

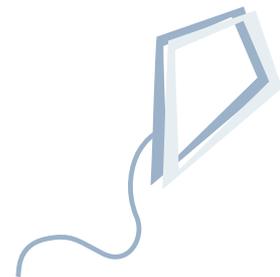


GOBIERNO  
DE ESPAÑA

MINISTERIO  
DE EDUCACIÓN

# PROGRAMACIÓN EN ACCESS

PROGRAMACIÓN



 AULA  
MENTOR

[educacion.es](http://educacion.es)



Nipo: 660-08-195-0

Autoría:  
**Alberto Carrera Martín**

Coordinación de contenidos:  
**Joaquín González Gigosos**

Edición y maquetación:  
**Verónica Borrego Polo**  
**Elena Aguado Sánchez**

Diseño gráfico de portada:  
**Lorena Gordo López**

# Programación en Access

## INDICE

0. Introducción.	
A quién va dirigido este curso .....	5
Introducción a la aplicación.....	6
¿Cuándo utilizar Access?.....	7
Distintas formas de programar con Access .....	8
Base de datos de ejemplo.....	9
1. El editor de VBA.	
Objetivos .....	10
Introducción.....	10
Nuestro primer módulo.....	16
Ayudas disponibles para la introducción de código .....	17
Recomendaciones a la hora de introducir código VBA .....	20
Tipos de módulos .....	21
2. El lenguaje VBA.	
Requisitos previos .....	22
Objetivo .....	22
Introducción.....	22
Variables y constantes .....	22
Alcance.....	26
Constantes.....	28
Constantes intrínsecas .....	29
Cómo referenciar los objetos .....	30
Operadores.....	31
Prioridad de los operadores .....	34
Instrucciones de control.....	35
Instrucciones de repetición .....	38
Procedimientos .....	40
Procedimientos Function.....	41
Procedimientos Sub.....	43
Paso de variables en procedimientos.....	43
Funciones predefinidas en VBA.....	44
Funciones de conversión de tipos .....	44
Funciones de fecha.....	46
Funciones matemáticas .....	49
Funciones de cadena de caracteres.....	49
Funciones de información .....	50
Función InputBox.....	51
Función MsgBox .....	52

3. Trabajando con formularios I.	
Objetivos .....	55
Introducción.....	55
Vista Diseño .....	56
Secciones de un formulario.....	58
Cambiar el tamaño de una sección.....	58
Agregar un botón de comando.....	59
Código VBA generado .....	61
Modificando el código VBA de un botón de comando.....	63
Formularios de sólo lectura .....	65
Procedimientos asociados a sucesos de formulario.....	67
Nomenclatura para referenciar los objetos .....	68
4. Cuadros combinados, filtros y grupos de opciones.	
Objetivos .....	69
Creación de cuadros combinados para buscar registros .....	69
Sincronizar el cuadro combinado con el formulario .....	77
Filtrado de datos. Tipos de filtrado.....	77
Filtrado de datos. Procedimientos de suceso.....	80
Filtrado de datos. Propiedades del formulario .....	81
Grupos de opciones .....	83
5. Trabajando con formularios II: Entrada de datos.	
Objetivos .....	91
Eventos de formulario y de control .....	91
Cancelación.....	92
Estructura de los procedimientos de evento.....	92
Personalización de la aplicación mediante los procedimientos de evento .....	93
Cambiar el valor de un control automáticamente .....	95
Validación de datos .....	97
Procedimiento de suceso Form_BeforeUpdate.....	98
Anexo: Tipos de eventos y orden de actuación.....	101
6. Procedimientos y funciones.	
Objetivos .....	104
Concepto e introducción. Motivos de su utilización .....	104
Módulos. Módulos de formulario.....	105
Módulos. Módulos estándar .....	109
Procedimientos y funciones.....	110
7. Tratamiento de errores.	
Objetivos .....	116
Introducción.....	116
Creación de puntos de interrupción.....	116
Tipos de errores .....	123
Errores de formulario y errores de procedimiento .....	124
Errores de procedimiento .....	127
Errores de formulario .....	129

8. Trabajando con formularios III: Otros métodos y propiedades.	
Objetivos .....	132
Introducción.....	132
Abrir un formulario .....	133
Cerrar formularios .....	138
Comprobar si un formulario existe.....	138
Comprobar si un formulario está cargado y en ejecución .....	140
Sincronizar formularios .....	142
Abrir un formulario para añadir registros relacionados.....	145
Redimensionar un formulario.....	152
Actualización de los datos de un formulario abierto por cambios realizados en otro .....	155
Formularios y subformularios .....	156
9. Cuadros combinados, de lista y otros controles.	
Objetivos .....	159
Introducción.....	159
Cuadros combinados .....	159
Convertir un cuadro de texto en un cuadro combinado .....	161
Añadir valores a un cuadro combinado.....	164
Cuadros de diálogo .....	166
Creación de un cuadro de diálogo y un cuadro de lista.....	167
Selección de registros en el cuadro de lista.....	171
Controles ficha y salto de página.....	174
10. El modelo de objetos de Microsoft Access.	
Objetivos .....	177
Introducción.....	177
El Examinador de objetos.....	178
Trabajando con las propiedades de los objetos .....	182
Trabajando con objetos y colecciones .....	183
Añadiendo nuevas propiedades a los objetos .....	186
Anexo: Intercambio de datos con otras aplicaciones.....	189
Manejo de Word desde Access .....	189
Enviando correo desde Access.....	193
11. El modelo de objetos ADO.	
Objetivos .....	195
Introducción.....	195
El objeto Connection .....	197
El objeto Command .....	199
El objeto Recordset.....	200
Localizar datos en un Recordset .....	204
Manipulación de datos en un Recordset.....	206
Utilización de transacciones .....	209
Manejo de errores .....	211
Anexo: Introducción al lenguaje SQL.....	212

---

12. Informes.	
Objetivos .....	222
Introducción.....	222
Creación de un informe con el asistente.....	222
Abrir informes .....	228
Cerrar un informe .....	232
Filtrado de datos en un informe.....	232
Gestión de errores.....	233
No abrir un informe vacío .....	233
Colorear secciones alternativas en un informe .....	234
Creación de subtotales .....	237
Eliminar filas en un informe reduciendo los espacios en blanco .....	239
Manejo de los parámetros de la impresora .....	241
13. Mejoras en la interfaz de usuario.	
Objetivos .....	244
Introducción.....	244
Creación de una barra de menús personalizada.....	245
Creación de una barra de herramientas personalizada.....	249
Programación de comandos de menú y botones de la barra de herramientas.....	250
Creación de un formulario de inicio de la aplicación I.....	252
Creación de un formulario de inicio de la aplicación II .....	256
Dividir la base de datos .....	258
Distribución de aplicaciones.....	259

# Introducción

*Alberto Carrera Martín*

## A QUIÉN VA DIRIGIDO ESTE CURSO

Microsoft Access es una potente herramienta de bases de datos que permite crear aplicaciones no solo monousuario sino también de tipo cliente/servidor. La utilización de la gran cantidad de asistentes que incorpora permite desarrollar aplicaciones de una manera sencilla y rápida. Si queremos aumentar todavía más la funcionalidad y posibilidades de esta herramienta es cuando necesitaremos utilizar el lenguaje de programación VBA que viene incluido en ella.

