

TEMA 1: INTRODUCCION

1. BASES DE DATOS

En Access una base de datos es **un archivo** que contiene datos (estructurados e interrelacionados) y los objetos que definen y manejan esos datos: tablas, consultas, formularios, informes, macros y módulos. Además, se pueden definir estrategias de seguridad y de integridad. Pueden coexistir en un disco varias bases de datos, en varios ficheros, absolutamente independientes entre sí (aunque es posible referenciar desde una base de datos objetos presentes en otras bases de datos, es decir, en otros ficheros).

NOTA: En algunos otros sistemas, como dBase, una base de datos es **un directorio** del disco en el que se encuentran multitud de archivos que contienen los datos y los objetos.

En las siguientes sesiones de prácticas comenzaremos a estudiar los diferentes objetos que permitirán crear una base de datos:

- **Tablas:** son los almacenes de datos. Se trata de listas de registros. El diseño de tablas consiste en la definición de los campos necesarios para almacenar correctamente la información de la base de datos.
- **Consultas:** operaciones que pueden realizarse sobre los datos de las tablas: limitaciones de acceso, selección de datos, inserción, modificación y borrado, cálculos...
- **Formularios:** pantallas que se presentan a los usuarios de una base de datos para que tengan un acceso amigable a los datos y operaciones.
- **Informes:** formatos de presentación de los datos para generar copias impresas de la información registrada en la base de datos.
- **Macros y módulos:** segmentos de código en lenguaje de alto nivel (Access Basic, Visual Basic, lenguaje de macros,...) que permiten realizar operaciones complejas con los datos y objetos de la base de datos.

Los objetos en Access deben estar identificados mediante un nombre. Dentro de cada grupo de objetos (tablas, consultas, formularios,...) los objetos se organizan como ficheros dentro de un directorio. Referencias posteriores a esos objetos se harán a través de su nombre identificativo.

2. ARCHIVO DE BASES DE DATOS .MDB

Vamos a comenzar a trabajar con Access. El primer paso va a ser crear una base de datos vacía. No se trata de un fichero vacío: al crear una base de datos vacía se almacenan en ella algunos datos administrativos y objetos del sistema, inicialmente ocultos al usuario y programador. Por tanto, antes de tener acceso a una base de datos "vacía", debemos elegir un nombre de fichero para almacenarla en disco.

Llegados a este punto es importante advertir que, al contrario de lo que ocurre en la mayoría de los programas de aplicación del mercado, los datos de una base de datos no necesitan ser explícitamente guardados. Estamos acostumbrados al hecho de tener que seleccionar la opción de guardado del menú archivo para que nuestros documentos y programas se almacenen en disco, mientras que el resto del tiempo trabajamos con una copia en RAM. En las bases de datos esto no es así. Los datos están siempre almacenados en disco. Únicamente se trabaja con una copia en RAM durante el tiempo en que un registro de una tabla está siendo accedido, y no es necesario utilizar una opción de guardado para hacer permanentes los cambios en el registro. Por lo tanto, cuando se introduzcan o borren datos de las tablas de una base de datos, Access (ni ninguna base de datos) preguntará si se desean guardar los cambios antes de cerrar, porque ya estarán guardados. Esto tiene como ventaja que no se perderán los datos en caso de que ocurra algún problema; como contrapartida sabemos que no será posible "deshacer" el borrado accidental de datos.

No obstante, Access tiene un entorno suficientemente amigable como para advertir al usuario de que puede realizar operaciones destructivas irreversibles antes de realizarlas.

Esto es únicamente aplicable a los datos. El resto de objetos en Access (definiciones de tablas, definiciones de consultas, formularios, informes,...) deben ser almacenados en disco dentro del fichero de la base de datos de forma explícita.

Por estas razones, Access requiere crear el fichero de una base de datos vacía antes de permitir trabajar con ella.

El fichero de una base de datos Access tiene extensión *MDB* (*Microsoft Data Base*). Además de este fichero, puede existir un fichero con extensión *LDB*. Este fichero contiene información de bloqueo de registros, que permite el acceso seguro de múltiples usuarios de forma concurrente. El fichero de la base de datos permanece abierto mientras se trabaja con los datos, y únicamente se cierra cuando se cierra la base de datos de forma explícita, con la opción correspondiente. Si por cualquier problema, no se realiza un cierre ordenado, la base de datos puede dañarse.

Cuando queramos volver a trabajar con una base de datos, lo único que hay que hacer es abrir de nuevo el fichero correspondiente, como si se tratara de un documento de un procesador de texto.

TEMA 2: TABLAS

1. INTRODUCCION

En este apartado vamos a estudiar la forma de definir tablas y trabajar con ellas. Las tablas son las estructuras que permiten almacenar los datos de la base de datos.

Una tabla es un conjunto de tuplas o registros. En este sentido podemos considerarla como un fichero que contiene *registros*, tantos como se quiera. Cada registro estará compuesto por una colección de *campos*. Cada campo tendrá un *tipo* que indica la clase de datos que puede almacenar. Y cada tipo tendrá unos atributos de *tipo* que limitan el dominio de los valores permitidos, así como una serie de restricciones y normas.

Por ejemplo, podemos tener una tabla para almacenar datos sobre los alumnos matriculados en una cierta asignatura. Esa tabla puede denominarse "Alumnos" y contendrá registros con, por ejemplo, los siguientes campos.

Campo	Descripción
NIF	NIF
Nombre	Nombre completo
Apellido1	Primer apellido
Apellido2	Segundo apellido
Edad	Edad
Parcial1	Nota obtenida en el primer parcial
Parcial2	Nota obtenida en el segundo parcial
Prácticas	Nota obtenida en las prácticas

Obsérvese que no se ha definido un campo *NotaMedia* para almacenar la nota final de la asignatura, quizá la más importante. Esto es debido a que es posible calcularla de alguna manera a través del resto de campos del registro. En una tabla no debe almacenarse información no necesaria. Además, si se almacenara, habría que recalcularla cada vez que se modificara alguno de los campos de los que depende, lo cual representa una fuente de inconsistencia. La forma adecuada de obtener la nota media es a través de consultas, que estudiaremos más adelante.

2. CREACION DE TABLAS

Una vez decididos los campos que necesitamos almacenar en la tabla, pasaremos a crear la tabla. Para ello seleccionaremos la opción correspondiente en el menú o en la ventana principal de la base de datos. Access nos mostrará un formulario para la definición de los campos. Esta ventana se denomina "de diseño de tabla" frente a la "de vista de tabla" que permitirá más adelante introducir los datos.

Para cada campo que definamos debemos asignarle un nombre, un tipo y una descripción. Además, para cada tipo existen una serie de atributos de tipo que permiten describir con precisión el dominio de valores aceptados por el campo.

Los tipos disponibles son los siguientes (Access 97):

- **Texto:** Una cadena de caracteres de longitud limitada. La longitud máxima se define como uno de los atributos, y no puede ser mayor de 255 caracteres. El espacio ocupado en disco por este campo corresponde a la longitud máxima definida. Un atributo importante del tipo texto es si se permite la longitud cero.
- **Memo:** Una cadena de caracteres de longitud ilimitada. El espacio ocupado en disco depende del texto almacenado. Tiene como desventaja respecto al tipo *Texto* que no permite algunas operaciones (orden, agrupación...).
- **Numérico:** Un número entero o real. Existen diversos tamaños y precisiones: Byte, Entero, Entero largo, real de precisión simple, real de precisión doble, identificador global. Dentro de cada uno de estos subtipos se pueden definir atributos de rango y precisión decimal.
- **Fecha/hora:** Fecha, hora o ambos a la vez. Existen diversos formatos de fecha y hora que pueden ser definidos como atributo.
- **Moneda:** Un número con formato monetario. Se trata de un valor numérico real con formato especializado para la manipulación de valores monetarios: abreviatura de la moneda local, separación con puntos y comas al estilo local,...
- **Autonumérico:** Un contador autoincrementado. Se trata de un valor numérico que el sistema genera automáticamente para cada registro de manera que nunca existirán dos registros en la misma tabla con el mismo valor en el campo autonumérico.
- **Si/no:** Un valor lógico.
- **Objeto OLE:** Un objeto de otra aplicación, vinculado mediante OLE: sonido, imagen, vídeo, gráfico...
- **Hipervínculo:** Un vínculo a un documento de Internet.

