Teoría de Bases de Datos.

- **Introducción a las técnicas de Bases de Datos.**

  El objetivo principal de las bases de datos es el de unificar los datos que se manejan y los programas o aplicaciones que los manejan. Anteriormente los programas se codificaban junto con los datos, es decir, se diseñaban para la aplicación concreta que los iba a manejar, lo que desembocaba en una dependencia de los programas respecto a los datos, ya que la estructura de los ficheros va incluida dentro del programa, y cualquier cambio en la estructura del fichero provocaba modificar y recompilar programas. Además, cada aplicación utiliza ficheros que pueden ser comunes a otras de la misma organización, por lo que se produce una REDUNDANCIA de la información, que provoca mayor ocupación de memoria, laboriosos programas de actualización (unificar datos recogidos por las aplicaciones de los diferentes departamentos), e inconsistencia de datos (no son correctos) si los datos no fueron bien actualizados en todos los programas. Con las bases de datos, se busca independizar los datos y las aplicaciones, es decir, mantenerlos en espacios diferentes. Los datos residen en memoria y los programas mediante un sistema gestor de bases de datos, manipulan la información. El sistema gestor de bases de datos recibe la petición por parte del programa para manipular los datos y es el encargado de recuperar la información de la base de datos y devolvérsela al programa que la solicitó. Cada programa requerirá de una cierta información de la base de datos, y podrá haber otros que utilicen los mismos datos, pero realmente residirán en el mismo espacio de almacenamiento y los programas no duplicarán esos datos, si no que trabajarán directamente sobre ellos concurrentemente. Aunque la estructura de la base de datos cambiara, si los datos modificados no afectan a un programa específico, éste no tendrá por qué ser alterado. Mediante estas técnicas de base de datos se pretende conseguir a través del Sistema Gestor de Bases de Datos(SGBD):

    - INDEPENDENCIA de los Datos: Cambios en la estructura de la Base de Datos no modifican las aplicaciones.
    - INTEGRIDAD de los Datos: Los datos han de ser siempre correctos. Se establecen una serie de restricciones (reglas de validación) sobre los datos.
    - SEGURIDAD de los Datos: Control de acceso a los datos para evitar manipulaciones de estos no deseadas.

- **Definición de Bases de Datos.**

  Es una colección de datos referentes a una organización estructurada según un *modelo de datos* de forma que refleja las relaciones y restricciones existentes entre los objetos del mundo real, y consigue independencia, integridad y seguridad de los datos.

  Lo que debemos tener claro es la diferencia entre Base de Datos y SGBD. La base de datos es el almacenamiento donde residen los datos. El SGBD es el encargado de manipular la información contenida en ese almacenamiento mediante operaciones de lectura/escritura sobre la misma. Además las bases de datos no sólo contendrán las tablas (ficheros) de datos, sino que también almacenará formularios (interfaces para edición de datos), consultas sobre los datos, e informes. El SGBD se encargará de manipular esos datos, controlar la integridad y seguridad de los datos, reconstruir y reestructurar la base de datos cuando sea necesario.

- **Definición de Modelo de Datos.**

  Un modelo de datos es un conjunto de CONCEPTOS y REGLAS que nos llevarán a poder reflejar la estructura de datos y operaciones aplicables sobre ellos de un sistema informático.

- **Introducción al Modelo Relacional.**

  Existen multitud de modelos de datos aplicables para el diseño de bases de datos, pero el modelo relacional es el más usado y extendido; actualmente los SGBD más implantados utilizan este modelo de datos.

La representación gráfica de este modelo es la TABLA.

Alumnos : Tabla

DNI	Apellidos	Nombr	Dirección	CPosta	Provinci	Teléfono
21501300A	Pérez Torregrosa	Javier	C/. Mayor, 5	03005	Alicante	965229236
21432501T	Rodríguez Esteban	Luis	Avda. América, 3	03008	Alicante	965122946
21405600D	Llum Piqueras	Ana	C/. Ochando, 15	03012	Alicante	965220277
21345654H	Zabala Puig	Eva	C/. Ríos, 56	03003	Alicante	96515545
21304567T	Rodríguez Ybarra	Esther	Avda. Aguilera, 5	03007	Alicante	965923342

Una tabla se compone de FILAS y COLUMNAS. Las FILAS se corresponden con los REGISTROS y las columnas se corresponden con los CAMPOS.

Un CAMPO será la unidad mínima de información.

Alumnos : Tabla

DNI	Apellidos	Nombr	Dirección	CPosta	Provinci	Teléfono
21501300A	Pérez Torregrosa	Javier	C/. Mayor, 5	03005	Alicante	965229236
21432501T	Rodríguez Esteban	Luis	Avda. América, 3	03008	Alicante	965122946
21405600D	Llum Piqueras	Ana	C/. Ochando, 15	03012	Alicante	965220277
21345654H	Zabala Puig	Eva	C/. Ríos, 56	03003	Alicante	96515545
21304567T	Rodríguez Ybarra	Esther	Avda. Aguilera, 5	03007	Alicante	965923342

A partir de éste se formarán los REGISTROS.

Alumnos : Tabla

DNI	Apellidos	Nombr	Dirección	CPosta	Provinci	Teléfono
21501300A	Pérez Torregrosa	Javier	C/. Mayor, 5	03005	Alicante	965229236
21432501T	Rodríguez Esteban	Luis	Avda. América, 3	03008	Alicante	965122946
21405600D	Llum Piqueras	Ana	C/. Ochando, 15	03012	Alicante	965220277
21345654H	Zabala Puig	Eva	C/. Ríos, 56	03003	Alicante	96515545
21304567T	Rodríguez Ybarra	Esther	Avda. Aguilera, 5	03007	Alicante	965923342

Para entender estos conceptos partamos de un típico fichero de ALUMNOS. Cada una de las fichas incluidas en él sería un registro y cada apartado de información (DNI, nombre, apellidos, dirección, teléfono, …) que se rellena referente al alumno sería un campo.

Como norma general, una tabla siempre dispondrá de un campo o conjunto de ellos denominado CLAVE PRIMARIA, que permitirá identificar de forma única cada registro de una tabla. Se podrán aplicar INDICES sobre las tablas, lo que permitirá acceder a la información más rápidamente. Las tablas las podremos RELACIONAR de forma que se evitará redundancia de información y la información será más correcta. Estas relaciones se establecerán entre uno o varios campos de una tabla (CLAVE AJENA) “contra” la clave primaria de otra tabla, por lo que una vez establecida la relación, los valores que se introduzcan deberán existir en la tabla relacionada.

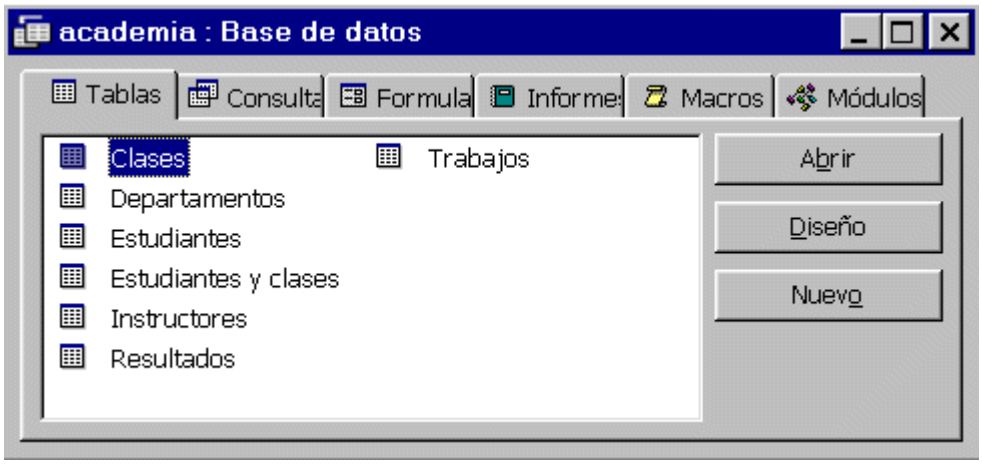
- **Access y las Bases de Datos.**

Access no es una base de datos es un sistema gestor de bases de datos que se apoya sobre motor de Bases de Datos MOTORJET que es el encargado de interactuar con la base de datos (entiende cómo

acceder a la información y manipularla), sirviendo a Access la información que necesite para que éste último interactúe con el usuario.