Para la práctica de los ejemplos y ejercicios se ha utilizado la versión Microsoft Access 2003 instalada en el equipo, pues en el momento de preparar el curso es la que se encuentra más consolidada, pero el alumno no tendrá el más mínimo problema si dispone de la versión anterior, Microsoft Access 2002 (conocida como Microsoft Access XP), pues los elementos y objetos utilizados están disponibles para ambas versiones, por este último motivo puede encontrar alguna diferencia si se utilizan versiones antiguas de Access como la 97 o 2000. También se ha revisado para la versión de Microsoft Access 2007 que ha aparecido recientemente en el mercado y que aunque presenta una interfaz de manejo de usuario muy diferente a las versiones predecesoras, no ocurre lo mismo ni con la sintaxis del lenguaje VBA ni con su interfaz de programación que se asemejan bastante a las de versiones anteriores.

El material que compone el curso no pretende ser un manual de programación en Access repleto de referencias y descripciones de los distintos componentes del lenguaje, pues para eso ya existe bastante bibliografía mucho más completa en el mercado, y lo que es mejor, una excepcional ayuda incorporada en la propia herramienta de Access así como en línea. Lo que se desea conseguir es que el alumno aprenda a desarrollar aplicaciones de bases de datos Microsoft Access practicando los conceptos con gran cantidad de ejemplos y ejercicios que encontrará a lo largo de este curso.

Una vez realizado el curso, el alumno podrá acreditar entre otros conocimientos de:

- Fundamentos del lenguaje VBA. Procedimientos y funciones.
- Programación de formularios, manejo de eventos, propiedades y utilización de controles. Tratamiento de errores.
- El modelo de objetos de Access. Relaciones entre Access y otros componentes de la suite de Office.
- Recuperación y manipulación de datos utilizando el modelo ADO y SQL.
- Programación y personalización de informes.
- Menús y barras de herramientas. Puesta a punto de la aplicación. Distribución y división de aplicaciones.

Este curso asume que el alumno que lo inicia posee algún conocimiento de manejo de Access a nivel de usuario y, sobre todo más importante que lo anterior, es que el alumno tenga algún tipo de experiencia, aunque sea pequeña, en otros lenguajes de programación puesto que conceptos como variables, constantes, operadores, expresiones, instrucciones, procedimientos, funciones... por citar algunos, se supondrá que ya han sido adquiridos por el alumno y éste podrá particularizarlos al lenguaje VBA en los próximos capítulos de este curso.

## INTRODUCCIÓN A LA APLICACIÓN

Microsoft Access apareció como sistema gestor de base de datos de escritorio en 1992, y ya por entonces se consolidó como el líder en su categoría tal y como ocurre en nuestros días con una comunidad de desarrolladores en Access que supera la cifra de 600.000.

Como entorno de desarrollo de aplicaciones, Access presenta unas características que lo hacen muy agradable para cualquier desarrollador como son:

- Fácil de utilizar. Con unos conocimientos mínimos se puede empezar a desarrollar aplicaciones. Además dispone de gran cantidad de asistentes para automatizar tareas reduciendo el tiempo de desarrollo y una ayuda integrada para aprovechar toda la potencia de Access.
- Forma parte de la suite Office y de esta manera amplía las capacidades del resto de componentes de la misma (como por ejemplo de Word y Excel).
- Permite la implementación de pequeñas / medianas aplicaciones de cualquier tipo: Facturación, contabilidad así como soporte para la toma de decisiones.
- Soporta aplicaciones cliente-servidor pudiéndose integrarse con grandes sistemas gestores de bases de datos como son Oracle, MySQL y sobre todo Microsoft SQL Server, utilizándose como generador de aplicaciones (front-end) para trabajar con estos grandes gestores de datos corporativos (back-end).
- Soporta también aplicaciones en Internet mediante las páginas de acceso a datos o guardando informes en formato de página Web.
- Puede manipular y compartir datos generados por otras aplicaciones u orígenes de datos, en un primer momento mediante ODBC, más tarde a través de OLE DB y a partir de la versión 2002 con los formatos XML y XSL.

## ¿CUÁNDO UTILIZAR ACCESS?

El deseo de utilizar Access está claro que es el de almacenar datos para luego ser tratados y obtener una información. Podríamos utilizar también para almacenar datos otros componentes de la suite de Microsoft Office como son Excel y OneNote. Si hacemos uso de la hoja de cálculo Excel podremos almacenar datos en filas y columnas de igual manera que en Access y además de una forma más fácil e intuitiva, puede ser una opción si el número de filas es pequeño. Lo que ocurre es que Access permite sacar más rendimiento a la información. Si se define un campo de una tabla de un determinado tipo de datos, por ejemplo numérico, Access no permitirá introducir en el mismo una cadena de caracteres u otro tipo de datos que no sea numérico. Este control no se puede establecer en una celda de una hoja de cálculo. Una forma de evitar datos repetidos en Access es mediante el establecimiento de claves primarias que garantizan que dos registros diferentes no van a tener almacenado el mismo valor en dicho campo; este control no se puede realizar en dos celdas diferentes de una hoja de cálculo. Además en Access pueden establecerse relaciones entre los campos de las tablas que van a condicionar el comportamiento y reglas de juego de la aplicación.

OneNote también permite registrar y organizar información (datos, imágenes, audio...), y de forma muy sencilla; lo que pasa que esta información no está estructurada como así ocurre en Access.

Por otro lado también podríamos utilizar otros productos de Microsoft como son MSDE y SQL Server. MSDE es un servidor de base de datos muy parecido a SQL Server, en realidad es una versión reducida de él para entornos de escritorio pues es el gestor de base de datos SQL Server sin herramientas gráficas de administración y además con disminución del rendimiento del sistema en cuanto se mantienen 8 accesos concurrentes. Si disponemos de una aplicación que requiere de las características de SQL Server (descritas más adelante) puede usarse MSDE en lugar de SQL Server, además todo el código escrito para MSDE puede utilizarse en SQL Server. El paquete MSDE se puede encontrar y utilizar sin coste tanto en el Visual Studio (la licencia de redistribución viene incluida en el propio Visual Studio) como en el Access Runtime incluido en el Access 2003. MSDE es robusto, trabaja bien con bases de datos superiores a 2 GB (Access tiene el límite en 2GB). Al ser una versión reducida de SQL Server, sufre también de las mismas vulnerabilidades, como p. ej. la del gusano Slammer que ha afectado a gran cantidad de sistemas provocando la ralentización o incluso la caída del sistema y por tanto hay que contar con las distintas actualizaciones y parches.