Cada uno de estos tipos dispone de atributos propios, por ejemplo, la precisión decimal y el rango de valores en los campos numéricos, frente a la longitud de cadena de los campos de tipo texto. Pero además existen una serie de atributos comunes, que se describen a continuación:

- **Formato:** define mediante una expresión el formato de los datos almacenados. Por ejemplo en campos numéricos define si se utilizan separadores de miles, o si el valor se interpreta como un porcentaje (%); en campos de texto puede forzar a que los caracteres se almacenen en mayúsculas; en campos lógicos si se utiliza SI/NO o VERDADERO/FALSO; etc. Al final del capítulo se incluye un anexo con los formatos de presentación de datos.
- **Máscara de entrada:** Fuerza a que los datos se introduzcan en un formato adecuado. Por ejemplo para un teléfono se puede ser interesante mostrar al usuario algo como "() ___-__-__", de manera que no pueda introducir caracteres extraños. La máscara de entrada es un conjunto de tres datos que definen los caracteres admitidos, el formato presentado al usuario y el formato almacenado. Al final del capítulo se incluye un anexo con información sobre formatos de máscara de entrada. El atributo *Formato* prevalece sobre la máscara de entrada, de manera que, al editar los datos del campo, una vez introducidos siguiendo una determinada máscara, el contenido del campo se mostrará con el aspecto definido en el *Formato* y no con el de la máscara.
- **Título:** Es el título de la columna que aparecerá en el modo de ver datos (al editar los datos almacenados) para el campo.
- **Valor predeterminado:** Un valor que automáticamente introducirá Access en el campo si el usuario no indica otro.
- **Regla de validación:** Una condición que debe cumplir el dato introducido para que sea aceptado. Por ejemplo, para un campo *Edad* se puede forzar a que el valor introducido esté comprendido entre 18 y 65 años con una expresión como ">=18 AND <=65".
- **Texto de validación:** Un mensaje que Access mostrará al usuario cuando intente introducir un valor no permitido por una regla de validación. Para el ejemplo anterior podría ser algo como "La edad debe estar comprendida entre 18 y 65 años."
- **Requerido:** Indica si el campo debe tener algún valor o se permite dejarlo en blanco. Por defecto un campo no contiene ningún valor. No hay que confundir un campo vacío con un campo de texto de longitud cero. De ahí la importancia del atributo de permitir longitud cero de los campos de tipo texto.
- **Indexado:** Establece si sobre el campo debe construirse un índice. Un índice es una estructura manejada por Access que hace que determinadas operaciones con el campo sean más rápidas (ordenaciones y búsquedas), a costa de un mayor espacio en disco y memoria (para almacenar el índice) y de que otras operaciones sean más lentas y costosas (inserciones y borrados). Existen dos formas de hacer índices: con y sin duplicados; esto permite configurar la posibilidad o no de que se repitan valores a lo largo de la tabla para un campo.

Para generar las expresiones que se pueden introducir para estas propiedades, Access proporciona un asistente que facilita la búsqueda de operadores y funciones. Para acceder a este asistente basta con pulsar sobre el botón "..."

que aparece a la derecha del editor (cuanto esté disponible). Quizá lo más importante del generador de expresiones es la posibilidad de explorar visualmente el conjunto de funciones incorporadas, entre las que destacan:

- Funciones de conversión entre tipos de datos.
- Funciones de fecha/hora. (Una función muy útil para su utilización como valor predeterminado es *fecha()*, que proporciona la fecha actual.)
- Funciones matemáticas.
- Funciones de tratamiento de cadenas.

A la hora de editar los datos en el modo de ver datos, Access, por defecto, utiliza un editor de texto en el que se pueden escribir los datos. A veces es útil proporcionar al usuario la posibilidad de elegir elementos de una lista, en vez de tener que escribirlos de nuevo. Para ello, Access permite seleccionar el tipo de editor que se va a utilizar para un campo (atributo *Búsqueda: mostrar control*):

- Cuadro de texto: un editor normal.
- Cuadro de lista: una lista de valores permitidos.
- Cuadro combinado: igual que el anterior, pero con más opciones.
- Casilla de verificación (sólo campos sí/no): un recuadro para activar y desactivar.

En el caso del cuadro de lista y del cuadro combinado, el conjunto de valores propuestos puede obtenerse de tres formas:

- Indicando explícitamente una lista de valores separados por símbolos de punto y coma.
- A través de una tabla/consulta, y entonces los valores propuestos se obtienen de los almacenados en una tabla (esta forma la estudiaremos más adelante).
- Indicando una tabla, y entonces los valores propuestos son el conjunto de campos de esa tabla (esto sólo es útil en tablas muy especializadas y no lo estudiaremos).

3. CLAVE PRINCIPAL

En toda tabla suele existir una clave principal (también llamada clave primaria o clave maestra). Una clave principal es un campo o conjunto de campos cuyos valores no se repiten y a través de los cuales se identifica de forma única al registro completo. Es decir, que no hay dos registros en una tabla con la misma clave. En el ejemplo de los alumnos, el campo *NIF* puede ser una clave principal, ya que no habrá dos alumnos con el mismo *NIF*, y además es posible identificar a un alumno a través de su *NIF*. El campo *apellido1* no puede ser clave principal porque puede haber más de un alumno con el primer apellido igual. El conjunto formado por (*apellido1*, *apellido2*, *nombre*) podría constituir una clave principal (suponiendo que no existen dos personas que se llamen

exactamente igual), ya que a través del conjunto podemos identificar de forma única e inequívoca al registro (al alumno).

Sobre un campo que se emplee como clave principal, forzosamente debe generarse un índice sin duplicados, y no se admitirán valores nulos para el campo.

Si no tenemos ningún campo o conjunto de campos candidato a clave principal, antes de terminar el diseño Access nos permitirá añadir un campo nuevo que sirva como clave. Este campo se denominará *id* y será de tipo autonumérico.

Por lo general, si en una tabla no podemos encontrar un campo o conjunto de campos para establecerlos como clave principal, lo más probable es que el diseño de la tabla no sea apropiado, y quizá deberíamos volver a plantearnos la estructura de tablas que necesita nuestra base de datos.

La declaración de índices y claves principales afecta al orden en que se muestran los registros a la hora de editar el contenido.

4. GUARDAR EL DISEÑO

Después de introducir todos los campos y definir sus tipos y atributos, podremos pasar a utilizar la tabla, introduciendo, modificando y utilizando los datos que puede almacenar. Antes deberemos guardar la definición de la tabla, utilizando la opción adecuada. Access pedirá un nombre para la nueva tabla.

Después de esto podemos pasar al modo de ver datos. Podremos volver a modificar la estructura de la tabla siempre que queramos con el modo de ver diseño. Entonces, antes de volver al modo de ver datos deberemos guardar la definición. Si ya hay datos introducidos en la tabla, Access no permitirá modificar la definición de la tabla, o impondrá restricciones, en el caso de que los cambios produzcan conflictos entre las nuevas definiciones y los datos ya almacenados (por ejemplo si intentamos cambiar el tipo de un campo de texto a tipo numérico y existen registros almacenando cadenas no numéricas).

