

EJERCICIO GUIADO. BASES DE DATOS

Recapitulando... GUI y PROCESO DE DATOS

Un programa básicamente recibe una serie de datos, que introduce el usuario, luego realiza una serie de cálculos con ellos, y finalmente presenta los resultados en pantalla.

El usuario usa una serie de elementos para introducir los datos en el programa y para darle las órdenes al programa para que realice los cálculos. Estos elementos son las ventanas con todos sus componentes: botones, cuadros de diálogos, etiquetas, listas desplegables, etc...

El programa realiza los cálculos con dichos datos y los resultados los presenta en elementos de la ventana para que el usuario pueda verlos de forma cómoda. Estos elementos donde aparecen resultados pueden ser etiquetas, cuadros de mensaje, etc...

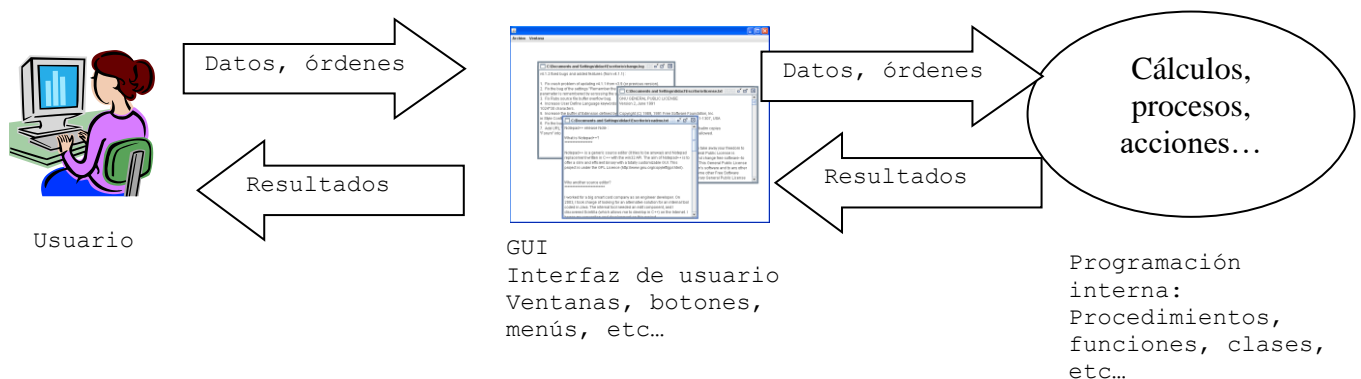
Hay que distinguir por tanto entre el GUI y el Proceso de Datos.

El GUI (interfaz gráfica de usuario) es el conjunto de elementos que permiten al usuario comunicarse con el programa. Estos elementos le permiten al usuario introducir datos, dar órdenes al programa y visualizar los resultados obtenidos. Para entendernos, el GUI es básicamente el diseño de las ventanas del programa.

Un buen diseño del GUI es vital. La interfaz de usuario debe hacer que el programa sea intuitivo, fácil de usar. Un programa incómodo, por muy bueno que sea, será rechazado por el usuario.

Aparte del GUI, internamente el programa debe realizar una serie de cálculos para obtener los resultados pedidos por el usuario. No hace falta decir que la programación de estos cálculos debe ser precisa y correcta, ya que en caso contrario el programa no serviría para nada, por muy buena que fuera la interfaz de usuario.

Todo lo anterior se puede representar gráficamente de la siguiente forma:



BASES DE DATOS

Ahora se va a dar una vuelta de tuerca más a lo indicado anteriormente añadiendo a todo este entorno el concepto de Base de Datos.

Hay que tener en cuenta que la mayoría de las aplicaciones que solicitan las empresas son programas cuya función principal es la de mantener datos de la propia empresa.