SQL Server es el más robusto de los sistemas gestores de base de datos tratados en este apartado. Su coste es elevado y necesita un equipo con un sistema operativo de servidor instalado. A diferencia de MSDE soporta bases de datos más grandes así como mayor número de usuarios y la posibilidad de crear bases de datos vía código.

## DISTINTAS FORMAS DE PROGRAMAR CON ACCESS

Primera forma: Utilizando macros. Como en otras aplicaciones, las macros proporcionan un conjunto limitado de herramientas para automatizar acciones en este caso en una base de datos, por ejemplo que al ejecutar una macro desde la ventana de Access o desde un botón de un formulario se abra un determinado formulario o informe. Son muy útiles pero limitadas. Realmente hay dos tareas que sólo se pueden hacer mediante macros:

- Asignar determinadas acciones a determinadas pulsaciones de teclas para que la aplicación sea más amigable para el usuario, p. ej. que al pulsar las teclas <ctrl.>+<v> se imprima el registro actual del formulario.
- Realizar determinadas acciones cuando una base de datos se abre. Cuando esto ocurre, en primer lugar se establecen para la aplicación las opciones de base de datos que figuran como opciones iniciales de la aplicación (se verá en el capítulo 13). A continuación se ejecutan las acciones que aparecen dentro de la macro *Autoexec*; entre estas podrían estar algunas que realizaran algún tipo de auditoría de comienzo de la aplicación.

No necesariamente hay que prescindir de las macros, muchos desarrolladores siguen utilizándolas por facilidad y costumbre y pueden seguir coexistiendo con el resto de las opciones de programación en Access.

Segunda forma: Utilizando SQL. SQL es el lenguaje que utiliza Access para almacenar las consultas de su base de datos. No es realmente un lenguaje de programación pero sus instrucciones pueden intercalarse entre las de un programa escrito en VBA (la tercera opción de programación que trataremos a continuación) para hacer este más claro, corto, rápido y legible. Además pueden utilizarse como veremos en este curso para definir determinadas propiedades (como p.ej. la propiedad *RecordSource* de un formulario, informe o lista desplegable) así como argumento de diferentes métodos de objetos (como p.ej. el método *Execute* de los objetos *Command* y *Connection*).

Tercera forma: Utilizando VBA (Visual Basic para Aplicaciones). El eje fundamental de este curso. Con este lenguaje de programación se puede automatizar cualquier operación que se necesite realizar en una base de datos. Las macros además, si así se deseara, se pueden convertir a código VBA.

Las ventajas de utilizar código VBA en lugar de macros son:

- Como se acaba de apuntar, la posibilidad de automatizar cualquier operación sobre la base de datos.
- El tratamiento de errores sólo se puede realizar mediante VBA.
- El código VBA es más rápido en ejecución que las instrucciones de las macros.
- El código VBA es más fácil de leer.
- Facilidad de mantener la base de datos. Si movemos un formulario de una base de datos a otra, automáticamente todo el código VBA que

contiene se mueve con él, cosa que no ocurre con las macros ya que son objetos que se guardan por separado.

- VBA es el lenguaje común a todas las aplicaciones de la suite de Microsoft y por tanto el código es fácil de portar e interactuar entre sus distintas aplicaciones (Word, Excel...).
- No se pueden pasar datos de una macro a otra ni interactuar con el usuario para recibir un dato (si en VBA).

## BASE DE DATOS DE EJEMPLO

En el fichero *bda\_ejemplo.mdb* encontrarás las tablas con las que practicaremos todos los ejemplos y ejercicios de este curso:

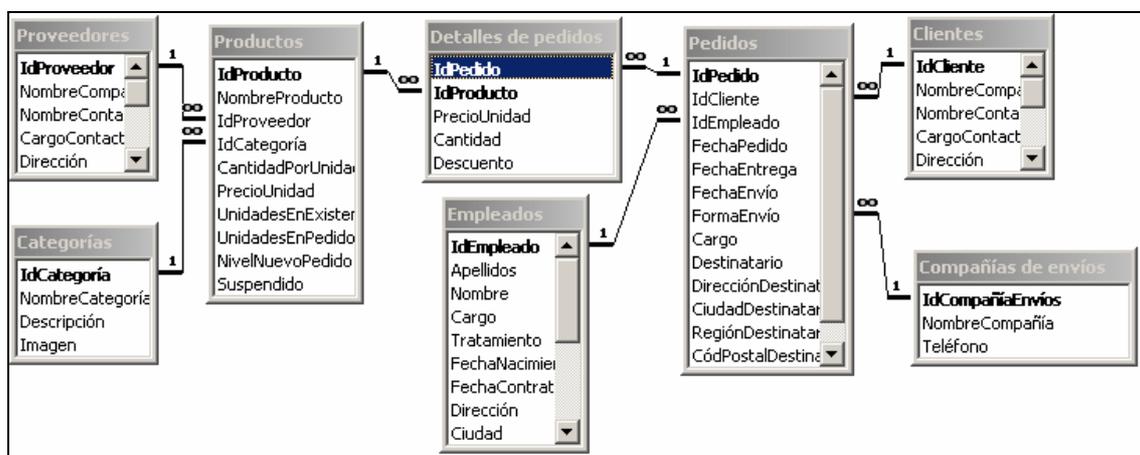


Ilustración 1. Base de datos de ejemplo

# Capítulo 1

## El editor de VBA

Alberto Carrera Martín

### OBJETIVOS

- Conocer el editor que incorpora Access para la escritura de programas, sus principales barras de menús, barras de herramientas y ventanas de la aplicación.
- Acceder al editor desde la ventana principal de Access y viceversa.
- Manejar la potente ayuda disponible que viene incorporada en el entorno de programación.
- Conocer algunas recomendaciones a la hora de introducir las instrucciones de los programas.
- Ejecutar procedimientos.
- Distinguir los tipos de módulos en un proyecto de Access.

### INTRODUCCIÓN

VBE es el editor de Visual Basic utilizado para escribir las líneas de código en el lenguaje Visual Basic para Aplicaciones (VBA).