Las bases de datos que utiliza Access tienen la extensión MDB (por ejemplo, registro.mdb, fabrica.mdb, etc …)

Como ya dijimos antes la base de datos no sólo contendrá las tablas si no que también contendrá los diferentes objetos necesarios para manipular información.



Estrictamente hablando, una base de datos es una colección de información relacionada con un asunto, tema o actividad específicos. Por ejemplo, la guía de teléfonos es una base de datos, así como su agenda o el catálogo de tarjetas de una biblioteca. En Microsoft Access, es posible almacenar información, pero se pueden hacer muchas más cosas. Por ejemplo, si mantiene una lista con todo el personal de la Universidad de Alicante, puede:

- Imprimir una lista de todos aquellos empleados que cumplan una determinada condición.
- Ordenar los empleados por el centro o departamento al que pertenecen.
- Crear un sencillo formulario de introducción de registros para que, cualquiera pueda utilizarlo sin ningún tipo de problema.

Access es un potente sistema de administración de bases de datos relacionales. Las bases de datos de Access son documentos combinados donde se divide la información por parcelas de objetos especializados. Lo primero que hay que hacer es crear un archivo para la base de datos. Dicho archivo contiene todo lo que se cree para la base de datos, no sólo los datos sino también los formularios personalizados y los índices.

Tablas:

El núcleo de cualquier base de datos son sus tablas. Toda los datos que vayamos introduciendo en la base de datos se irán almacenando en la tabla o tablas correspondientes. Normalmente, se crea una tabla para cada tipo de datos de los que se compone la base de datos, así por ejemplo tendríamos tablas para personal, centros, registros, … en la que nos interesa tener almacenada información diversa relacionada con estos conceptos como puede ser nombre, extensión telefónica, años de antigüedad, email para personal; dirección postal, nombre del director, teléfono para centro; fecha de entrada/salida, cargo para los registros; etc.

La apariencia física de una tabla en Access es cómo una hoja de cálculo, donde la información podemos verla distribuida en filas, columnas y celdas. Las filas o Registros almacenan toda la información perteneciente a un elemento de la tabla (por ejemplo, un centro). Las columnas o Campos contienen la información relativa a un determinado tipo de información dentro de la tabla, por tanto, toda la información almacenada en un campo va a ser del mismo tipo (por ejemplo un campo puede ser el precio de un recibo, donde almacenará datos de tipo monetario). En la intersección de los campos y las filas se encuentran las celdas.

Los archivos de bases de datos pueden tener muchas tablas y aunque se crean como elementos independientes, pueden crearse relaciones entre distintas tablas para recuperar datos de ellas mediante una consulta, formulario o informe.

Formularios:

Para facilitar su almacenamiento, todos los datos que se introducen en una base de datos acaban estando en una tabla. Es posible introducir información directamente en una tabla, aunque ello resulte un poco incómodo. Lo normal es crear un formulario especial en pantalla mediante el que se introducen los datos de una manera sencilla y cómoda. La apariencia física de un formulario es parecido a una hoja en la que se rellenan a mano los espacios en blanco, como por ejemplo una solicitud de empleo. Access vincula los formularios a las tablas y almacena en estas últimas la información que introduzca en los primeros.

Informes:

Mientras que los formularios están pensados para su utilización en pantalla, los informes se han diseñado para imprimirse. Son colecciones de datos con un formato específico organizadas siguiendo sus especificaciones. Los informes se utilizan primordialmente para presentar, resumir e imprimir los datos en la forma que resulte más apropiada para cada proyecto. Se pueden crear informes que incorporen cálculos basados en los datos de las tablas para mostrar resultados totales o promedios, o bien para generar e imprimir catálogos, listas de nombres y direcciones o etiquetas postales.

Los informes pueden crearse en cualquier momento, no es necesario planificarlos antes de crear la base de datos.

Consultas:

Las consultas se utilizan para localizar y recuperar los datos específicos que cumple unas determinadas condiciones especificadas por el usuario. Las consultas permiten, además actualizar varios registros al mismo tiempo, así como realizar operaciones de muy diversa índole con los datos almacenados en las tablas.

Macros:

Son la forma que brinda Access para la automatización de la programación,.. Con ella el usuario puede utilizar funciones predefinidas sin tener que hacerlas él mismo. Existe una gran variedad de estas funciones y la combinación de las mismas aumenta el poder de las macros. Las macros y módulos se utilizan para la creación de funciones específicas y la personalización de un sistema.

Módulos:

Cuando un usuario ha alcanzado cierto dominio sobre el manejo de una base de datos, probablemente desea crear funciones de mayor complejidad. Para ello Access proporciona opciones de programación, los módulos, que son rutinas de programación creadas por el usuario y que pueden ser llamados para efectuar una acción específica.

Controles:

Los elementos en un formulario o informe que presentan e imprimen los datos se llaman controles. Con un control se pueden presentar datos en un campo, resultados de un cálculo, palabras para el título, o bien, gráficas, dibujos u otros objetos, e incluso otro formulario e informe. Un control se puede colocar dentro de un formulario o informe.

Como resumen podemos decir que la parte principal son las tablas que contienen los datos; de éstas se obtienen las consultas. De las consultas y/o tablas se crean los formularios y los informes. En cuanto a las macros y módulos, sirven de apoyo para realizar funciones más especializadas.

Cómo planificar la base de datos:

Antes de crear una base de datos, debería hacerse las siguientes preguntas:

- ¿Qué datos deseo almacenar y cuál es la mejor forma de hacerlo?
- ¿Cómo introduzco los datos relativos a mi empresa o mi afición?. Esto indica los formularios que va a necesitar.
- De qué entidades nos interesa tener almacenada información. Esto indica las tablas que va a necesitar.
- ¿Qué entidades están relacionadas con otras dentro de nuestra base de datos?. Nos indica las relaciones entre tablas a la hora de realizar las consultas a la base de datos para obtener sólo la información que nos interese.
- ¿Qué información de mi empresa o afición deseo mostrar?. Esto indica los tipos de informes que va a necesitar.

Normalizar la base de datos:

Cuando la organización de las tablas es deficiente, se dice que no está normalizada. Hay ciertas reglas que indican cómo se deben almacenar las tablas en una base de datos relacional. A estas reglas se las denomina de normalización de datos. Las reglas más importantes de normalización son:

- *Evitar información repetida*: Supongamos que desea tener almacenado los registros de entrada y de salida que le llegan, y además desea incluir la persona que lo envía. Si utilizara una sola tabla, tendría que repetir el nombre del empleado cada vez que le llega un registro de la misma persona. Además un cambio en el nombre de un registro supondría buscar en todas las transacciones el nombre de la persona a cambiar. Si cree que existe la posibilidad de que en el futuro termine repitiendo datos en la tabla, piense ahora cómo puede dividir la información que se repetirá en la misma.

- *Evitar repetición de datos*: En ningún caso debemos tener dos valores iguales en una tabla, todos los registros de una tabla deben ser siempre distintos en por lo menos un campo. No obstante cabe tener presente que todos los campos entre dos registros de una misma tabla no tienen que ser necesariamente distintos.