Estos programas suelen tener siempre las mismas funciones básicas. Deben ser capaces de:

- Almacenar datos producidos por la empresa: facturas, datos de clientes, productos, gastos, ingresos, trabajadores, etc...
- Visualizar dichos datos.
- Modificar / Eliminar dichos datos.
- Consultar dichos datos. Es decir, visualizar solo aquellos datos que cumplan una condición. Por ejemplo: visualizar las facturas del mes de enero, o mostrar los productos que estén en stock, mostrar los trabajadores que hayan trabajado más de 100 horas, etc...
- Realizar cálculos con los datos. Por ejemplo, calcular el total de ingresos entre dos fechas, calcular lo que hay que declarar de iva, etc...

Así pues, la mayor parte de las aplicaciones que encarga una empresa son aplicaciones que deben gestionar una serie de datos. Estos datos deben estar guardados en un fichero o ficheros. A este conjunto de datos que debe manipular el programa se le denomina BASE DE DATOS.

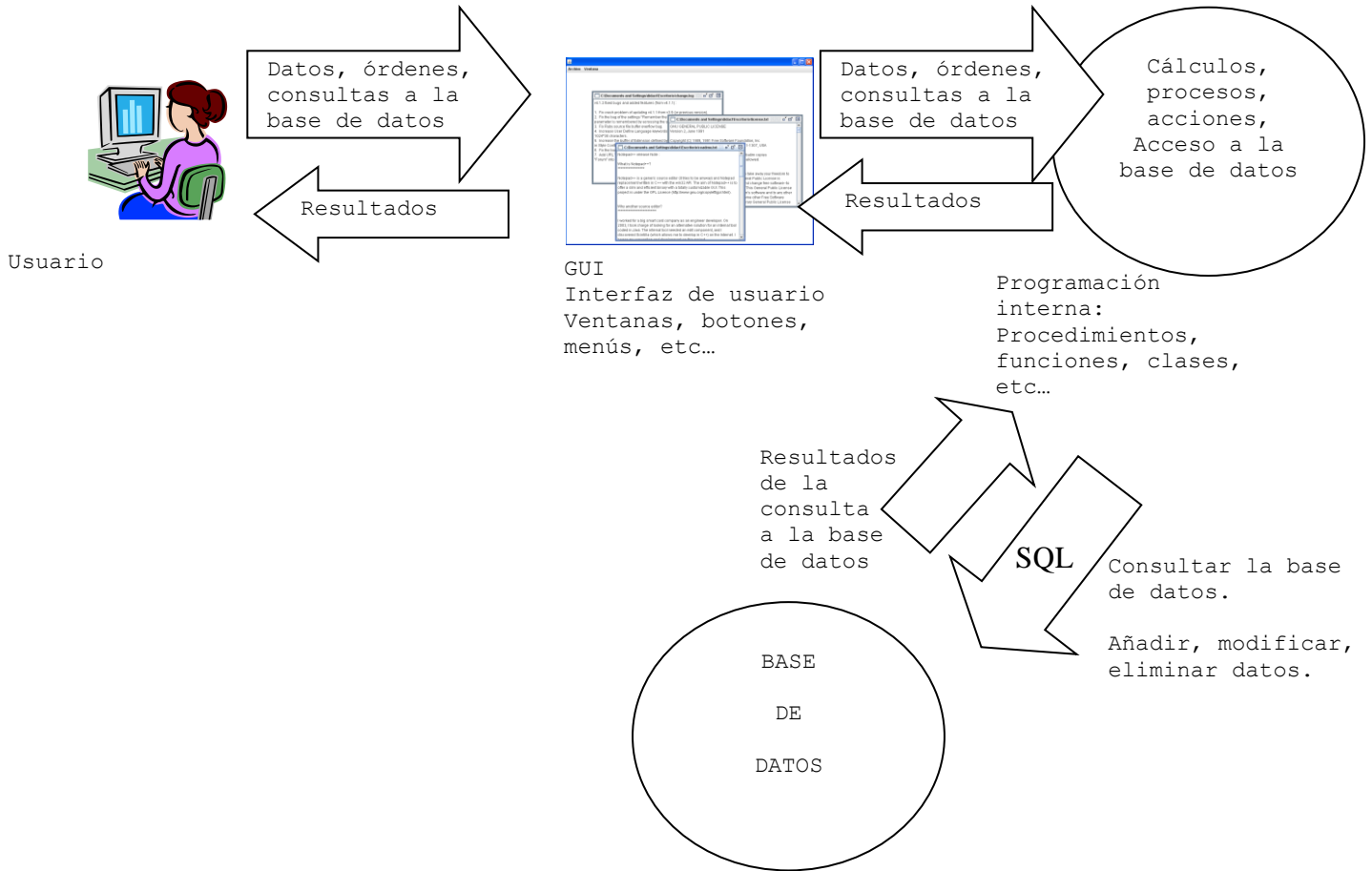
Una Base de Datos no es solamente un conjunto de datos almacenados en un mismo sitio. Hay que tener en cuenta que para que estos datos sean fáciles de manipular, es necesario que tengan una organización determinada. Una Base de Datos mal organizada se convierte simplemente en un cajón de sastre donde resulta imposible encontrar lo que se busca.