Abre la base de datos *bda\_ejemplo*. Si utilizas una versión anterior a Access 2007 el aspecto es:

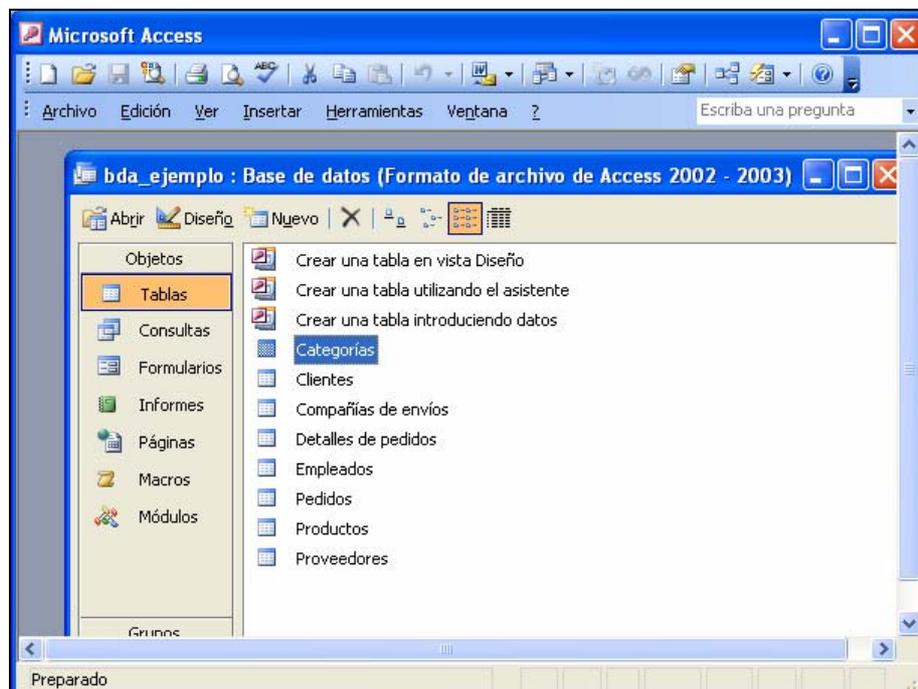
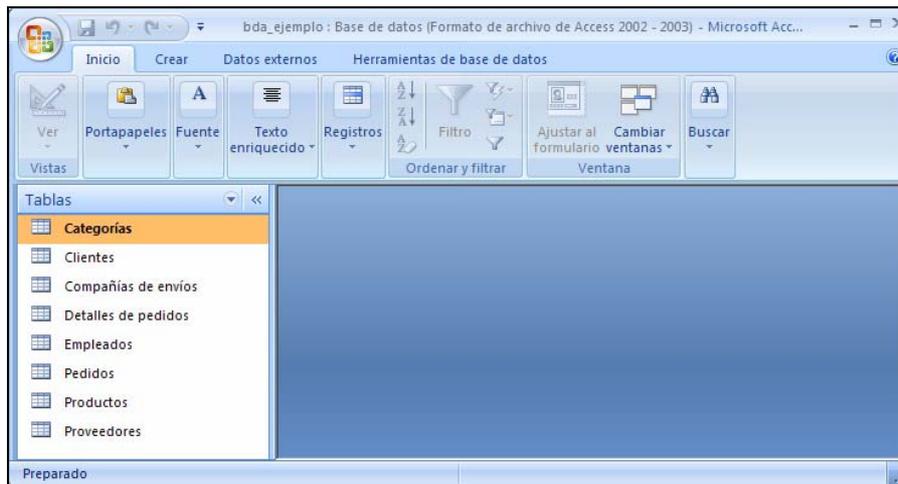


Ilustración 1. Ventana principal de Microsoft Access versión 2003

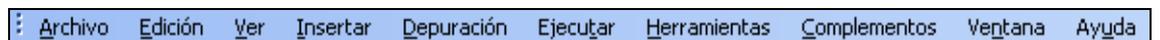
Si la versión empleada es Access 2007 entonces su aspecto cambia a:



**Ilustración 2. Ventana principal de Microsoft Access 2007**

Para acceder al editor lo puedes hacer de diversas formas. En este punto como todavía no hay introducida ninguna línea de código la forma más rápida de entrar es mediante la combinación de teclas <Alt.>+<F11>. Una vez en él te encontrarás una ventana con los siguientes elementos:

- La barra de menús predeterminada, debajo de la barra azul de título de la ventana:



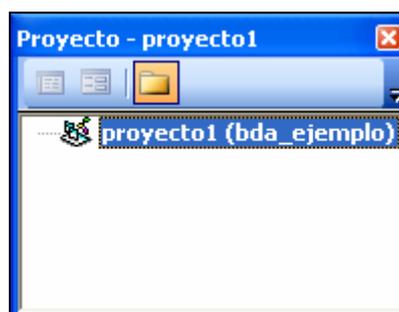
**Ilustración 3. Barra de menús predeterminada**

- La barra de herramientas Estándar:



**Ilustración 4. Barra de herramientas Estándar**

- La ventana Explorador de proyectos que muestra una lista jerárquica de los proyectos y de todos los elementos que contiene así como los referenciados por cada uno de los proyectos. Como puedes apreciar en la imagen solo hay un proyecto y además se encuentra vacío, sin elementos pues acabamos de empezar a desarrollarlo.



**Ilustración 5. Explorador de proyectos**

- La ventana Propiedades enumera las propiedades de los objetos seleccionados mostrando sus valores actuales con la posibilidad de poder cambiarlos.



Ilustración 6. Ventana Propiedades

- La ventana de Código que utilizaremos ya en este capítulo para introducir nuestras primeras líneas de código VBA. Quizás no aparezca como en la siguiente ilustración pues por el momento no tenemos ningún elemento en nuestro proyecto y en lugar de la imagen de la ilustración siguiente nos encontremos con un área extensa de pantalla con fondo oscuro.

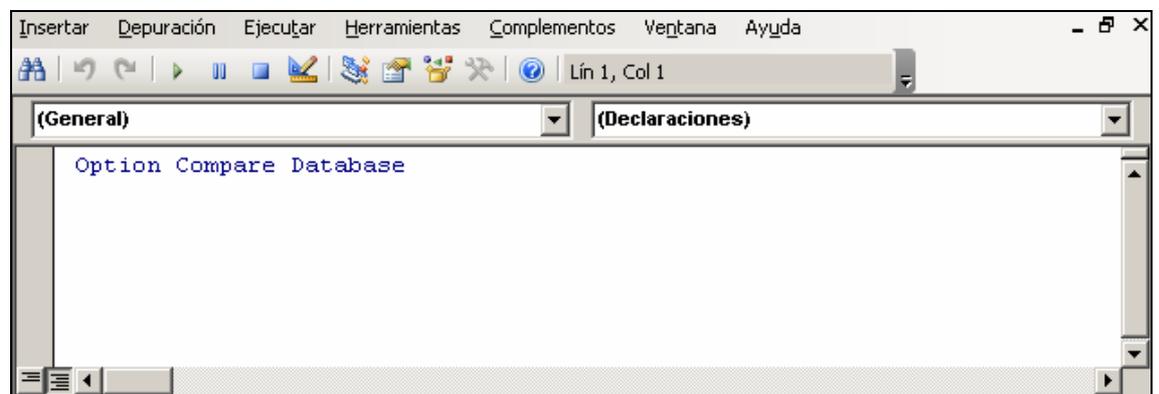


Ilustración 7. Ventana de Código

- La ventana Inmediato que puede utilizarse para mostrar los resultados de la ejecución del código (pulsas la combinación de teclas <ctrl.>+ <g> si no aparece en pantalla).