- **Limitaciones de una base de datos en Access.**

  Cuando se diseñe una base de datos en Access se deberán tener en cuenta las siguientes limitaciones:

  - Una tabla podrá contener un máximo de 255 campos.
  - Una tabla podrá contener un máximo de 32 índices.
  - Un índice de campo múltiple puede tener hasta 10 columnas. La suma de las longitudes de las columnas no puede exceder de 255 bytes.
  - Un registro de una tabla, excluyendo campos memo y objetos OLE, no puede exceder de 2 Kbytes aproximadamente.
  - Un campo Memo puede almacenar hasta 1 Gbyte de información, pero sólo serán visualizables desde formulario u hoja de datos un campo memo con longitud superior a 32 Kbytes.
  - Un objeto OLE puede ser de hasta 1 Gbyte de tamaño.
  - No existe un límite sobre el número de registros de una tabla, pero una base de Datos en Access no puede ocupar más de 1Gbyte, por lo que para bases de datos más extensas habrá que crear varias y efectuar vinculación entre ellas.

- **Tipos de Datos en Microsoft Access**

Hemos visto que para almacenar los datos de nuestra organización utilizaremos las tablas. Una tabla estará compuesta por una serie de campos que contendrán un tipo determinado de valores con una longitud, y estos pueden ser:

- **Texto**: Este tipo de campo se usa cuando el campo va a contener caracteres de una extensión más o menos fijas (apellidos, nombre, direcciones, poblaciones, cualquier tipo de descripción …) y también se utilizan cuando se mezclan letras y números tal que en NIF, CIF, … Su longitud predeterminada es de 50 caracteres. Almacena cualquier carácter pudiendo ser de hasta 255 caracteres o con la longitud establecida en la propiedad Tamaño del Campo.
  Propiedades de un campo texto:
  - Tamaño del Campo: Valor numérico que especifica la longitud del campo. No se permitirá la introducción de mayor número de caracteres para ese campo.
  - Formato: Permite especificar la forma en que se presentarán los datos. Es posible crear formatos de Texto y Memo personalizados mediante los siguientes símbolos.

    Símbolo Descripción
    @ Se necesita un carácter de texto (ya sea un carácter o un espacio).
    & No se necesita un carácter de texto.
    < Convertir todos los caracteres a minúsculas.
    > Convertir todos los caracteres a mayúsculas.

    Como ejemplos podríamos citar:

Valor	Datos	Muestra
@@@-@@-@@@@	465043799	465-04-3799
@@@@@@@@@@	465043799	465043799
>	davolio	DAVOLIO
<	JAVIER	javier
	Juan	juan
@;"Desconocido"	Valor Null	Desconocido
	Cadena de longitud cero	Desconocido
	Cualquier texto	El mismo texto introducido

  - Máscara de Entrada: Obliga a que los datos introducidos en un campo se ajusten a un formato determinado. Puede definir una máscara de entrada mediante los siguientes caracteres:

    Carácter Descripción
    0 Dígito (0 a 9, entrada obligatoria, signos más [+] y menos [–] no permitidos).
    9 Dígito o espacio (entrada no obligatoria, signos más y menos no permitidos).
    # Dígito o espacio (entrada no obligatoria; los espacios se muestran en blanco en el modo Edición, pero se eliminan cuando se guardan los datos; signos más y menos están permitidos).
    L Letra (A a Z, entrada obligatoria).
    ? Letra (A a Z, entrada opcional).
    A Letra o dígito (entrada obligatoria).
    a Letra o dígito (entrada opcional).
    & Cualquier carácter o un espacio (entrada obligatoria).
    C Cualquier carácter o un espacio (entrada opcional).
    . , : ; - /
    Marcador de posición decimal y separadores de miles, hora y fecha (el carácter depende del valor del cuadro de diálogo Propiedades de Configuración regional en el Panel de control de Windows).
    < Hace que todos los caracteres se conviertan a minúsculas.
    > Hace que todos los caracteres se conviertan a mayúsculas.

!	Hace que la máscara de entrada se muestre de derecha a izquierda, en lugar de mostrarse de izquierda a derecha. Los caracteres introducidos en la máscara siempre se rellenan de izquierda a derecha. Puede incluir el signo de exclamación en cualquier lugar de la máscara de entrada.
\	Hace que el carácter siguiente se muestre como un carácter literal (por ejemplo, \A se muestra sólo como A).

El establecimiento de la propiedad MáscaraDeEntrada a la palabra "Contraseña" Crea un control de entrada de contraseña. Cualquier carácter introducido en el control se almacena como el carácter pero se muestra como un asterisco (*). Se utilizará la máscara de entrada de Contraseña para impedir que se muestren los caracteres escritos en la pantalla.

Como ejemplos podemos citar:

Máscara de entrada	*Valores de ejemplo*
(000) 000-0000	(206) 555-0248
(999) 999-9999	(206) 555-0248 ( ) 555-0248
(000) AAA-AAAA	(206) 555-TELE
#999 2000	–20
>L????L?000L0	GREENGR339M3 MAY R 452B7
>L0L 0L0	T2F 8M4
00000-9999	98115- 98115-3007
>L<??????????????	María Brendan
SSN 000-00-0000	SSN 555-55-5555
>LL00000-0000	DB51392-0493

- Título: Especifica el nombre que se utilizará en Formularios e Informes para ese campo.
- Valor Predeterminado: Valor que tomará el campo por omisión al agregar un nuevo registro a la tabla.
- Regla de Validación: Determina las condiciones que debe cumplir el dato que se pretende introducir en el campo para ser aceptado. Podríamos decir que un valor esté entre un determinado rango de valores, o que sea mayor que el valor de otro campo, …, es decir se evaluará una expresión que si su resultado es cierto, el campo introducido tendrá un valor correcto.
- Texto de Validación: Texto que se mostrará si no se pasa la Regla de Validación especificada en la propiedad anterior.
- Requerido: Indica que es obligatorio introducir un dato en el campo.
- Permitir Longitud cero: Permite que se guarden cadenas de longitud 0 en el campo, es decir, permite valores “”.
- Indexado: Determina si este campo será un índice de la tabla (para acelerar las búsquedas).

- **Memo:** Este tipo se utiliza cuando no se sabe la longitud que tendrá el campo, es decir, notas al efecto, comentarios, observaciones, … Tendrá una longitud máxima de 65535 caracteres. Las propiedades inherentes a este tipo de dato son Formato, Titulo, Valor Predeterminado, Regla de Validación, Texto de Validación, Requerido y Permitir Longitud cero, y la explicación de éstas coincide con las explicadas para el campo Texto.
- **Numérico:** Este tipo está pensado para incluir aquellos números con los que se vayan a efectuar cálculos (cantidades). Por ejemplo el Código Postal se suele especificar como Texto ya que sobre él no se efectúan cálculos. Permite la representación de dígitos del 0 al 9 y combinación de ellos. Con él podríamos representar información tal que ingresos, gastos, sueldos, edades, …
  Propiedades de un campo Numérico:

- Tamaño del Campo: Número máximo de datos que puede almacenar en un campo.

Configuración	*Descripción*
Byte	Almacena números entre 0 y 255 (sin fracciones).
Entero	Almacena números entre –32768 y 32767 (sin fracciones).
Entero Largo	Almacena números entre –2147483648 y 2147483648 (sin fracciones).
Simple	Número de coma flotante de 4 bytes que contiene valores que van desde $–3,4 \times 10^{38}$ hasta $3,4 \times 10^{38}$.
Doble	Número de coma flotante de 8 bytes que contiene valores que abarcan entre $–1797 \times 10^{308}$ y 1797×10^{308}.
ID de réplica	Identificador único global de 16 bytes (GUID).

- Formato: Establece forma en que los datos serán presentados. No se cambia la manera en que se almacenan los datos, pero sí se influye en la manera de visualizarlos.

