en el mismo modo.

Puedes **detener** la macro pulsando el botón **Detener todas las macros**. En cualquier momento en la reproducción de una macro puedes detenerla también utilizando la combinación de teclas **CTRL+INTRO**. Esto te puede ser muy útil si una macro entra en un bucle infinito y el sistema deja de responder.