Ilustración 8. Ventana Inmediato



te encuentras todavía en él, vuelve a la ventana principal de Access mediante una de estas cuatro opciones:

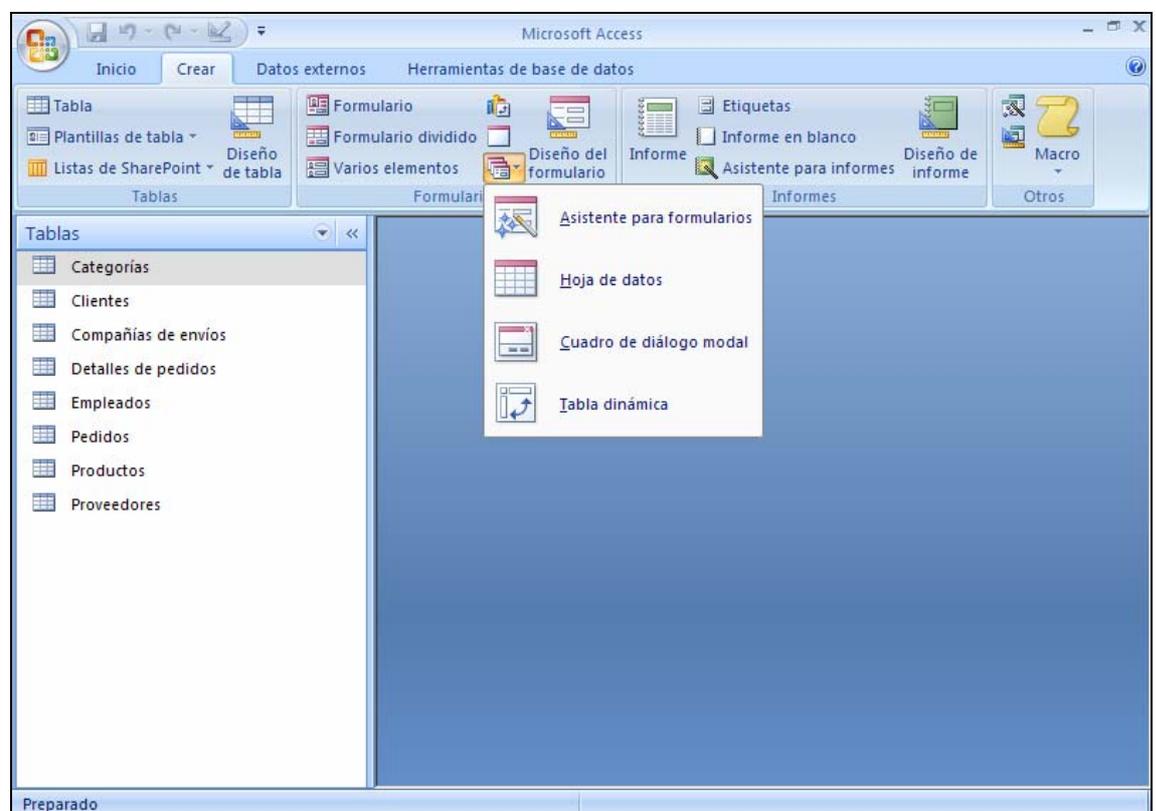
- Combinación de teclas <Alt.>+<F1>.
- Comando de la barra de menús *Ver* → *Microsoft Office Access*.
- Icono  de la barra estándar.
- Desde la barra de tareas del sistema operativo.

Una vez situado en la ventana principal de Access, si utilizas la versión de Access anterior a la 2007 (para esta última versión se explica en el siguiente párrafo), selecciona el objeto *Formularios* y después la segunda opción de éste: *Crear un formulario utilizando el asistente*. A partir de allí y en las sucesivas pantallas del asistente:

- Selecciona la tabla *Productos* con todos sus campos.
- Distribución en columnas.
- Estilo estándar.
- Título del formulario: *Productos*.

Si utilizas la versión de Access 2007 entonces los pasos para crear el formulario anterior desde la ventana principal serían:

- De la pestaña o solapa *Crear*, elige el botón *Más formularios* situado dentro del marco de *Formularios* entre los botones *Varios elementos* y el botón *Diseño del formulario*; a continuación selecciona la primera entrada, *Asistente para formularios*:



**Ilustración 12. Asistente para formularios en Microsoft Access 2007**

- A partir de allí y en las sucesivas pantallas del asistente:
  - o Selecciona la tabla *Productos* con todos sus campos.
  - o Distribución en columnas.
  - o Estilo oficina.
  - o Título del formulario: *Productos*.

Al volver al editor, independientemente de la versión de Access utilizada, te encontrarás con un proyecto con el que no contabas, el *acwzmain*.

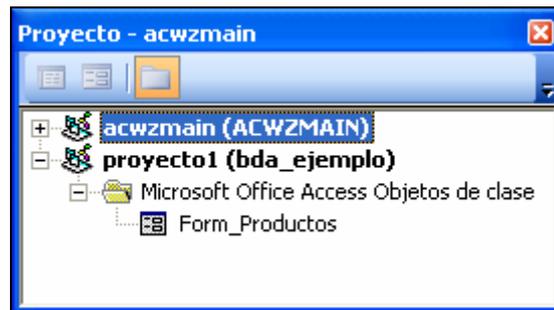


Ilustración 13. Proyecto acwzmain

Ello es debido a que los asistentes de Microsoft Access tienen este proyecto asociado que se carga en memoria cuando se utilizan éstos, en este caso por la creación del formulario mediante asistente, y se mantiene durante toda la sesión de Access siendo sus objetos inaccesibles para el usuario. Puedes comprobarlo cerrando la aplicación de Microsoft Access y volviendo a abrir la base de datos ejemplo, verás que ya no existe el proyecto. Quizás no te aparezca el icono del objeto de clase *Form\_Productos* que figura en la ilustración anterior pues aunque se ha creado el objeto (formulario) con el asistente, todavía no se ha introducido ninguna línea de código en dicho módulo de formulario.

Abre (si la has cerrado en el párrafo anterior) la aplicación de Microsoft Access haciendo doble clic con el botón izquierdo del ratón sobre la base de datos ejemplo; puedes volver al editor VBE con otras opciones además de la que ya conoces (entre ellas la combinación de teclas <Alt.>+<F11>), seleccionando el formulario *Productos* del objeto *Formularios* de la Ventana Principal y

- Comando *Ver* → *Código* de la barra de menús (versión anterior a Microsoft Access 2007).
- Icono  de la barra de base de datos (versión anterior a Microsoft Access 2007).
- Icono de *Visual Basic* del recuadro *macro* de la pestaña *Herramientas de base de datos* de la ventana principal de Access (versión Microsoft Access 2007).