Configuración	*Descripción*
Numérico General	Es el valor predeterminado(sin puntos ni símbolos de monedas; los lugares decimales mostrados dependen de la precisión de los datos. Presenta el número tal y como se introdujo.
Moneda	Usa el separador de millares y la moneda (el tipo de moneda depende de la configuración de Windows/Configuración Regional). Por defecto los Lugares Decimales son 2, y podrá ser modificado.
Fijo	Presenta por lo menos un dígito. La configuración de la propiedad lugares decimales es 2, siendo posible alterarla.
Estándar	Utiliza el separador de miles, siendo los lugares decimales predeterminados, 2.
Porcentaje	Multiplica el valor por 100 agregando un signo de porcentaje. Los lugares decimales por defecto son 2.
Científico	Usa la notación científica estándar, es decir, si se introdujese 1000, se visualizaría 1,00E+03.

También se pueden crear formatos numéricos de usuario usando estos códigos:

Carácter	*Indicación*
,	Separador Decimal
.	Separador de Millares
0	Marcador de dígitos. Muestra el dígito o el 0.
#	Marcador de Dígitos. Muestra el dígito o nada.
$	Muestra el carácter literal '$'.
%	Valor se multiplica por 100 y se agrega símbolo de porcentaje.
E- o e-	Notación científica con exponentes negativos.
E+ o e+	Notación científica con exponentes positivos.

- Lugares Decimales: Para los tipos de datos numérico y moneda, podemos especificar el número de lugares decimales visualizados por Access. Predeterminadamente Access controla estos lugares Automáticamente, pero permite especificar una precisión de 0 a 15.
- Máscara de Entrada: Obliga a que los datos introducidos en un campo se ajusten a un formato determinado. Los caracteres utilizables coinciden con los especificados para máscaras de entrada para tipo de datos texto.

Las propiedades Título, Regla de Validación, Texto de Validación, Requerido e Indexado ya fueron explicadas anteriormente para el tipo de dato texto, y su aplicación es exacta en este tipo de dato.

- **Fecha/Hora:** Almacena fechas y horas en diferentes formatos (fechas de nacimiento, de compra, hora de compra, …, hora salida, hora llegada, …). Las únicas propiedades que varían en función del tipo de dato respecto de los anteriores son:

    - Formato: Los diferentes formatos son

Configuración	*Descripción*
Fecha General	Valor Predeterminado: En EEUU: mm/dd/aa hh:mm:ss AM/PM En UK: dd/mm/aa hh:mm:ss
Fecha Larga	Miércoles, 15 de Diciembre de 1999
Fecha Mediana	15-Dic-99
Fecha corta	15/12/99
Hora Larga	5:30:20 PM
Hora media	5:30 PM
Hora Corta	17:30

- **Moneda:** Se utiliza para cantidades numéricas que son monetarias, es decir, campos tales que importe, precio, totales a pagar, … Tiene las mismas propiedades que los campos Numéricos.
- **Autonumérico:** El tipo de datos Autonumérico es un tipo especial de datos, ya que Access incrementa su valor de manera automática cada vez que se añade un nuevo registro a la tabla. Las propiedades de este tipo son Tamaño del Campo, Formato, Título, Indexado que son comunes a las explicadas para los otros tipos, pero la importante aquí es la propiedad Nuevos Valores (que podrá ser Incrementalmente si los valores se van añadiendo en secuencia 1 a 1, o Aleatoriamente, el cual para cada nuevo registro le otorgará un valor aleatorio al campo de este tipo).
- **Sí/No:** Son campos lógicos que sólo pueden representar 2 estados: Sí/No, Activado o Desactivado, Verdadero o Falso. Los formatos podrán ser Sí/No, Activado/Desactivado, Verdadero/Falso. El resto de propiedades se tratan al igual que los anteriores.
- **Objeto OLE:** Este tipo de datos se emplea para la vinculación, incrustación de objetos en tablas, formularios e informes. Sus únicas propiedades son Título y Requerido. Para insertar un objeto, desde campo de formulario u hoja de datos, botón derecho e Insertar Objeto.
- **Hipervínculo:** Está relacionado con la inclusión de vínculos sobre archivos y direcciones de páginas web, así como a direcciones de correo electrónico, servidores FTP, gopher, …, es decir, acceso a cualquier URL.
- **Asistente para Búsquedas:** Crea un campo que permite elegir un valor de otra tabla o de una lista de valores mediante un cuadro combinado. Al elegir esta opción se inicia el asistente para búsquedas, que permite la creación de campos de búsqueda.

- **<u>Construcción de EXPRESIONES en Access.</u>**

Como hemos visto en los tipos de datos que soporta Access existen ciertas propiedades tal que la Regla de Validación que se han de cumplir para que el dato sea correcto. Además de estas reglas de validación a la hora de efectuar un filtro sobre un formulario o un criterio sobre una consulta deberemos especificar una EXPRESION que deberá ser validada para obtener los resultados.

Para poder construir expresiones disponemos de una serie de operadores que a continuación pasamos a describir:

- Operadores Aritméticos:
    - + : Suma
    - - : Resta
    - * : Multiplicación
    - / : División
    - \ : División entera.
    - ^: Elevado a

- mod : Resto de un número

- Operadores de “Comparación”:
  - < : Menor que
  - > : Mayor que
  - <= : Menor o igual que
  - >= : Mayor o igual que
  - = : Igual
  - EN(IN): Determina si un valor es igual a los valores de una lista.
    Ejemplo: regla validación para campo IVA –> in (7,16,28).
  - ENTRE(BETWEEN): Determina si valor está comprendido entre rango de valores.
    Ejemplo: regla validación para campo edad –> ENTRE(18 y 26). La edad tendrá que estar comprendida entre 18 y 26.
  - COMO(LIKE): Comprueba que un campo texto o memo coincide con un modelo determinado. Con este operador se podrán utilizar los siguientes comodines:
    - * : Sustituye cualquier patrón de caracteres. Ejemplo: “Car*”, “C*”. Se buscarían los que empezasen por Car o por C, respectivamente. Se utiliza para definir caracteres iniciales, finales, o cadenas incluidas en otras que no coinciden con ninguno de los caracteres del modelo.
    - ? : Sustituye carácter por cualquiera. Ejemplo: “C?a”. Se buscarán cadenas que contengan como primera letra la ‘C’, como tercera la ‘a’, y la segunda podrá ser cualquiera.
    - # : Este comodín especifica que podrá ser cualquier número.
  - ES NULO(Is Null): Se validará si campo contienen valor NULO. Un campo tiene valor nulo cuando no se ha especificado ningún valor para él. En caso contrario aun siendo de longitud 0 ya no es valor nulo.

- Operadores LOGICOS:

  - Y (AND): Utilizado para “unir” varias expresiones.simples. Se deberán cumplir todas para que se valide la regla o criterio especificado.
    Ejemplo: edad >= 16 Y sexo=”M”. Se deberá cumplir que la edad sea mayor de 16 y el sexo sea M. Si no se cumpliera cualquiera de las dos, no se validaría expresión.
  - O (OR): Utilizado para “unir” varias expresiones simples. La expresión conjunta se validará siempre que se cumpla alguna de ellas. Con el ejemplo anterior edad>= 16 O sexo=”M”, con que se cumpla una de las dos la expresión se evaluará como cierta.
  - NEGADO (NOT): Se evalúa a cierto si no se verifica la expresión simple que contiene.

Expresiones Simples:

Edad >= 26
Fechaingreso entre 16-10-1999 y 21-11-1999
Sexo=”M”
Pagado=Sí

Expresiones compuestas:

Edad>16 y pagado=Sí
(Edad>16 y pagado=Sí) O fechaingreso in (12-10-1999,16-11-1999)
negado(edad>=16 y pagado=Sí)

Conforma vayamos explicando las reglas de validación y diferentes criterios de filtro o consulta iremos viendo más ejemplos.