Para el ejemplo de la tabla de alumnos, los campos definidos pueden tener las siguientes características:

Campo	Tipo	Descripción	Otros atributos
NIF	Cadena (9)	NIF	requerido; indexado sin repetición; máscara de entrada: <90.000.000\L; 0; "*">; clave principal.
Nombre	Cadena (40)	Nombre completo	requerido
Apellido1	Cadena (25)	Primer apellido	requerido
Apellido2	Cadena (25)	Segundo apellido	requerido
Edad	Numérico byte	Edad	>=18 AND <65;
Parcial1	Numérico simple	Nota primer parcial	1 decimal; predeterminado 0; >=0 AND <=10;
Parcial2	Numérico simple	Nota segundo parcial	1 decimal; predeterminado 0; >=0 AND <=10;
Prácticas	Numérico simple	Nota prácticas	1 decimal; predeterminado 0; >=0 AND <=2;

5 RELACIONES E INTEGRIDAD REFERENCIAL

En una base de datos lo habitual es que existan varias tablas relacionadas entre sí. Por ejemplo, si tenemos una tabla de productos de un almacén y una tabla de proveedores de productos, es probable que en la de productos exista un campo que indique cuál es el proveedor que suministró ese producto. En tal caso, debemos establecer unas restricciones sobre ambas tablas de manera que no existan productos que se refieran a proveedores que no existen, y no se nos permita eliminar ningún proveedor al que hagan referencia los productos.

La *integridad referencial* de una base de datos pretende asegurar que no se den situaciones de inconsistencia como la antes descrita. Para establecer las reglas de integridad referencial en Access debemos indicar qué campos de qué tablas están relacionados. La opción *Relaciones* del menú de herramientas (Access 97).

Esta opción permite diseñar un esquema de dependencias de forma visual. Disponemos de un escritorio sobre el que situamos las tablas de la base de datos, y mediante líneas, unimos los campos dependientes. Siguiendo con en ejemplo de los productos y los proveedores, supongamos que tenemos las siguientes tablas:

PRODUCTOS		PROVEEDORES
ID_producto		ID_proveedor
Nombre		Nombre
Proveedor		Dirección
Cantidad		NIF
		FAX

La tabla de productos no debe almacenar la información completa de los proveedores, ya que la haber más de un producto del mismo proveedor estaríamos almacenando información innecesaria (redundante). Así, sólo se almacena un pequeño código *Proveedor*, que se refiere a uno de los registros de la tabla *PROVEEDORES*: aquel cuyo *ID_proveedor* coincida. Es decir, que el campo *PRODUCTOS.Proveedor* hace referencia a *PROVEEDORES.ID_proveedor*. Para no perder información, ahora tendremos que asegurarnos de no eliminar entradas de *PROVEEDORES* que estén referenciadas por entradas de *PRODUCTOS*.

Existen diversas posibilidades para mantener la integridad referencial, correspondientes a las diversas respuestas que podemos hacer a la pregunta: ¿qué pasa si intentamos borrar un proveedor?

- a. **Lo ignoramos.** Como hemos visto no es recomendable.
- b. **No lo permitimos** (a menos que antes borremos todos los registros asociados).
- c. **Borramos** también todos los productos relacionados

En este caso la opción recomendable es la b), ya que para el ejemplo no tiene sentido que perdamos la información de nuestros productos al eliminar el proveedor; aunque en muchos otros casos la adecuada puede ser la c). En general, nunca la a).

También podemos hacernos la pregunta: ¿qué pasa si intentamos cambiar el identificador de un proveedor?

- a. **Lo ignoramos** y permitimos el cambio.
- b. **No lo permitimos** (a menos que antes modifiquemos todos los registros asociados).
- c. **Modificamos** también todos los productos relacionados.

En este caso la opción recomendable es la c), ya que es la que resulta transparente al usuario.

Al establecer una relación de dependencia en Access entre dos tablas es necesario especificar cuál de las posibilidades anteriores es la que queremos aplicar.

Existen tres opciones para definir la integridad referencial:

- **Exigir integridad referencial:** activa o desactiva la integridad referencial entre los campos de las tablas seleccionadas. Las demás opciones no pueden activarse si no se activa ésta.
- **Actualizar en cascada:** si se activa, hace actualizaciones en cascada; si no, no permite las actualizaciones.
- **Eliminar en cascada:** si se activa, hace borrados en cascada; si no, no permite los borrados.

TEMA 3: CONSULTAS

1. INTRODUCCION

Las consultas son operaciones que se realizan sobre los datos de una base de datos. Estas operaciones pueden ser de diversos tipos:

- **Consultas de selección de datos:** permiten recuperar los datos almacenados en las tablas en el formato y orden adecuados. Además permiten filtrar y agrupar la información. El resultado de estas consultas consiste en una tabla "virtual": una tabla que físicamente no ocupa espacio (porque trabaja sobre los datos originales de las tablas sobre las que se define), pero que permite ser manejada de la misma forma que una tabla real.
- **Consultas de inserción de datos:** permiten agregar registros a una tabla.
- **Consultas de modificación:** permiten modificar los valores de los campos de los registros de una tabla.
- **Consultas de borrado:** permiten eliminar registros de una tabla.
- **Consultas de creación de tabla:** permiten crear nuevas tablas cuyos campos y registros se obtienen a partir de los almacenados en otras tablas.

Dentro de las consultas de selección podemos resaltar algunos grupos importantes:

- **Consultas de selección simple:** permite filtrar tuplas y añadir o descartar campos de los registros. Se utilizan para crear "vistas": consultas que se utilizan como tablas para restringir el acceso a los datos a determinados usuarios.
- **Consultas de unión:** permiten relacionar los datos de distintas tablas a través de campos clave.
- **Consultas de agrupamiento:** permiten obtener resultados estadísticos de conjuntos de registros, como medias de un campo, totales, etc.

Las consultas a los datos de las bases de datos se hacen a través de los denominados "lenguajes de consulta". El más utilizado de este tipo de lenguajes es el SQL (*Standard Query Language*). Access permite realizar consultas en SQL. Pero, dada la filosofía visual seguida en Access, también se permite la definición de consultas a través de una interfaz visual. Ambos métodos son compatibles e intercambiables: se puede pasar de uno a otro sin problemas. Sin embargo, SQL es un estándar que aparece en multitud de sistemas sin variaciones, mientras que el desarrollo visual depende del fabricante.

TEMA 4: FORMULARIOS

1. INTRODUCCION

Los formularios son formatos de pantalla destinados a que el usuario los utilice como interfaz para la manipulación de los datos. Son equivalentes al modo de vista de datos de las tablas, pero permiten mayor libertad a la hora de personalizar los controles que se muestran en pantalla: colocación de los controles, aspecto de los mismos y de la pantalla en general, presentaciones anidadas, gráficos e imágenes, presentación de objetos multimedia, etc.

En general, un formulario está asociado a una tabla, de la que toma la información. Inicialmente el formulario contiene un conjunto de controles, cada uno de ellos asociado a un campo de la tabla subyacente. La modificación de los datos de estos controles se refleja automáticamente en los datos de la tabla; de hecho, todas las consideraciones hechas sobre la edición de datos sobre las tablas son aplicables a los formularios.

La creación y definición de formularios se hace de la misma forma en que se programa sobre un entorno visual. Los controles se sitúan sobre el espacio del formulario con ayuda del ratón, y las propiedades de todos los objetos implicados se muestran y modifican mediante diálogos, de manera que el código de programación del formulario queda oculto.

Se da por supuesto que el alumno conoce las metodologías de programación visual y sabe situar controles sobre un formulario, por lo que en las secciones siguientes sólo se comentará el significado de las propiedades de los principales objetos disponibles.