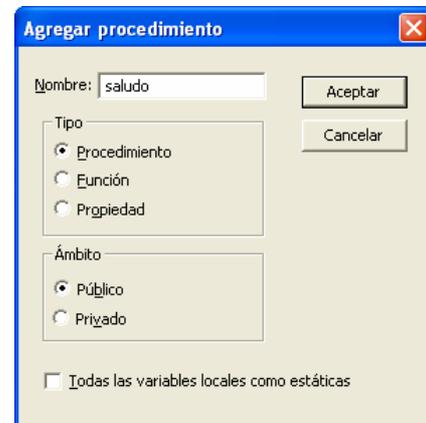


## NUESTRO PRIMER MÓDULO

Crearemos nuestro primer módulo en VBA y dentro de él un par de procedimientos:

De la barra de menús del editor VBE elige:

- 1) *Insertar* → *Módulo*  
y a continuación:
- 2) *Insertar* → *Procedimiento*.

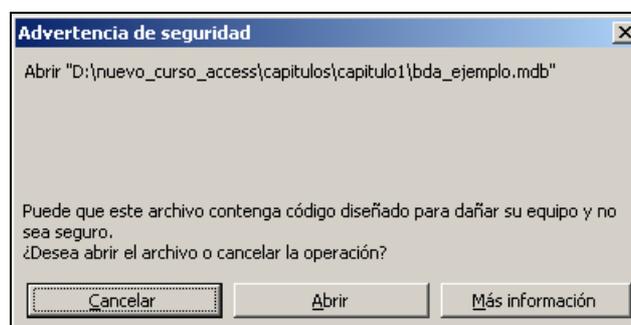


**Ilustración 14. Cuadro de diálogo "Agregar Procedimiento"**

- 3) Nombre del procedimiento: *saludo*. Deja las opciones restantes tal y como aparecen en la ventana de la ilustración anterior.
- 4) Botón *Aceptar*
- 5) Completa el procedimiento añadiendo la segunda línea:

```
Public Sub saludo()
MsgBox "Hola Mundo"
End Sub
```

Para ejecutarlo y ver los resultados bastará con que te sitúes sobre cualquier carácter del procedimiento y pulses la tecla <F5> (o comando de menú *Ejecutar* → *Ejecutar Sub / UserForm*). Puede ocurrir que hayas abierto Access en modo seguro de tal manera que no permita ejecutar ningún procedimiento o macro, en estos casos debes guardar el trabajo realizado como se indica en el párrafo siguiente y volver a abrir Access habilitando esta posibilidad: Si utilizas una versión de Access anterior a la 2007 pulsa el botón *Abrir* que aparece en la ilustración 15 en el momento en que abras el fichero, pues estás seguro de que esta aplicación no contiene código "malicioso":



**Ilustración 15. Advertencia de seguridad en Microsoft Access 2003**

Si utilizas la versión de Access 2007, entonces debes pulsar el botón *Opciones...* de la barra de advertencia de seguridad que aparece en la ventana principal de Access al abrir la aplicación:



Ilustración 16. Advertencia de seguridad en Microsoft Access 2007

y una vez hecho habilitar este contenido:

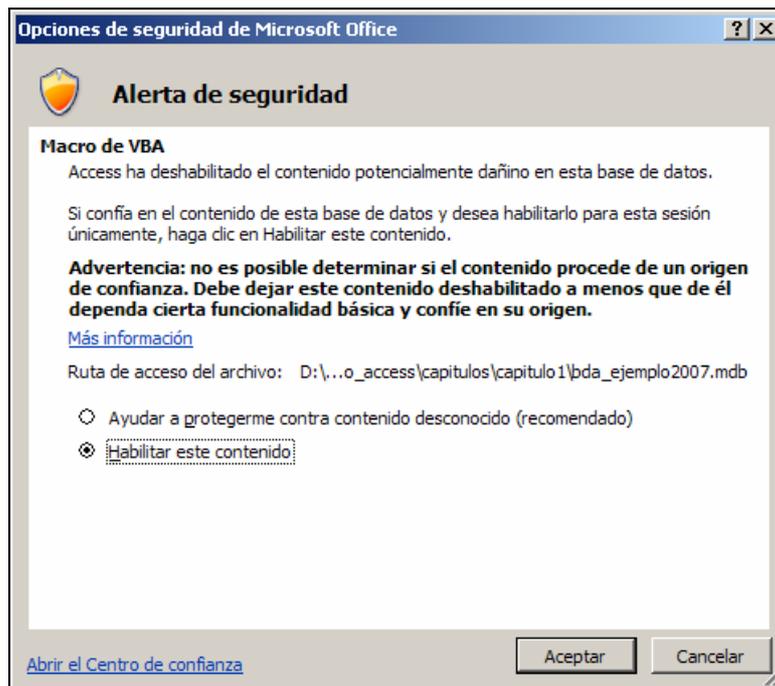


Ilustración 17. Habilitar la ejecución de código VBA en Microsoft Access 2007

Para guardar un nuevo módulo o almacenar los cambios realizados en un módulo ya existente antes de salir del editor VBE, haz clic sobre el botón *guardar* de la barra de herramientas estándar o el comando *Archivo* → *Guardar Nombre\_del\_Proyecto* de la barra de menús. Cuando se guarda un nuevo módulo, Access abre el cuadro de diálogo *Guardar como*, allí se puede aceptar el nombre del módulo propuesto o cambiarlo por otro distinto. El nombre del módulo se puede cambiar en cualquier momento seleccionándolo y modificando la propiedad nombre (*name*) de la ventana de propiedades.

## AYUDAS DISPONIBLES PARA LA INTRODUCCIÓN DE CÓDIGO

El siguiente procedimiento *Lista\_Productos()* que viene a continuación debajo del anterior muestra una lista de todos los nombres de los productos y sus existencias; nos servirá, aunque no lo podamos comprender por el momento, para utilizar una característica especial de VBE llamada *IntelliSense*, que ayuda a completar las palabras de las instrucciones que vayamos escribiendo limitando los posibles valores a los relacionados con la palabra que hemos escrito:

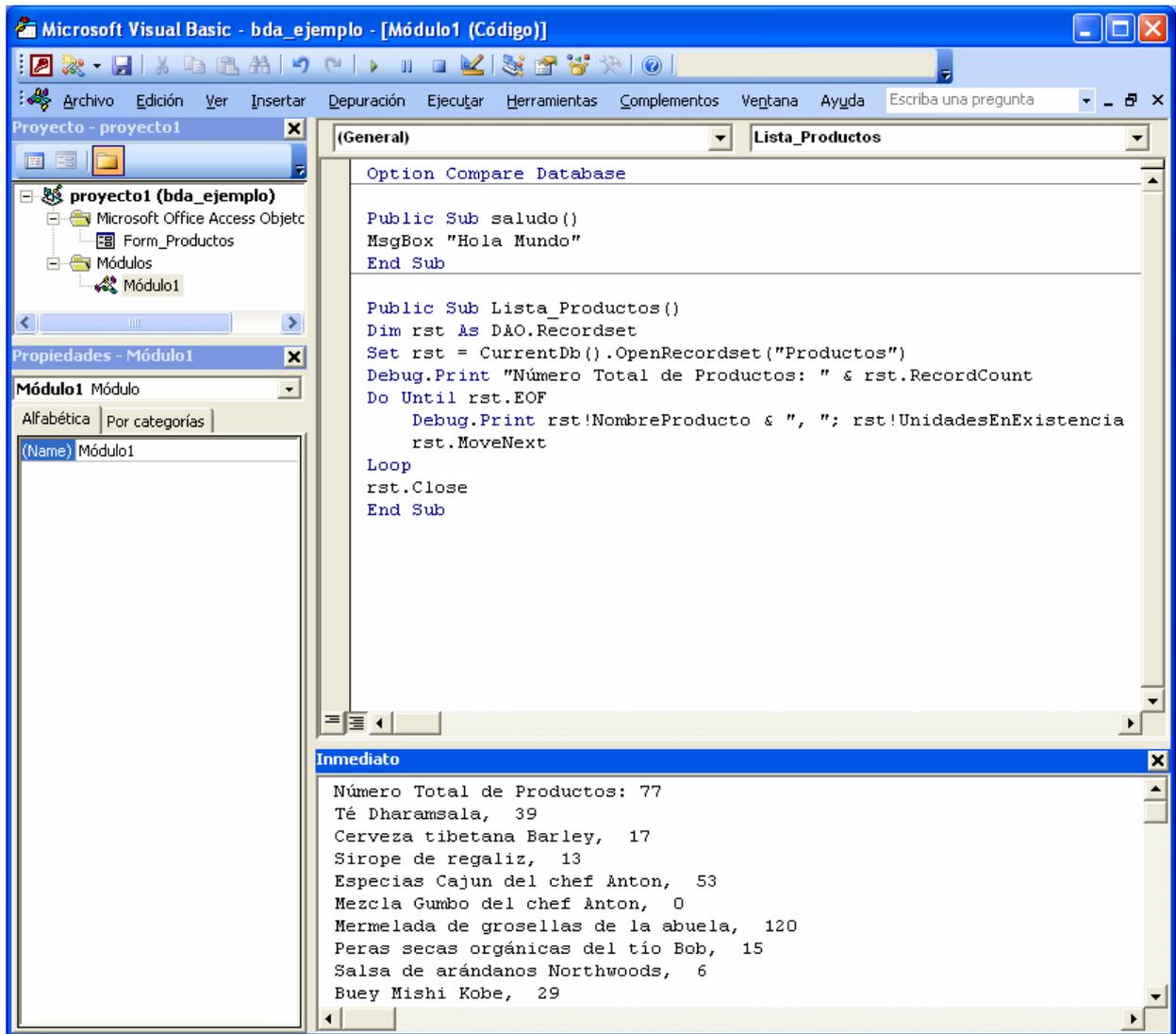


Ilustración 18. Módulo con dos procedimientos

A la vez que vas escribiendo el código, VBE lo va completando:

Nota: Aumenta si es necesario el zoom del programa lector con el que estás visualizando este documento para poder distinguir claramente cada carácter y cada palabra:



Ilustración 19. Característica IntelliSense

Tras escribir la palabra DAO (primera línea del procedimiento *Lista\_Productos*), después del punto aparecerá una lista de palabras relacionadas con ella; pulsando la letra R se salta a la posición de la lista con las palabras que comienzan por dicha letra. Para elegir la deseada basta pulsar la tecla <intro>:

Antes de ejecutarlo, asegúrate que la ventana de inmediato está abierta pues de lo contrario no podrás ver los resultados que aparecen por ella (ilustración 18).

VBE dispone de un potente y cómodo sistema de ayuda. Basta abrir la ventana de ayuda para acceder a toda esta gran cantidad de información bien estructurada. Se puede hacer, desde dentro de VBE mediante el comando de la barra de menús *Ayuda* → *Ayuda de Microsoft Visual Basic*.



Ilustración 20. Ventana de Ayuda

El funcionamiento es muy similar a cualquier otro sistema de ayuda. Se puede ir consultando los diferentes temas de la tabla de contenido a través de los hiperenlaces o buscar una palabra o texto en el recuadro de *Buscar* para encontrar todos los temas relacionados con la búsqueda deseada.

Una forma muy rápida de obtener ayuda de una palabra clave de nuestro código (función, propiedad...) consiste en seleccionar la palabra o situar el cursor al comienzo de la misma dentro del editor VBE y pulsar la tecla <F1>.