- **Detección de Relaciones en una base de datos.**

Existen varias formas de relaciones entre tablas según participen los registros de cada una de ellas en la relación. Este curso, al ser de iniciación, no va a tocar las relaciones teóricamente ni va a

especificar los diferentes tipos que nos podemos encontrar; sólo se pretende hacer ver al alumno la necesidad de crear bases de datos sin redundancia de información y en la que la información sea correcta. Partimos de la base de que el alumno requiere almacenar información sobre algo específico y que lo plasmará inicialmente en una única tabla.

Ejemplo:
En la secretaría de nuestro centro queremos tener una base de datos que contenga la información referente a los alumnos matriculados, de los que se quiere saber sus datos personales (NIF, apellidos, nombre, número de cuenta bancaria), las asignaturas en las que se ha matriculado y departamento al que éstas pertenecen (pueden ser de otros departamentos debido a los créditos de libre configuración) , así como los créditos que otorga la asignatura al plan de estudios del alumno, y la nota que éste ha obtenido.

Tras este enunciado lo normal es crear una tabla que contenga todos los campos que queremos almacenar y esta quedaría de esta forma:

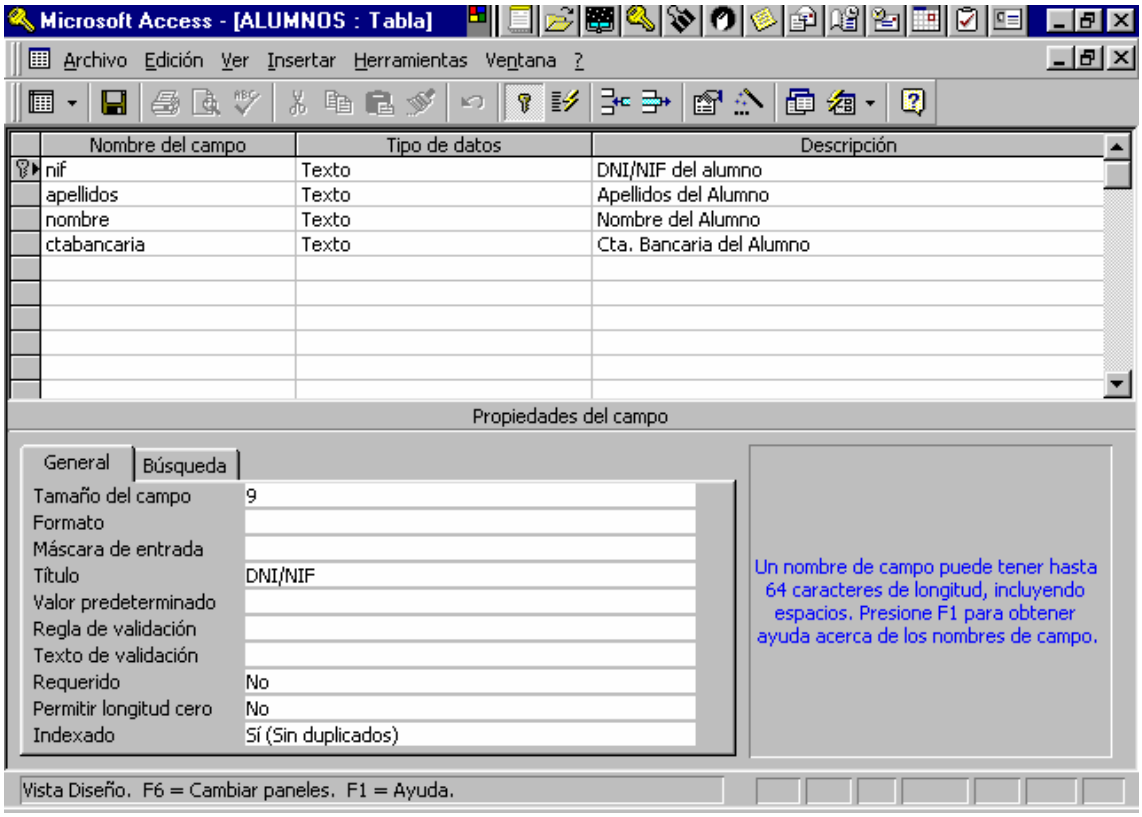
DNI/NIF	Apellidos	Nombre	Núm. Cuenta Bancaria	Asignatura	Cr	Departamento	Cal
21500302F	López Huertas	José Luis	12004500620304057473	Historia del Arte	6	Historia Medieval y Moderna	6
21500302F	López Huertas	José	12004500620304057473	Introducción a la Filosofía del	4	Filosofía del Derecho	8
22400125H	Martínez Luján	Vicente	21006000101347564646	Microinformática	3	Informática y Computación	3
23500218T	Llinares Such	Diego	58665974595859898989	Microinfomática	5	Informática y Computación	7
23500218T	Llinares Such	Diego	83488438934493889348	Literatura Valenciana	6	Filología Catalana	4
22400125H	Martínez Luján	Vicente	21006000101347564646	Hiustoria del Arte	6	Historia del Arte	9

Tras introducir unos cuantos registros ya estamos en disposición de empezar a ver deficiencias en nuestra base de datos de acuerdo a lo indicado a lo largo de este tema con relación a la integridad y redundancia de la información:

1. Vemos que para un mismo alumno (DNI: 21500302F) el nombre ha variado por lo que uno de los registros está erróneo. (Inconsistencia de Datos).
2. Vemos que podríamos tener alumnos con un mismo DNI y eso no es posible. (necesidad de CLAVE PRIMARIA).
3. También existe un alumno (DNI: 23500218T) en el que su número de cuenta varía y nosotros queremos que la matrícula de un alumno se cobre a un solo número de cuenta bancaria. (Inconsistencia de Datos).
4. La asignatura Microinformática en uno de los registros ha sido mal codificada por lo que al hacer referencia a ella, los registros que no incluyan exactamente la asignatura no se tendrán en cuenta. (Inconsistencia de Datos).
5. Cada vez que un alumno se matricule de una asignatura tenemos que codificar de nuevo todos sus datos personales lo que provoca una REDUNDANCIA de información innecesaria y además hace más fácil la equivocación al tener que duplicar multitud de registros y teniendo que teclear esa información en cada uno de ellos. Lo mismo ocurre con las asignaturas de las que siempre tenemos que volver a introducir los créditos que otorga, siendo estos fijos y no dependientes en ningún caso de la matrícula de un alumno en ella.
6. Vemos asignaturas que son las mismas mal codificadas en diferentes registros (Inconsistencia de Datos).
7. La asignatura Historia del Arte pertenece a departamentos diferentes. (Información Incorrecta).
8. Vemos que siempre tenemos que especificar TODOS los caracteres del título de la asignatura, cada vez que un alumno se matricula, además de replicar todos los datos de la misma para cada uno que se matricule. (NECESIDAD DE CLAVE PRIMARIA y REDUNDANCIA). Esto se produce igual para todos los departamentos a codificar.