Así pues, es necesario tener en cuenta la forma en que se organizarán los datos dentro de la base de datos, o dicho de otra forma, el diseño de la base de datos.

En las próximas hojas guiadas se verán conceptos básicos para el diseño y creación de una base de datos, así como un lenguaje de consulta llamado SQL que nos permitirá acceder a los datos de una base de datos y realizar determinadas acciones sobre la base de datos.

Acceso a la BASE DE DATOS desde la aplicación

Añadiendo el concepto de Base de Datos a la idea de usuario, GUI y Proceso de Datos, podemos representar gráficamente un programa como sigue:



Analicemos el gráfico anterior:

1. El usuario le pide algo al programa. Por ejemplo el usuario quiere obtener un listado de clientes, o quiere saber cuanto se ha ingresado el último mes...
2. Para ello, el usuario activa los elementos del GUI necesarios. Es decir, activará la opción del menú que corresponda, o el botón de la barra de herramientas, etc. Dicho de otra manera, el usuario da una orden.
3. Es posible que el usuario tenga que introducir algún dato. Por ejemplo, si quiere saber los ingresos del mes de enero, tendrá que introducir "enero" para que el programa sepa de que mes hay que calcular los ingresos. Para ello el usuario usará un cuadro de texto del GUI, o un desplegable, etc...
4. Hay que tener en cuenta que para obtener un listado de clientes, o el total de ingresos de un mes, hay que consultar la base de datos. El usuario no se preocupará de eso ya que será el programa el que haga el trabajo.
5. El GUI traspasa la petición del usuario al módulo de Proceso de Datos. Este módulo es el corazón del programa, el que se encarga de ejecutar las órdenes. Esta parte del

programa analizará la orden y acto seguido buscará los datos necesarios en la base de datos.

6. Para acceder a la base de datos, el módulo de Proceso de Datos usará un lenguaje ideado para el acceso a bases de datos llamado SQL.
7. Gracias al lenguaje SQL, el módulo de Proceso de Datos obtendrá los datos necesarios que pedía el usuario (por ejemplo el listado de clientes) o bien los datos necesarios para realizar el cálculo (los ingresos de cada día del mes)
8. Una vez obtenido los resultados, el módulo de Proceso de Datos se los enviará al GUI y los presentará de forma agradable para el usuario.
9. El usuario mirará su pantalla, y verá aparecer en la ventana del programa (en el GUI) el resultado que había pedido.

Teniendo en cuenta lo visto, para la creación de un programa hay que:

- Diseñar un GUI agradable, cómodo y fácil de usar.
- Diseñar la base de datos de forma que el acceso a datos sea rápido.
- Hacer el programa de forma que responda a las peticiones del usuario transmitidas por el GUI accediendo a la base de datos y obteniendo información de ella, o bien realizando acciones en ella.

GESTOR DE BASE DE DATOS

Un gestor de base de datos es un programa que nos permite crear bases de datos, mantenerlas, realizar consultas en ellas, etc...

A través de un gestor de base de datos podemos crear la base de datos que usará nuestro programa.