En Access 97 las propiedades de los objetos se agrupan en 4 conjuntos, según su funcionalidad. Un diálogo flotante muestra y permite modificar el estado de las propiedades de los controles de un formulario y del propio formulario. Este diálogo se muestra haciendo doble clic sobre cualquier control. Contiene 5 subcarpetas, que permiten acceder a los 4 conjuntos de propiedades, y un quinto conjunto que contiene todas las propiedades mezcladas. Los cuatro conjuntos son:

- **Formato:** Contiene propiedades referidas al aspecto visual de los controles: posición, tamaño, color, fuentes, formato de presentación de datos, etc.
- **Datos:** Contiene propiedades referidas a la asociación con los campos de la tabla relacionada: campos asociados, máscaras de entrada, valores predeterminados, reglas de validación, etc.
- **Eventos:** Contiene propiedades referidas a las respuestas que los controles ofrecen ante determinadas situaciones. Los eventos se conocen en otros SGBD como *triggers* o disparadores, y consisten en fragmentos de código en un determinado lenguaje que se ejecutan cuando se producen determinadas situaciones.
- **Otras:** Otras propiedades. Destaca el nombre del control, que es un identificador que permite referirse al control para acceder a sus propiedades.

Algunas propiedades de los controles, especialmente las del conjunto de Datos, pueden contradecir las propiedades que se establecieron para los campos a los que asocian dichos controles. Por ejemplo, reglas de validación y máscaras de entrada. En tal caso, prevalecen los definidos para el formulario, sin perjuicio de los definidos para los campos de la tabla. Por ejemplo:

- Una máscara de entrada definida para el control en un formulario prevalece sobre la definida para el campo, ya que la edición de los datos se realiza sobre el control del formulario.
- Una regla de validación definida para el control en un formulario debe verificarse para que el contenido del control sea aceptado por el control, pero además debe verificarse la regla definida para el campo de la tabla, porque de lo contrario la tabla no aceptará el contenido que le pasa el control (aunque sea válido para el control).

Por defecto, al asociar un campo a un control, el control hereda automáticamente las propiedades compatibles del campo, como las reglas de validación, los formatos de entrada y salida, etc.

2. CREACION DE FORMULARIOS

Los formularios, al igual que las tablas y el resto de objetos en Access, tiene dos modos de operación: uno para modificar la definición (vista de diseño) y otro para su utilización (vista de formulario). Y al igual que el resto de objetos, los formularios necesitan un nombre identificativo.

Existen asistentes que permiten crear algunos tipos habituales de formularios. Sin embargo, raras veces consiguen obtener el formulario tal y como se necesita, por lo cual no serán estudiados.

Al crear un nuevo formulario, aparece una ventana vacía (el formulario) sobre la que podemos situar los controles, y una paleta de herramientas, que contiene los controles disponibles.

El formulario en sí es un objeto que posee propiedades como si fuera un control. Antes de comenzar a añadir controles a un formulario deberían establecerse las propiedades del propio formulario, ya que permitirá que Access nos facilite luego la labor de añadir controles.

Antes de comenzar a estudiar las propiedades de un formulario, hay que distinguir dos tipos fundamentales de formularios y señalar que el formulario en sí está dividido en varias partes.

Existen **dos tipos de formularios**:

- **Continuos**: cada registro ocupa una línea, como en una tabla; debajo de cada registro hay otro registro.
- **Simple**: cada registro se presenta en una página completa. Debajo del registro no hay otro registro. La organización de las tuplas se asemeja a un libro de fichas: una página por registro.

Un tercer tipo, **Hoja de datos**, permite utilizar el formulario como si se tratara de una tabla.

Tanto los formularios simples como los continuos están divididos en varias partes. Estas partes pueden o no estar presentes en un formulario, y funcionan como objetos (controles) en el sentido de que tienen sus propias propiedades. Los controles del formularios pueden estar situados indistintamente en cada una de estas partes, dependiendo de la funcionalidad que se quiera dar al control. Estas partes son:

- **Encabezado del formulario.** Aparece siempre en la parte superior del formulario.
- **Encabezado de página.** Aparece siempre en la parte superior del formulario, entre el Encabezado del formulario y el Detalle.
- **Detalle.** Contiene los registros. Si el formulario es de tipo continuo, aparecen múltiples líneas, una con cada registro de la tabla subyacente. Si es de tipo simple, sólo aparece un registro, que va cambiando al avanzar y retroceder entre las páginas (registros). Si es demasiado grande para el tamaño de la ventana, o hay más registros de los que caben (en modo continuo), el Detalle se desplazará para visualizar las partes que no quepan, mientras que el resto de las partes del formulario no sufrirán cambios.
- **Pie de página.** Aparece siempre en la parte inferior del formulario, entre el Detalle y el Pie del formulario.
- **Pie del formulario.** Aparece siempre en la parte inferior del formulario.

Las principales propiedades de un formulario son las siguientes:

- **Título:** texto que aparece en la parte superior de la ventana.
- **Vista predeterminada:** formulario simple, continuo o tabla.
- **Vistas permitidas:** ¿se puede entre simple/continuo y tabla durante la ejecución?
- **Barras de desplazamiento:** qué barras de desplazamiento se mostrarán, en caso de que sea necesario.
- **Selectores de registro:** indica si debe mostrarse el recuadro que permite seleccionar el registro compeltto.
- **Botones de desplazamiento:** indica si los botones de la parte inferior del formulario que sirven para navegar entre los registros deben mostrarse o no.
- **Separadores de registros:** indica si debe mostrarse una línea de separación entre los registros (formularios continuos).
- **Ajuste de tamaño automático:** indica si el formulario debe adaptarse a su contenido la próxima vez que se abra.
- **Centrado automático:** indica si el formulario debe centrarse en pantalla la próxima vez que se abra.
- **Estilo de los bordes:** indica si se permite cambiar el tamaño de la ventana durante la ejecución.
- **Cuadro de control:** indica si debe existir el menú de ventana.

- **Botones Minimizar Maximizar, Botón cerrar, Botón qué es esto:** indica si debe aparecer los botones de maximizar, minimizar, cerrar y ayuda.
- **Ancho:** indica el ancho del formulario.
- **imagen, Tipo de imagen, Modo de cambiar el tamaño de la imagen, Distribución de la imagen, Mosaico de imágenes:** permiten poner una imagen de fondo en el formulario y la forma de colocar la imagen.
- **Línea X, Línea Y:** tamaño de las rejillas horizontal y vertical: número de subdivisiones horizontales y verticales por unidad de medida de la cuadrícula de alineación.
- **Origen del registro:** tabla o consulta de la que se obtienen los datos.
- **Filtro: expresión WHERE** que deben cumplir los registros para que sean visualizados.
- **Ordenar por:** lista de campos por los que se ordenan los registros.
- **Permitir filtros:** indica si se activan las opciones que permiten al usuario añadir filtros durante la ejecución.
- **Permitir ediciones:** indica si se permite modificar el contenido de los registros existentes.
- **Permitir eliminación:** indica si se permite eliminar registros existentes.
- **Permitir agregar:** indica si se permite añadir nuevos registros.
- **Entrada de datos:** indica si el formulario sólo permite añadir datos.
- **Bloqueos de registro:** indica la forma en que deben bloquearse los registros utilizados para permitir el acceso concurrente de varios usuarios.
- **Emergente:** indica si el formulario debe quedar delante de las demás ventanas de Access.
- **Modal:** indica si se permite activar otros formularios mientras el formulario esté ejecutándose.
- **Ciclo:** indica si después del último control, la tecla de tabulación debe saltar al primer control, al primer control visualizado o al siguiente registro.
- **Barra de menús, Barra de herramientas, Menú contextual, Barra de menús contextuales:** permiten cambiar el menú y las barras de herramientas de Access mientras se ejecuta el formulario.
- **Archivo de ayuda, Id. del contexto de ayuda:** permiten especificar un archivo de ayuda de Windows y el índice del tema para el formulario.