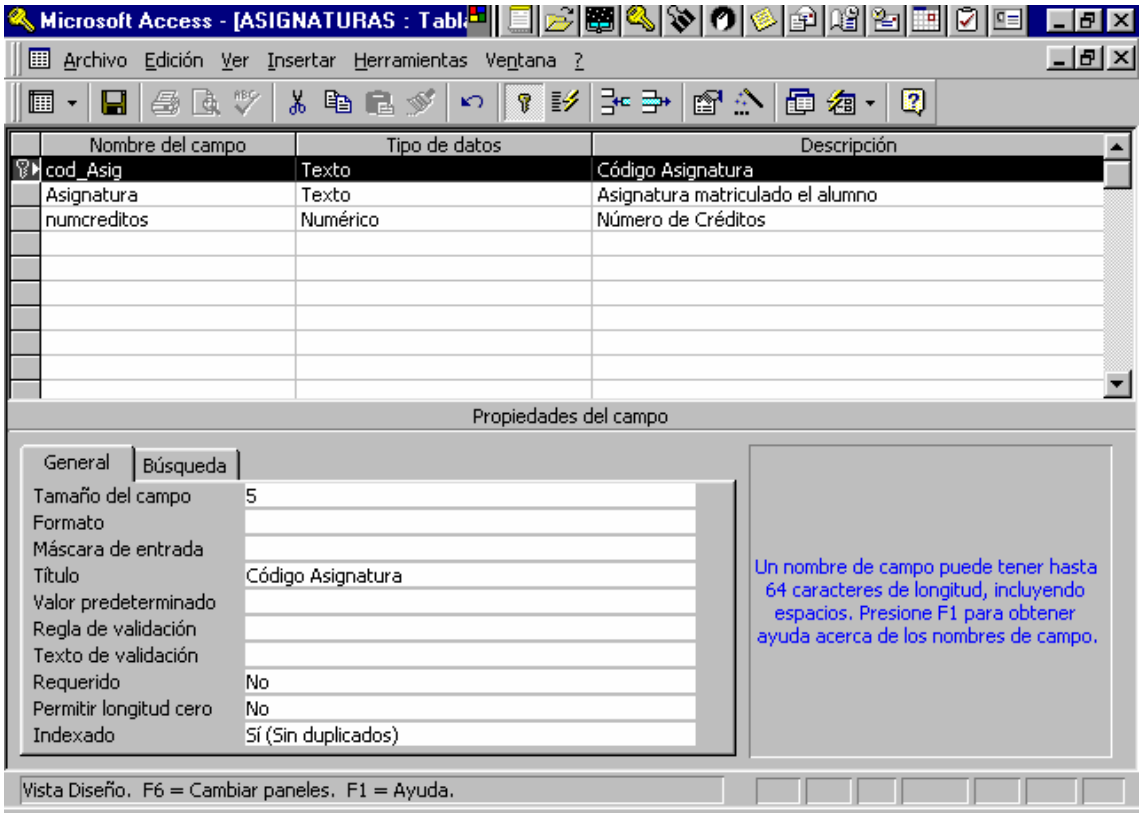
Estamos en disposición de separar nuestra tabla en varias que tendrán una relación. Una manera de ver qué campos compondrán las diferentes tablas es ver las dependencias entre ellos.
En nuestro ejemplo tratamos con matrículas de alumnos con la cual hemos construido la tabla principal (la cual contiene errores). Estas matrículas constan de un alumno que se matricula de una asignatura, la cual pertenece a un departamento.

De los alumnos podemos observar que podemos identificar unívocamente a cada uno de ellos por su NIF, y que con él podremos hacer referencia al resto de sus datos. Y así evitaremos tener que duplicar la información referente a él. Además está claro que los datos del alumno no varían en diferentes matriculaciones en una asignatura. Son siempre los mismos. Así que crearemos una tabla ALUMNOS que contendrá los datos referentes a él y cuyo diseño será como el que sigue:

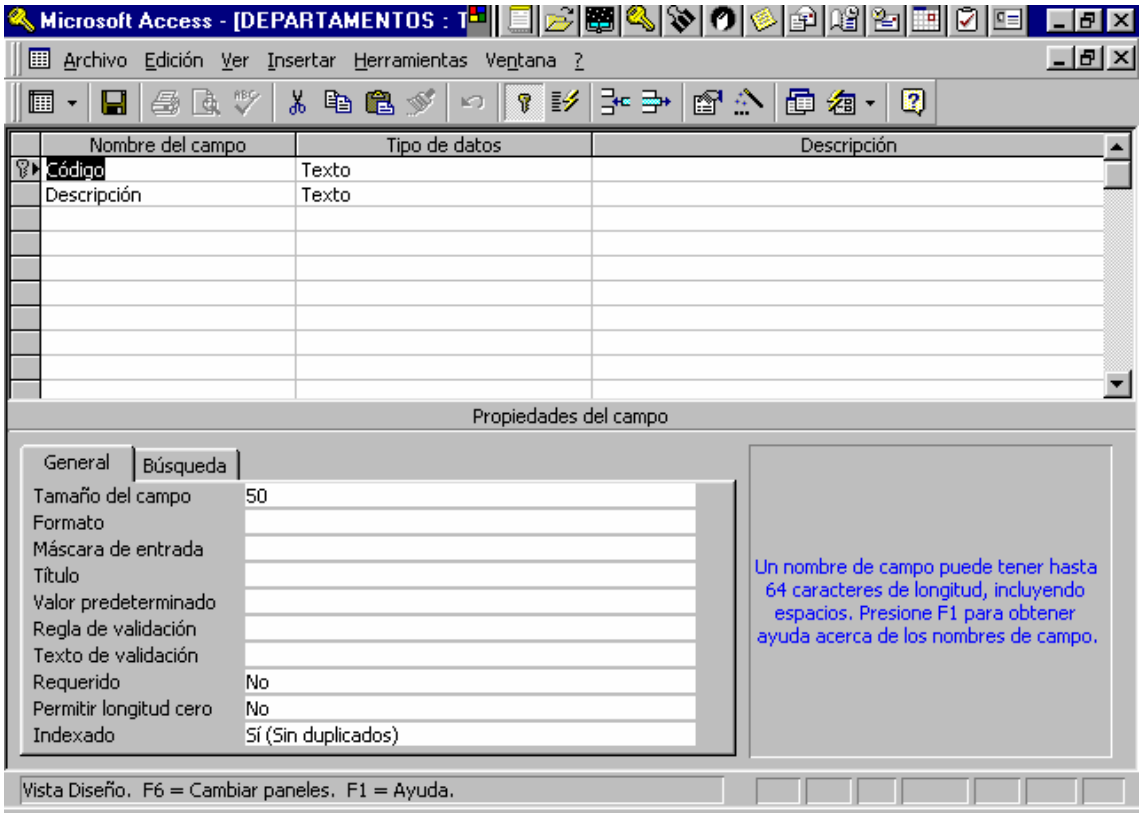


De las asignaturas podemos ver que para cada una de ellas tenemos que especificar demasiados caracteres para hacer referencia a ella así que lo más indicado es poner un campo código mucho más corto de extensión como clave primaria que nos permitirá identificar por un único código cada una de las asignaturas y que el número de créditos de una asignatura siempre es fijo y además depende de la asignatura y no de la matrícula de un alumno en ella.

La tabla ASIGNATURAS quedará como sigue:



Con los Departamentos ocurre lo mismo que con las asignaturas. Necesitarán de un identificador clave que nos proporcionará consistencia de datos.