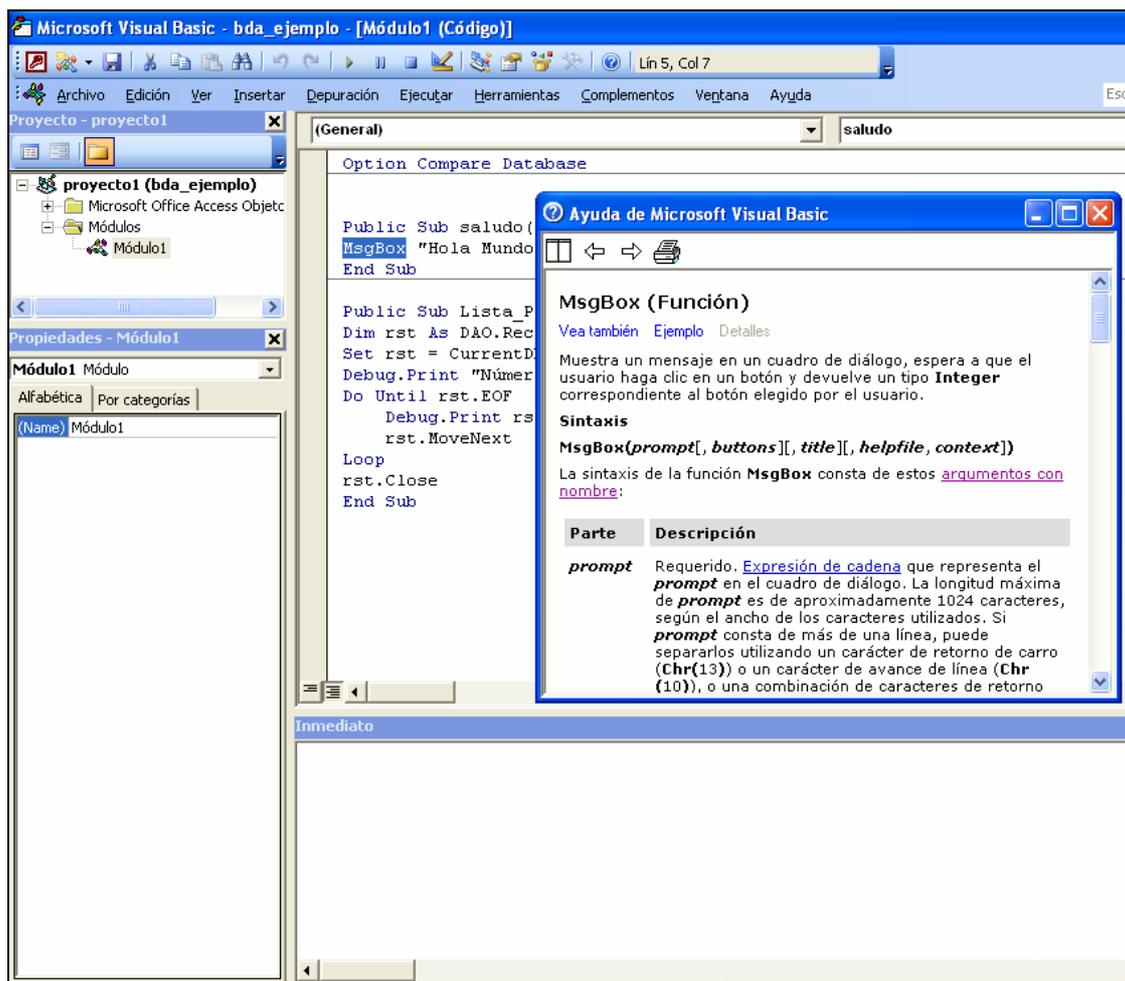


Ilustración 21. Consultando la Ayuda

Otra ayuda extra se puede obtener si se pulsa sobre la palabra clave el botón derecho del ratón y se elige de la lista desplegable la ayuda adicional que se desea.

## RECOMENDACIONES A LA HORA DE INTRODUCIR CÓDIGO VBA

Cuando escribamos código VBA deberemos tener en cuenta que ese código necesitará por diversos motivos (errores de ejecución encontrados, adaptar la aplicación a nuevos cambios...) ser revisado en un futuro. Por eso, desde un primer momento hay que tener en cuenta aspectos como:

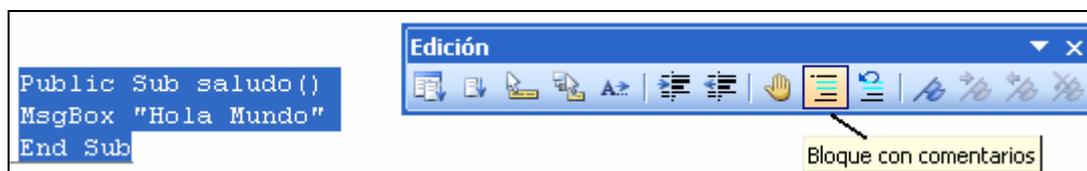
- 1) Utilización de nombres de objetos significativos, que indiquen por sí mismos la función que realizan. Aporta más información nombrar a un procedimiento como *Lista\_Productos()* en lugar de hacerlo como *xxx ()*.
- 2) Sangrar el código mediante tabuladores o espacios en blanco para apreciar mejor la estructura de flujo de los procedimientos tal y como se ha hecho en el ejemplo anterior con la instrucción *Do Until...Loop*. Puede hacerse de una forma rápida marcando el bloque a sangrar y con las opciones *Edición* → *Sangría derecha* o *Edición* → *Sangría izquierda* de la barra de menús.
- 3) Comentar las líneas de código, sobre todo aquellas que presenten mayor grado de complejidad. Para ello se utilizará el carácter de comilla simple ':

```
Public Sub Lista_Productos()  
'Visualización de algunos campos de la tabla Productos  
Dim rst As DAO.Recordset  
'La variable rst representa a la tabla Productos  
'de la base de datos actual  
Set rst = CurrentDb().OpenRecordset("Productos")  
'Imprimir por la ventana de Inmediato el total de productos  
Debug.Print "Número Total de Productos: " & rst.RecordCount  
'Mientras queden registros por recorrer de productos  
Do Until rst.EOF  
    'Imprimir el campo nombre y existencias del registro  
    Debug.Print rst!NombreProducto & ", "; rst!UnidadesEnExistencia  
    'Pasar al siguiente registro  
    rst.MoveNext  
Loop  
'Cierra y libera de memoria el conjunto de productos  
rst.Close  
Set rst = Nothing  
End Sub
```

Ilustración 22. Comentarios en el código VBA

Relacionado con este punto indicar que en algunas ocasiones en que hemos depurado procedimientos y hemos tenido necesidad de poner todo su código entre comillas para que no fuera interpretado a la hora de ejecutarse, en vez de tener que comentar línea a línea se puede hacer todo

el conjunto al mismo tiempo seleccionando el bloque de líneas y seguidamente el icono de bloque con comentarios de la barra de herramientas de Edición (desplegarla antes si no estuviera con el comando *Ver → Barras de Herramientas → Edición* de la barra de menús):



**Ilustración 23. Bloque con comentarios**

De esta manera todo nuestro procedimiento comentado quedaría:

```
'Public Sub saludo()
'MsgBox "Hola Mundo"
'End Sub
```

Para eliminar los comentarios selecciona todo el bloque y pulsa el icono de bloque sin comentarios a la derecha del anterior icono utilizado ( ).

## TIPOS DE MÓDULOS:

- 1) Estándar: Es el módulo (*módulo1*) que hemos creado en este capítulo con dos procedimientos. Contienen código VBA independiente que puede ser invocado desde cualquier parte del proyecto y por tanto desde cualquier tipo de módulo. Muchas veces se suelen incluir en estos módulos utilidades de la aplicación (operaciones con fechas, cálculo de días festivos, comprobar si un formulario está abierto, visualización de datos con un determinado formato...). Además pueden exportarse y utilizarse en otras bases de datos Access.
- 2) Objeto: Contienen los procedimientos asociados a formularios e informes, cada uno de ellos tendrá su módulo correspondiente. Cuando se escribe un procedimiento para responder a un determinado suceso o evento en la aplicación (apertura de un formulario p.ej.), el código se coloca en este tipo de módulos. Como se aprecia en la ilustración 24 todos ellos “cuelgan” de *Microsoft Office Access Objetos de clase*. Puedes comprobar que el único módulo de este tipo que tenemos hasta el momento se encuentra vacío.



**Ilustración 24. Módulo Objeto Form\_Productos**

- 3) Clase: Contienen código que define objetos personalizados. Se utilizan durante la creación de clases de objetos utilizados en la programación mediante objetos para de esta manera utilizar objetos propios distintos de los que aparecen en las librerías de Access. No se tratan en este curso.