En la lista anterior se han omitido las propiedades del conjunto de eventos. Antes de tratarlas debemos estudiar los eventos.

3. EVENTOS

Un evento es una acción determinada que se produce en o con un objeto particular. Access puede responder a una variedad de eventos: clics del ratón, cambios en los datos, formularios que se abren o se cierran, y muchos otros. Los eventos son normalmente el resultado de una acción del usuario.

Cada objeto o control está preparado para responder a un conjunto de eventos, coherentes con el tipo de control y su utilidad. Se puede establecer una **macro** para cada evento de cada objeto, de manera que cuando se produce ese evento en ese objeto, se ejecuta el código de la macro. Un ejemplo básico consiste en asociar una macro que abra un formulario al evento de pulsación de un botón, de manera que cuando el pulse el botón, se abra el formulario. De esta forma se pueden hacer programas tan complejos como se quiera con los formularios de Access.

Para muchas situaciones existen dos eventos: uno denominados "*Antes de XXX*" y otro "*Después de XXX*". Ambos eventos se producen cuando se da la situación *XXX*, con la diferencia de que uno de los eventos tiene lugar justo antes de que la situación *XXX* haga su efecto, y el otro justo a continuación.

A continuación se listan los principales eventos para formularios y controles.

Eventos para los formularios

- **Al activar registro:** cuando el enfoque pasa a un nuevo registro.
- **Antes de insertar, Después de insertar:** cuando se añade un nuevo registro.
- **Antes de actualizar, Después de actualizar:** cuando se modifica un registro.
- **Al eliminar, Antes de confirmar la eliminación, Después de confirmar la eliminación:** al eliminar un registro existente.
- **Al abrir:** al abrir el formulario.
- **Al cargar:** al cargar el formulario.
- **Al cambiar el tamaño:** cuando se cambia el tamaño de la ventana del formulario.
- **Al descargar:** al descargar el formulario.
- **Al cerrar:** al cerrar el formulario.
- **Al activar:** cuando el formulario es activado, es decir, pasa al frente de la pantalla y comienza a funcionar.
- **Al desactivar:** cuando el formulario es desactivado, es decir, deja de estar al frente de la pantalla y se activa otro.
- **Al recibir enfoque:** cuando el enfoque pasa al formulario.
- **Al perder enfoque:** cuando el enfoque pasa del formulario a otro objeto.
- **Al hacer clic:** cuando se presiona y se suelta el botón izquierdo del ratón sobre la superficie del formulario.

- **Al hacer doble clic:** cuando hace doble con el botón izquierdo del ratón sobre la superficie del formulario.
- **Al bajar el mouse:** en el momento en que se presiona el botón izquierdo del ratón.
- **Al subir el mouse:** en el momento en que se libera el botón izquierdo del ratón.
- **Al mover el mouse:** en el momento en que se desplaza el puntero ratón sobre la superficie del formulario.
- **Al bajar una tecla:** en el momento en que se presiona una tecla.
- **Al subir una tecla:** en el momento en que se libera una tecla.
- **Al presionar una tecla:** cuando se produce una pulsación completa (presionar y liberar) de una tecla.
- **Tecla de vista previa:** indica si los eventos de pulsaciones de teclado deben afectar antes a los controles del formulario o al propio formulario.
- **Al ocurrir un error:** cuando se produce un error en tiempo de ejecución.
- **Al filtrar:** cuando se modifica el filtro sobre los registros de la tabla asociada.
- **Al aplicar el filtro:** cuando se aplica o elimina un filtro sobre los registros de la tabla asociada.
- **Al cronómetro:** cada cierto tiempo.
- **Intervalo de cronómetro:** número de milisegundos entre dos eventos Al cronómetro sonsecutivos.

Eventos para los controles

Para controles independientes ("pasivos", no asociados a campos de tablas): textos, dibujos, etc.:

- **Al hacer clic:** cuando se presiona y se suelta el botón izquierdo del ratón sobre la superficie del control.
- **Al hacer doble clic:** cuando hace doble con el botón izquierdo del ratón sobre la superficie del control.
- **Al bajar el mouse:** en el momento en que se presiona el botón izquierdo del ratón sobre el control.
- **Al subir el mouse:** en el momento en que se libera el botón izquierdo del ratón sobre el control.
- **Al mover el mouse:** en el momento en que se desplaza el puntero ratón sobre la superficie del control.

Para controles dependientes ("activos", que contienen información asociada a campos de tablas): editores, cuadros de lista, pulsadores, etc. (además de los eventos anteriores):

- **Antes de actualizar, Después de actualizar:** cuando se modifica el contenido del control (en el momento en que se realiza el cambio definitivo sobre el campo o registro).

- **Al cambiar:** al cambiar el contenido del control, sin que sea necesario que se cambie definitivamente sobre el registro o campo.
- **Al entrar, Al recibir el enfoque:** cuando el control recibe el enfoque desde otro control.
- **Al salir, Al perder el enfoque:** cuando el control pierde el enfoque a favor de otro control.
- **Al bajar una tecla:** en el momento en que se presiona una tecla.
- **Al subir una tecla:** en el momento en que se libera una tecla.
- **Al presionar una tecla:** cuando se produce una pulsación completa (presionar y liberar) de una tecla.

4. CONTROLES

En un formulario se pueden incluir tantos controles como se desee. Cada control debe ir identificado por un nombre, que puede ser diferente del nombre del campo al que esté asociado (aunque a menudo suelen coincidir, por regularidad). Si se ha definido la tabla asociada a un formulario, Access nos permite añadir un control para cada campo de forma fácil, simplemente arrastrándolos con el ratón desde una ventana que contiene la lista de campos disponibles. Los controles así añadidos adoptan las propiedades que se deriven de los atributos asignados en la definición de los campos de las tablas de origen.

Cuando se agregan controles a un formulario, éstos adoptan una determinada apariencia por defecto en cuanto color, tamaño, fuentes, y otras propiedades. Los valores por defecto de estos atributos pueden ser cambiadas a través de un control de ejemplo (con la opción *Definir valores predeterminados de los controles* en el menú *Formato*).

Los controles dependientes, por defecto añaden automáticamente una etiqueta al ser añadidos. Esta etiqueta se mueve o borra al mover o borrar el control, pero no al revés.

Cada control tiene un nombre. Se puede acceder al contenido de un control a través de su nombre. Por ejemplo, podemos establecer una regla de validación que verifique que el valor del control actual no sea superior al de un control cuyo nombre es "*ValorMaximo*". La expresión que habría que introducir en la propiedad "Regla de validación" del control sería " \leq *ValorMaximo*".

Cada control dependiente suele ir asociado a un campo, pero no es necesario. En tal caso el control actúa como una mera variable cuyo valor puede ser modificado por el usuario. No hay que confundir por lo tanto el nombre de un control con el nombre del campo al que está asociado, aunque a menudo sean iguales. Otra posibilidad es que los controles muestren el resultado de algún cálculo (lo veremos más adelante).

Al cambiar de registro activo, los controles asociados a los campos de la tabla se cargan automáticamente con los valores del nuevo registro. De la misma forma, las modificaciones que se realicen sobre los datos de los controles se reflejan automáticamente en la tabla.