En este momento tenemos las tablas “maestras” ya generadas, es decir, tablas que contienen los datos base con los que nosotros trabajaremos. Ahora se trata de establecer las relaciones de la tabla MATRICULAS respecto a los alumnos, asignaturas y departamentos que conforman una solicitud de

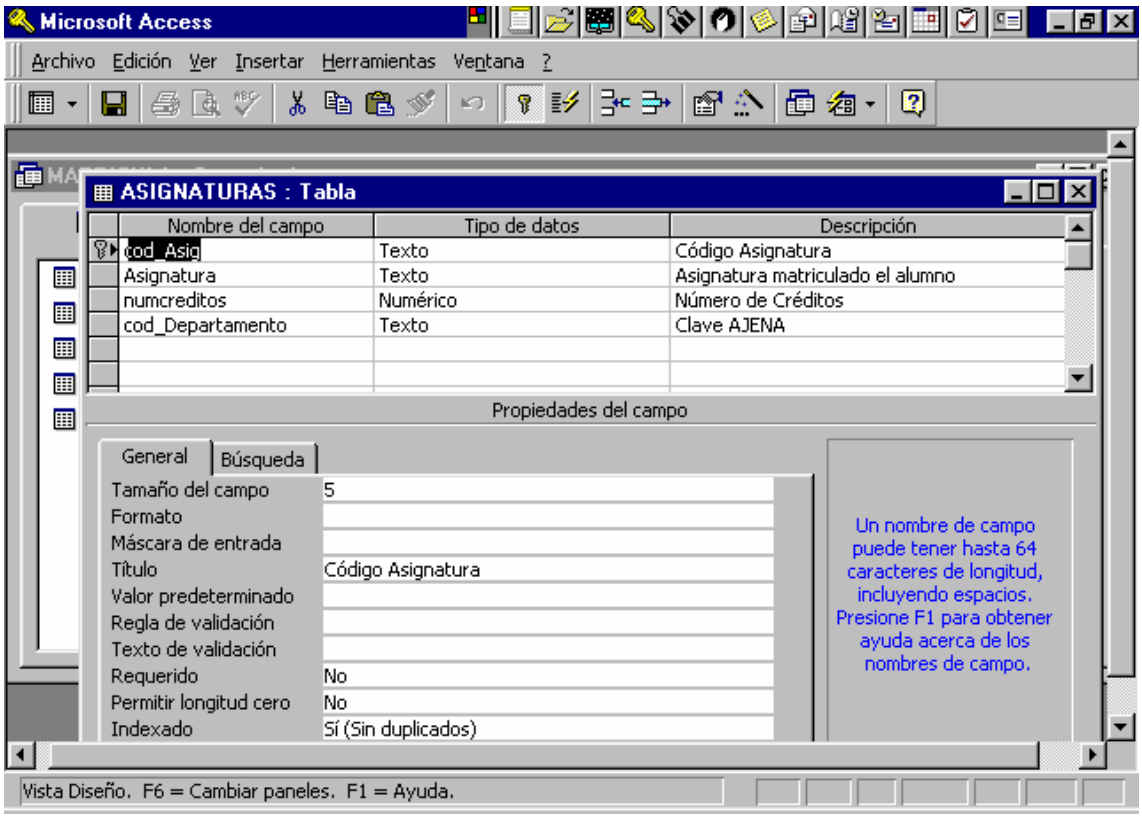
matrícula de un alumno a una asignatura. Como hemos dicho antes no vamos a entrar en la especificación de las diferentes relaciones que pueden surgir en el diseño de una base de datos, pero se hace inevitable decir que las relaciones que vamos a manejar son conocidas como relaciones **1 a muchos**, es decir, un registro de una tabla se puede corresponder con muchos de la tabla con la que está relacionada. En este caso podemos ver:

- Un alumno podrá hacer varias matriculaciones.
- Una asignatura podrá formar parte de varias matriculaciones.
- Un Departamento puede estar formado por varias asignaturas.

Lo especificado anteriormente nos indica que en nuestra base de datos existen tres relaciones.:

- Entre MATRICULAS y ALUMNOS. Para ello dejaremos en la tabla matrícula el Nif del alumno como CLAVE AJENA (campo cuyo valor o es NULO o coincide con el valor de la clave primaria de la tabla a la que referencia, siempre que se exija INTEGRIDAD REFERENCIAL). Al efectuar una matrícula sólo con especificar el NIF tendremos acceso al resto de datos del alumno gracias a esta relación, con lo que evitaremos inconsistencia y duplicidad de datos.
- Entre MATRICULAS y ASIGNATURAS. Para ello dejamos en tabla matrícula el código de la asignatura como clave ajena. Con sólo especificar el código de la asignatura accederemos a los datos de la misma.
- Entre ASIGNATURAS y DEPARTAMENTO. Para ello introducimos el código del departamento al que pertenece la asignatura en la tabla asignaturas como clave ajena, para que a partir de él, se obtengan todos los datos referentes al departamento que siempre serán los mismos y nunca dependerán de la matriculación o no de un alumno en él.

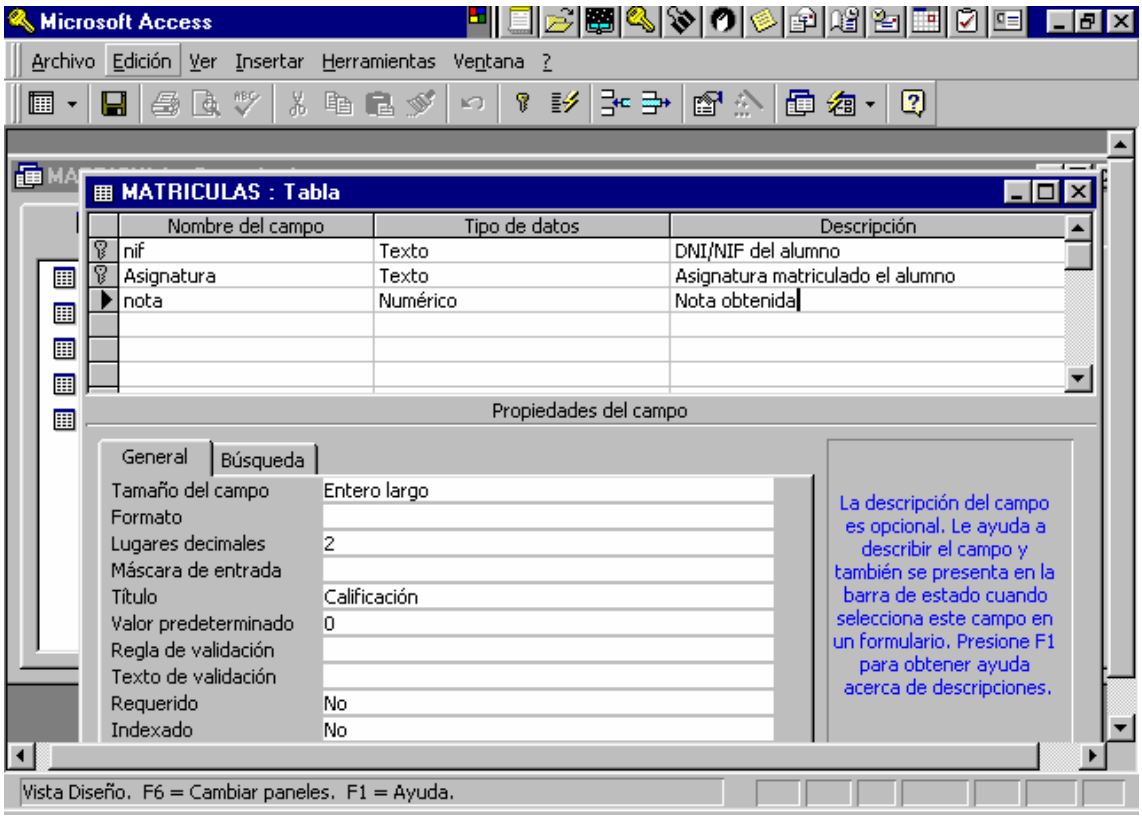
Los diseños de las tablas “maestras” quedan como antes excepto ASIGNATURAS que incluirá el código de departamento como clave ajena y quedará tal que:



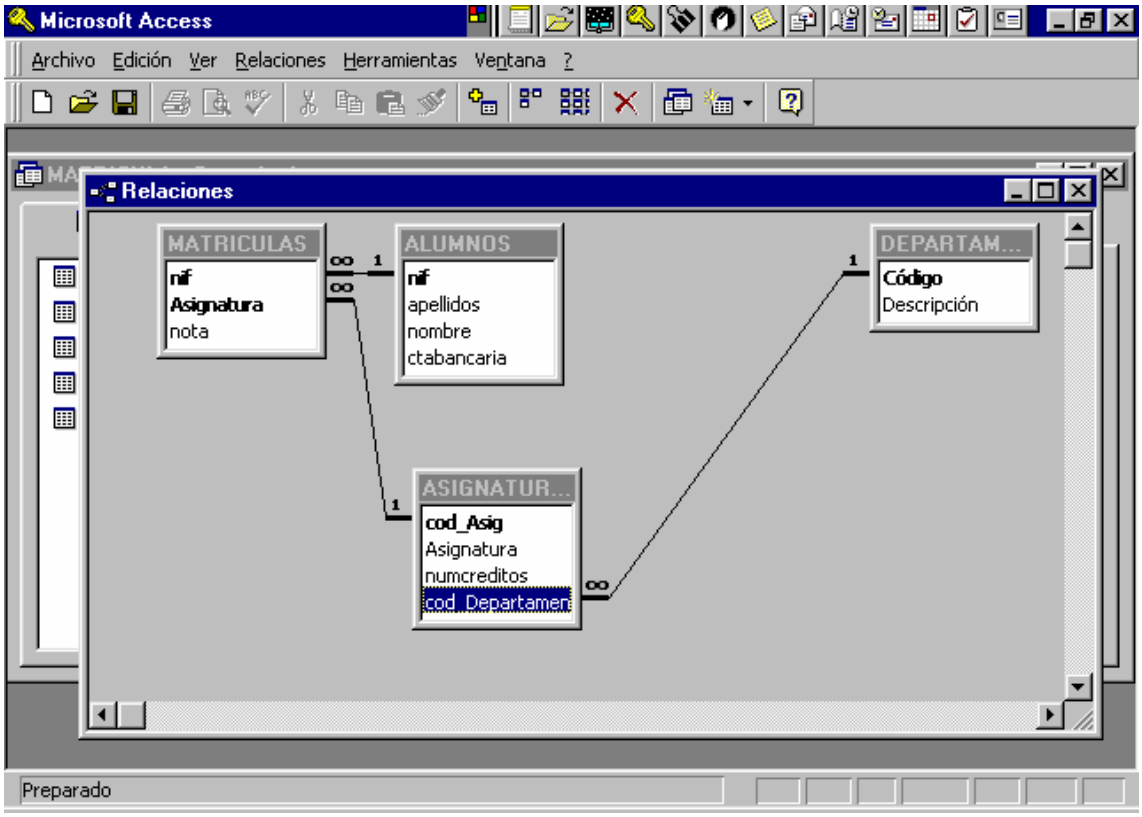
Y ahora nos toca abordar nuestra tabla que contendrá las matriculaciones. Sólo con el Nif del alumno, la asignatura y la nota tendremos ya acceso a todos los datos referentes a una matriculación en concreto. Ahora lo que tenemos es que buscar la clave primaria adecuada para nuestra tabla.

Vemos que si sólo especificamos el nif del alumno como clave primaria, al ser un valor único no podríamos tener a un alumno matriculado en más de una asignatura. Por la misma "regla de tres" si sólo ponemos la asignatura, sólo podrá haber un alumno matriculado por asignatura, cosa que tampoco queremos. Necesitamos entonces que la clave principal esté formada por ambos campos de forma que podremos introducir varios registros de un mismo alumno para distintas asignaturas ya que el valor único ha de estar compuesto por la pareja nif+asignatura. Así que ni que decir tiene que no podremos tener dos registros para un mismo alumno y asignatura.

El diseño de la tabla MATRICULA quedará así:

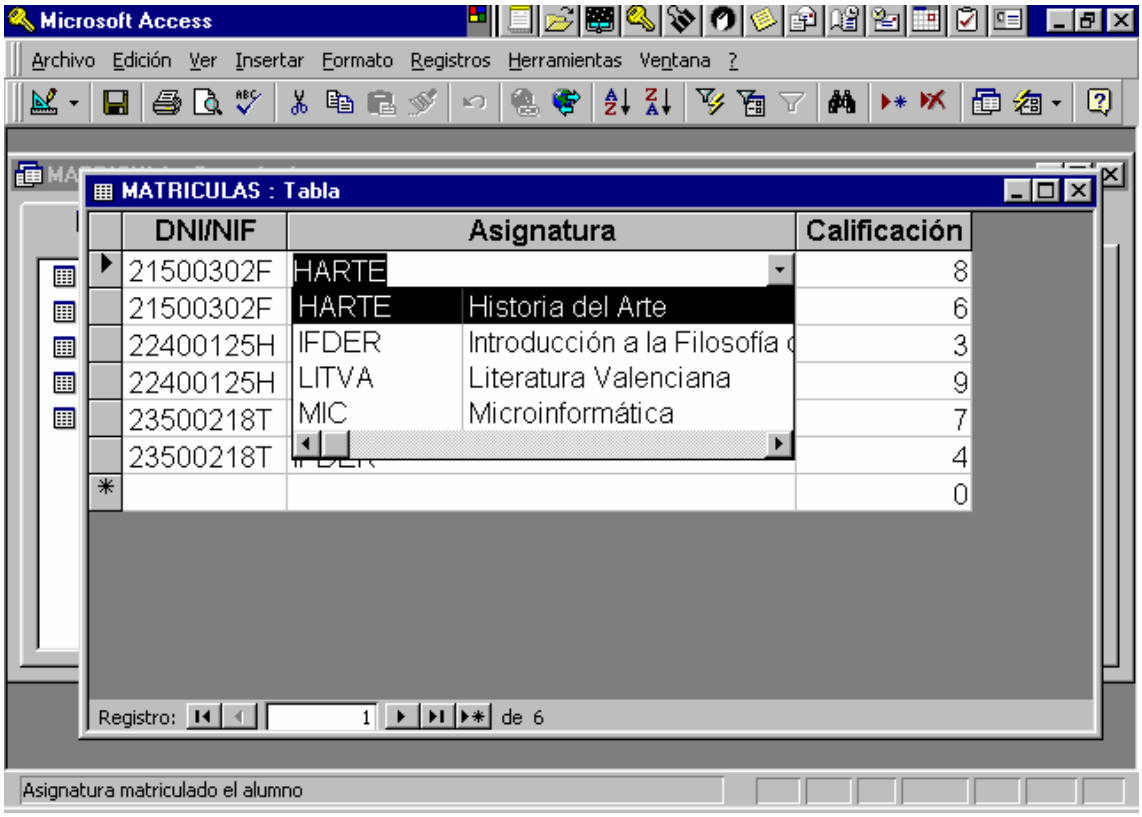


Y la ventana de Relaciones quedaría:



Por último indicar que una clave ajena puede tener valor NULO o su valor coincide con el de la clave de un registro de la tabla a la que referencia. Si permitimos valores NULOS en ellas, la introducción de datos será más flexible, pero tendremos registros de matriculación sin asignatura asociada por ejemplo. Si no exigimos INTEGRIDAD REFERENCIAL al crear relación podremos tener valores que no estén en la tabla maestra con la consiguiente inconsistencia de datos ya que tendremos valores en matriculaciones que no referenciarán a otros datos en tablas relacionadas, es decir, puede haber alumnos matriculados en asignaturas de las que no sepamos ni su nombre ni datos asociados.

Muy importante es la utilización del tipo de datos *asistente para búsquedas* entre los campos relacionados ya que nos permitirán introducir la información a través de una lista sin tener que teclearla. Por ejemplo el campo asignatura de la tabla MATRICULAs extraería los datos de la tabla ASIGNATURAS tal y cómo se ve en esta imagen:



Este “diseño” realizado podría haberse efectuado de diferentes maneras pero a nuestro parecer incluye todos los conceptos que son inherentes a un curso de iniciación posponiendo entrar a detalle a los diferentes tipos de relaciones para cursos de mayor nivel.