Propiedades comunes

A continuación se listan las principales propiedades que tienen casi todos los controles:

- **Visible:** indica si el control debe mostrarse o permanecer oculto.
- **Mostrar cuando:** indica si el control es visible en pantalla o en al imprimir el formulario.
- **Izquierda:** indica la posición horizontal, es decir, la distancia desde el borde izquierdo del formulario, en cm.)
- **Superior:** indica la posición vertical, es decir, la distancia desde el borde superior de la sección actual del formulario (encabezado, detalle, pie...) en cm.
- **Ancho, Alto:** dimensiones del control, en cm.
- **Color del fondo, Efecto especial, Estilo de los bordes, Ancho de los bordes, Color de los bordes, Color del texto, Nombre de la fuente, Tamaño de la fuente, Fuente en cursiva:** definen el aspecto general del control. Los colores se definen como color real RGB, codificado como un número de 24 bits, 8 para cada color. El efecto especial permite dar un aspecto tridimensional al formulario.
- **Origen del control:** campo asociado al control.
- **Valor predeterminado:** igual que en la definición de los campos de las tablas.
- **Regla de validación:** igual que en la definición de los campos de las tablas.
- **Texto de validación:** igual que en la definición de los campos de las tablas.
- **Activado:** indica si el control se puede utilizar o está desactivado.
- **Bloqueado:** indica si el control se puede utilizar o está inutilizado.
- **Nombre:** nombre identificativo del control, para referencias desde otros controles.
- **Texto de la barra de estado:** texto de ayuda que aparece en la barra de estado de Access cuando el control recibe el enfoque.
- **Punto de tabulación:** indica si se puede acceder al control pulsado la tecla de tabulación.
- **Índice de tabulación:** indica en qué número de orden se accede al control pulsado la tecla de tabulación.
- **Texto de ayuda del control:** texto flotante que aparece sobre el control al situarse el puntero del ratón sobre el control.
- **Id. del contexto de ayuda:** índice del tema de ayuda relacionado con el control dentro del archivo de ayuda establecido por el formulario.
- **Información adicional:** un comentario que puede almacenarse sobre el objeto.

Como se ha visto, la asociación de un control a un campo se hace a través de la propiedad **Origen del control**. Si esta propiedad se establece al nombre de un campo, el control queda asociado a ese campo, y las modificaciones en el control se reflejará en los datos del campo. Si la propiedad se deja en blanco, el control es independiente y actúa como una variable. Finalmente, si en la propiedad **Origen del control** se introduce una expresión comenzando por un símbolo igual (=), entonces el control contiene un valor calculado: el resultado de evaluar la expresión.

Por ejemplo, si tenemos un campo "*Precio*" con su control correspondiente con el mismo nombre, podemos añadir un control llamado "*PrecioConIva*" que nos muestre el resultado de añadirle el IVA, sin que necesitemos almacenarlo en ningún campo. Esto lo haríamos escribiendo la siguiente expresión en la propiedad *Origen del control* del control *PrecioConIva*: " $=\text{Precio} * 1,16$ ". Cada vez que cambie el contenido de *Precio* automáticamente se recalcula el contenido de *PrecioConIva*.

Propiedades de los controles de listas

Estas propiedades son comunes a los controles de tipo *Cuadro combinado* y *Cuadro de lista*. Ambos tienen en común el hecho de que pueden contener una lista de valores permitidos para seleccionar el valor del control (y por tanto del campo asociado, si lo hubiera).

Estos controles tienen algunas propiedades específicas que permiten manejar la información referente al contenido de la lista de valores:

- **Tipo de origen de la fila:** indica si el contenido de la lista se tomará de los datos contenidos en alguna tabla, de una lista de valores explícitamente especificada, o de una lista de campos pertenecientes a alguna tabla.
- **Origen de la fila:** dependiendo del valor de la propiedad anterior, especifica, bien la tabla de origen para obtener los datos o campos a mostrar, o bien la lista de valores permitidos, separados por punto y coma (;).
- **Número de columnas:** especifica el número de columnas que se mostrarán en la lista. Tiene especial significado cuando los valores permitidos se obtienen de una tabla, la cual, puede contener varios campos. Esta propiedad indica cuántos campos de la tabla se mostrarán en la lista. El orden de aparición de los campos es el mismo en que se encuentren definidos en la tabla.
- **Encabezados de columna:** indica si en la parte superior de la lista debe aparecer el título de la columna, indicando el nombre del campos visualizado.
- **Ancho de columnas:** especifica el ancho, medido en centímetros, de cada una de las columnas mostradas con los valores permitidos. Se trata de una lista con el ancho de cada columna, separado por punto y coma (;). Se permite ancho 0.
- **Columna dependiente:** cuando una lista de valores permitidos se muestra con más de una columna, sólo una de estas columnas es la que

proporciona el valor para el campo. Esta propiedad indica cuál es el número de orden de la columna.

- **Filas en lista:** indica cuántas líneas con valores permitidos deben mostrarse en pantalla como máximo. Si hay más, se mostrará una barra de desplazamiento.
- **Ancho de la lista:** indica el ancho total de la lista de valores. Por defecto (automático) el ancho de la lista es el mismo que el tamaño en horizontal que tenga el control de lista, y **no la suma de los anchos de las columnas mostradas**. Si no caben las columnas en el ancho especificado se muestra una barra de desplazamiento horizontal.
- **Limitar a lista:** indica si el valor aceptado por el control debe ser obligatoriamente uno de los visualizados en la lista, o si, por el contrario, se permite que el usuario introduzca uno distinto.

5. SUBFORMULARIOS

Algunas veces resulta interesante anidar formularios, es decir, incluir formularios dentro de otros formularios. Supongamos como ejemplo una base de datos para almacenar nuestras cintas de música. Tenemos dos tablas: canciones y cintas:

Supongamos que queremos presentar un formulario en el que aparezcan los datos de una cinta, y todas las canciones que contenga. La forma más adecuada de hacer esto es utilizando subformularios. El proceso es el siguiente:

1. **Construcción del subformulario.** En nuestro caso, se trata del formulario que contiene los datos de las canciones.
2. **Construcción del formulario principal.** En nuestro caso, el de cintas. Debe contener un control de tipo subformulario, que se refiera al formulario de canciones.
3. **Enlace de los formularios.** Hay que asociar dos campos, uno en el formulario principal y otro en el subformulario, de manera que en el subformulario sólo se muestran las filas que contengan un campo cuyo valor coincida con el de un determinado campo en el principal. Así, no se mostrarán todas las canciones, sino sólo aquellas que pertenezcan a la cinta que se muestra en el principal. Se trata de los campos que establecen la relación entre ambas tablas: *Canciones.Cinta* y *Cintas.IDcinta*.

Para la construcción del subformulario, debemos tener en cuenta lo siguiente:

- Probablemente nos convenga utilizar un formulario de tipo continuo: muchas canciones para la misma cinta; parece lógico que sea un formulario de tipo continuo (como una tabla).
- Probablemente el campo que enlazará el subformulario con el formulario principal (*Canciones.Cinta*) no se muestre en el subformulario: parece que no tiene sentido que para todas las canciones de la cinta se muestre en el formulario repetido el mismo valor de la clave externa. Además,

este campo será llenado por Access de forma automática al valor adecuado al agregar nuevas filas a la tabla de canciones.

- No hay que establecer filtros ni condiciones sobre el campo dependiente, ya que Access lo hará automáticamente al establecer la dependencia.
- Probablemente sea adecuado no introducir encabezados y pies en el subformulario, ya que al ir integrado dentro de otro formulario, estas partes podemos introducirlas directamente en el formulario principal.

En cuando al formulario principal, hay que tener el cuenta que al introducir el control de subformulario, hay que darle el tamaño y forma adecuado para que quepa en él el subformulario completo. Si es de tipo continuo, hay que darle el ancho adecuado para que no sobre ni falte espacio, y el alto adecuado para que quepan el número de filas deseado.

Por último, hay que establecer las siguientes propiedades del control subformulario:

- **Objeto origen:** nombre del formulario utilizado como subformulario.
- **Vincular campos secundarios:** nombre del campo dependiente en el subformulario (que establece la relación entre el formulario principal y el subformulario: *Canciones.Cinta*).
- **Vincular campos principales:** nombre del campo o control dependiente en el formulario principal (que establece la relación entre el formulario principal y el subformulario: *Cintas.IDcinta*). En el subformulario sólo se mostrarán las filas para las cuales el valor del campo establecido en *Vincular campos secundarios* coincida con el del control establecido para esta propiedad.

Los campos de vinculación no han de estar incluidos con controles en el formulario principal o en el subformulario. Es suficiente con que estén en las tablas o consultas base de éstos.

TEMA 5: INFORMES

1. INTRODUCCION

Los informes son formatos de presentación de datos para la recuperación de datos por impresora. Son semejantes a los formularios en cuanto a la forma de definirlos y manejarlos, con la diferencia de que en los informes aparecen algunas características propias del medio físico al que van orientados, es decir, el papel (y no la pantalla).

Tal es la similitud entre formularios e informes que Access utiliza las mismas herramientas y controles para ambos. Incluso es posible convertir un formulario a un informe. Al fin y al cabo, los formularios también podían imprimirse.

Todo lo estudiado en el capítulo anterior sobre diseño y manipulación de formularios es aplicable a los informes. Además, estudiaremos las posibilidades adicionales de los informes, que no existían en los formularios por propia naturaleza.

2. CONTROLES

No existe ninguna diferencia en el uso de controles en informes respecto a lo estudiado para formulario. Los controles son idénticos y conservan todas las propiedades presentes para los formularios, excepto aquéllas que se refieren a la modificación de datos (reglas de validación, eventos sobre la modificación, etc...), ya que los informes no permiten la modificación de los datos, sino sólo mostrar el contenido de las tablas o consultas.

Adquiere un especial significado el control de salto de página, que no se utilizaba en los formularios porque no existía el concepto de página (salvo al imprimir).

3. AGRUPAMIENTOS

Una característica de gran utilidad en el diseño de informes es la posibilidad de agrupar los datos según el valor de un campo o conjunto de campos. A diferencia de una consulta con agrupamiento (que utilice GROUP BY), este tipo de agrupamiento que ofrecen los informes no se limita a la utilización de funciones de agregado, sino que permite introducir separadores entre conjuntos de tuplas con una característica común (que sirve como cláusula de agrupamiento).

Por ejemplo, supongamos un gran listado de alumnos admitidos para la realización de un curso. Para generar este listado parece interesante ordenarlos alfabéticamente e imprimir una letra en grande al principio de cada grupo para facilitar la búsqueda, como en la figura 4.1.

Access permite agrupar los datos en múltiples conjuntos anidados (p.e. en un listado de alumnos de la Universidad, se puede agrupar primero por *escuela*, luego por *carrera* y luego por la primera letra del *apellido*). Los grupos se pueden generar en orden creciente o decreciente, y para cada uno de estos

grupos puede existir o no un encabezado y un pie de grupo. El encabezado de grupo aparece justo antes de los registros que pertenecen al grupo, y el pie justo después. Típicamente en los encabezados y pies se muestran funciones de agregado calculadas sobre los registros que pertenecen al grupo (p.e. cuenta de elementos, totales...). Esto se puede hacer fácilmente con cuadros de texto con funciones del tipo " $=\text{Suma}([\text{Precio}]*[\text{Cantidad}])$ " (que calcularía la suma total para una sección del producto de los campos *Precio* y *Cantidad* de cada registro).

Los grupos se pueden definir sobre valores de un determinado campo o sobre valores calculados sobre algún campo o conjunto de campos. En este sentido pueden resultar muy útiles funciones como las de la siguiente tabla:

Función	Significado
<i>Izq(Cadena; Número)</i>	Proporciona los <i>Número</i> primeros caracteres de una <i>Cadena</i> .
<i>Der(Cadena; Número)</i>	Proporciona los <i>Número</i> últimos caracteres de una <i>Cadena</i> .
<i>Medio(Cadena; Posición; Número)</i>	Proporciona los <i>Número</i> caracteres de una <i>Cadena</i> , comenzado por una <i>Posición</i> .
<i>Año(Fecha)</i>	Proporciona el año contenido en una variable o campo de tipo fecha/hora.
<i>Mes(Fecha)</i>	Proporciona el mes contenido en una variable o campo de tipo fecha/hora.
<i>Día(Fecha)</i>	Proporciona el día contenido en una variable o campo de tipo fecha/hora.
<i>Hora(Fecha)</i>	Proporciona la hora contenida en una variable o campo de tipo fecha/hora.
<i>Minuto(Fecha)</i>	Proporciona los minutos contenidos en una variable o campo de tipo fecha/hora.
<i>Segundo(Fecha)</i>	Proporciona los segundos contenidos en una variable o campo de tipo fecha/hora.

Al acceder al diálogo de definición de grupos, Access nos muestra las siguientes opciones:

- **Campo/Expresión:** permite identificar los campos o expresiones que generarán los grupos, por ejemplo el campo *Apellido1*. El campo o expresión debe estar referida a los datos de la tabla o consulta en que se base el informe.

- **Orden:** para cada grupo, indica si los inicios de cada grupo deben ir apareciendo de forma ascendente o descendente. Por ejemplo, si agrupamos por fechas, el orden puede ser de fechas ascendentes o descendentes. No se refiere al orden de los registros pertenecientes al grupo, sino a los encabezados de grupo, a los grupos en sí.
- **Encabezado del grupo:** para cada grupo, indica si debe incluirse una sección de encabezado del grupo, que aparecerá justo antes que los registros del grupo.
- **Pie del grupo:** para cada grupo, indica si debe incluirse una sección de pie del grupo, que aparecerá justo después que los registros del grupo.
- **Agrupar en:** permite especificar una forma precisa de generar los grupos. Dependiendo del tipo del campo o expresión por la que se agrupe, podremos seleccionar:
 - **Campos de tipo texto:** se puede seleccionar que se agrupe por el campo/expresión completa o por los n primeros caracteres.
 - **Campos de tipo fecha:** se puede seleccionar que se agrupe por el campo/expresión completa o año, mes, día, hora, minuto, segundo, trimestre, semana, etc...
 - **Campos numéricos:** se puede seleccionar que se agrupe por el número en concreto o por intervalos del número.
- **Intervalo del grupo:** indica cada cuántos valores distintos del criterio de agrupación debe crearse un grupo. Por ejemplo, si se decide agrupar por fechas, y además por semanas, y esta propiedad se establece a 2, se harán grupos para fechas que se diferencien en dos semanas.
- **Mantener juntos:** indica si Access debe disponer todos los registros de un grupo en la misma página (siempre y cuando no ocupen más de una página), comenzando si es necesario una nueva página al principio del grupo.

3. FORMATO DE PAGINA

Como ya sabemos, los informes utilizan como soporte básico el papel y la impresora. Este es un medio con muchas posibilidades de configuración: existen diversos tipos de impresoras e infinidad de tamaños y formas de papel. Los informes deben diseñarse, hasta cierto punto, teniendo en cuenta el papel sobre el que irán impresos.

Esta configuración, junto con otras opciones de presentación se encuentran en la opción *Configurar página* del menú *Archivo*, y debe establecerse durante el diseño del informe (aunque podrá modificarse en el momento de imprimir). Las opciones disponibles de Access 97 son:

- **Impresora** (la predeterminada de Windows o una en concreto).
- **Tamaño, orientación y origen** del papel.
- **Márgenes.**
- **Número de columnas.** Se refiere al número de columnas por página que se utilizarán para listar los datos en la sección de *Detalle*. Además,

se puede especificar la separación entre columnas y la dirección de escritura a lo largo de las columnas.

Es importante establecer correctamente estas opciones. El área disponible para introducir los controles no se ve limitada por el tamaño del papel; sin embargo, a la hora de imprimir, nuestro diseño puede quedar dividido en varias hojas si hemos excedido la superficie disponible (que se puede calcular fácilmente conociendo el tamaño del papel y los márgenes, ya que Access sitúa los controles utilizando centímetros como unidad de medida).

Otro detalle importante de conocer es el ancho de las columnas. Cuando sólo se trabaja a una columna, que es lo más habitual, no existe mayor problema: el ancho de la columna coincide con el ancho de todas las secciones. Sin embargo al trabajar a 2 o más columnas, existen dos posibilidades:

- **Las columnas tienen el mismo ancho que todas las secciones**, aunque realmente sólo se escribe a dos columnas la sección de detalle.
- **Las columnas tienen un ancho fijo determinado**, que no es el mismo que el de todas las secciones, ni siquiera el de la de detalle. Así, los encabezados y pies pueden tener mayor (o menor) ancho que las columnas, lo cual puede dar un aspecto más elegante al informe. (Así se ha hecho en el ejemplo de la figura 4.1)

4. FUNCIONES

Las siguientes funciones pueden resultar interesantes a la hora de confeccionar informes:

Función	Significado
Página	Número de página actual (para numerar las páginas).
Páginas	Número de páginas total en el informe.
Fecha()	Fecha en que se imprime.
Ahora()	Fecha y hora en que se imprime.

TEMA 6: MACROS

1. INTRODUCCION

Una macro es un **conjunto de una o más acciones que cada una realiza una operación determinada**, tal como abrir un formulario o imprimir un informe. Las macros pueden ayudar a automatizar las tareas comunes. Por ejemplo se puede ejecutar una macro que imprima un informe cuando el usuario presione un botón de comando.

Una macro consta de una secuencia de acciones, cada una de las cuales puede realizar una operación de entre un conjunto de operaciones posibles. De forma opcional, para cada acción se puede indicar una condición lógica de manera que la acción sólo se ejecute si la condición se evalúa como verdadera.

2. ACCIONES

La siguiente tabla contiene todas las acciones posibles en orden alfabético, junto con su descripción:

Acción	Descripción
<i>AbrirConsulta</i>	Abre un objeto de tipo consulta en cualquiera de los modos posibles.
<i>AbrirFormulario</i>	Abre un objeto de tipo formulario en cualquiera de los modos posibles.
<i>AbrirInforme</i>	Abre un objeto de tipo informe en cualquiera de los modos posibles, o lo manda a la impresora.
<i>AbrirMódulo</i>	Abre un procedimiento o función de un módulo en vista de diseño.
<i>AbrirTabla</i>	Abre un objeto de tipo tabla en cualquiera de los modos posibles, con opciones especiales para añadir, editar o sólo lectura.
<i>AgregarMenú</i>	Permite añadir una opción al menú
<i>AplicarFiltro</i>	Aplica una condición WHERE o una consulta de selección sobre los datos del formulario activo.
<i>Bip</i>	Emite un sonido.
<i>BuscarRegistro</i>	Busca un registro que contenga un valor en un campo, utilizando las opciones habituales para buscar en Windows.

<i>BuscarSiguiente</i>	Continúa la última búsqueda efectuada.
<i>CambiarNombre</i>	Cambia el nombre de un objeto de tipo tabla, consulta, formulario, informe, macro o módulo.
<i>CancelarEvento</i>	Cancela el evento que activó la macro y la acción que elevó el evento (p.e. la actualización de un registro)
<i>Cerrar</i>	Cierra el objeto (tabla, consulta, formulario, informe, macro o módulo) actual o uno determinado, con opciones de guardar los cambios.
<i>CopiarObjeto</i>	Reproduce el objeto (tabla, consulta, formulario, informe, macro o módulo) indicado en la base de datos (con otro nombre) o en otra base de datos.
<i>CuadroMsj</i>	Muestra un diálogo estándar de Windows, indicando un mensaje, con un icono, sonido y botones estándar.
<i>DesplazarTamaño</i>	Cambia la posición y/o tamaño de la ventana (objeto) activa.
<i>DetenerMacro</i>	Termina la ejecución de la macro.
<i>DetenerTodasMacros</i>	Termina la ejecución de la macro actual y de todas las que se están ejecutando (que invocaron a la actual).
<i>Eco</i>	Establece si se muestran o no los mensajes emitidos por la macro que se está ejecutando, mostrando opcionalmente un mensaje en la barra inferior. Los mensajes de error no se ven afectados.
<i>EjecutarAplicación</i>	Lanza un programa Windows o MS-DOS.
<i>EjecutarCódigo</i>	Ejecuta un procedimiento o función de un módulo (en Visual Basic).
<i>EjecutarComando</i>	Ejecuta un comando de un menú de Access.
<i>EjecutarMacro</i>	Ejecuta una macro. Cuando termine la nueva macro se continúa por la actual.
<i>EjecutarSQL</i>	Ejecuta una consulta SQL (de inserción, borrado, actualización...).
<i>EliminarObjeto</i>	Borra un objeto determinado de la base de datos (tabla, consulta, formulario, informe, macro o módulo), o el seleccionado en la ventana de la base de datos.

<i>EnviarObjeto</i>	Envía un objeto de la base de datos por correo electrónico.
<i>EnviarTeclas</i>	Simula la pulsación de una secuencia de teclas y ejecuta las acciones que se deriven de la pulsación.
<i>EstablecerAdvertencias</i>	Especifica si deben mostrarse mensajes de confirmación antes de determinadas acciones, o se asumirá que se acepta.
<i>EstablecerElementoDelMenú</i>	Activa, desactiva, bloque o desbloquea una opción de un menú.
<i>EstablecerValor</i>	Establece el valor contenido en un control de un formulario o informe.
<i>Guardar</i>	Guarda el objeto activo (tabla, consulta, formulario, informe, macro o módulo) o un objeto determinado.
<i>Imprimir</i>	Imprime el objeto de la base de datos (tabla, consulta, formulario, informe, macro o módulo) activo, con las opciones habituales de rango de páginas, número de copias, etc.
<i>IrAControl</i>	Lleva el enfoque a un control determinado del formulario actual.
<i>IrAPágina</i>	Lleva el enfoque al primer control de la página especificada de un formulario.
<i>IrARegistro</i>	Lleva el enfoque al registro indicado.
<i>Maximizar</i>	Maximiza la ventana activa.
<i>Minimizar</i>	Minimiza la ventana activa.
<i>MostrarBarraDeHerramientas</i>	Muestra u oculta una barra de herramientas.
<i>MostrarTodosRegistros</i>	Elimina los filtros aplicados a los datos de una tabla, formulario o informe, de manera que vuelven a aparecer todos los registros existentes.
<i>NuevaConsulta</i>	Recarga los datos en un control del formulario activo, o recalcula su valor.
<i>RelojDeArena</i>	Cambia el aspecto del cursor del ratón a un reloj de arena o su aspecto predeterminado.
<i>RepintarObjeto</i>	Vuelve a dibujar en pantalla (refrescar) el objeto (tabla, consulta, formulario, informe, macro o módulo) indicado.

<i>Restaurar</i>	Restaura a su tamaño normal una ventana, después de maximizar o minimizar.
<i>SalidaHacia</i>	Exporta los datos de un objeto de la base de datos (tabla, consulta, formulario, informe, macro o módulo) a formato RTF (<i>Rich Text Format</i>), Microsoft Excel, HTML o texto.
<i>Salir</i>	Sale de Microsoft Access, con opciones para guardar los últimos cambios.
<i>SeleccionarObjeto</i>	Permite activar un objeto para a continuación ejecutar otra acción que se aplique sobre el objeto activo.
<i>TransferirBaseDatos</i>	Permite importar/exportar objetos de la base de datos desde/a otras bases de datos, así como vincular tablas con bases de datos externas.
<i>TransferirHojaCálculo</i>	Importa, exporta o vincula los datos de una hoja de cálculo Microsoft Excel.
<i>TransferirTexto</i>	Importa, exporta o vincula los datos de un fichero de texto.