Existen diversos programas gestores de base de datos, pero el que se usará será el programa Microsoft Access.

DISEÑO DE BASE DE DATOS. MODELO ENTIDAD - RELACIÓN

Antes de empezar a crear la base de datos usando el programa Access (o cualquier otro gestor de base de datos que vaya a usar) es necesario pensar en el diseño que tendrá dicha base de datos. O dicho de otro modo, como se organizarán los datos en la base de datos.

Para facilitar la tarea de organizar los datos en una base de datos, se usa una representación gráfica llamada Modelo Entidad – Relación, o simplemente Modelo E-R.

Podríamos comparar el Modelo E-R a los Diagramas de Flujo. El Modelo E-R nos facilita el diseño de bases de datos mientras que los Diagramas de Flujo nos facilitan el diseño de algoritmos.

EJERCICIO GUIADO Nº 1

Veremos en este ejercicio guiado como diseñar una base de datos a partir de la información aportada por una empresa, y un ejemplo de como representar dicha base de datos usando el Modelo E-R.

Supuesto Inicial

La empresa MANEMPSA (Mantenimiento de empresas S.A.) se encarga de realizar todo tipo de reparaciones y limpieza en los locales de sus clientes.

A esta empresa le interesa almacenar los datos de sus clientes. Concretamente le interesa almacenar el CIF de la empresa del cliente, el nombre de dicha empresa, la dirección y dos teléfonos de contacto.

Por otro lado, MANEMPSA quiere guardar los datos de los trabajadores que están en plantilla. Le interesa saber el nombre y apellidos de dichos trabajadores, el DNI, el sueldo y la fecha en que entraron en la empresa.

MANEMPSA quiere controlar los servicios que realiza cada trabajador. Concretamente, MANEMPSA quiere almacenar la fecha en que se hizo el servicio, el tipo de servicio que se hizo (fontanería, limpieza, electricidad, etc...) La cantidad cobrada por el servicio y un comentario donde se indique qué acciones se realizaron.

Hay que indicar que cada servicio lo realiza un solo trabajador, y que cada servicio se realiza a una sola empresa.

Cada trabajador tiene asignado un coche de la empresa. Hay un coche por cada trabajador y cada trabajador siempre usa el mismo coche. A MANEMPSA le interesa almacenar información de cada coche. Le interesa almacenar la matrícula, la marca, el modelo y el año de compra.

MANEMPSA le encarga un programa para gestionar toda esta información. Se pide que diseñe la base de datos correspondiente que se usará en el programa. Represente dicha base de datos usando el Modelo E-R.

1. Concepto de Entidad.

Entidad. Una entidad es un objeto que se quiere representar a través de una serie de características.

Las entidades son aquellos elementos que juegan un papel importante en la base de datos. Son los elementos sobre los que queremos almacenar información.

2. Lo primero que hay que hacer es distinguir las entidades del supuesto. Es decir, averiguar los elementos que se quieren almacenar en la base de datos.

En el supuesto, las entidades son: los clientes, los trabajadores, los servicios que realizan los trabajadores y los coches de los trabajadores.

3. Atributos de una Entidad.

A la empresa le interesa almacenar información sobre las entidades. Es decir, le interesa almacenar información sobre los clientes, los trabajadores, los servicios y los coches.

La información que se quiere almacenar de cada entidad se denomina **atributos** de una Entidad.

Por ejemplo, de los clientes, se quiere almacenar las siguientes propiedades: el CIF de la empresa, el nombre, etc...

A continuación se indican los atributos que se almacenarán de cada entidad, teniendo en cuenta el supuesto:

Entidad Clientes

Atributos: CIF, nombre empresa, dirección, teléfono1, teléfono2.

Entidad Trabajadores

Atributos: nombre, apellidos, DNI, sueldo, fecha de entrada.

Entidad Servicios

Atributos: fecha, tipo, cantidad, comentario.

Entidad Coches

Atributos: matrícula, marca, modelo, año de compra.

4. Atributos clave.

Un Atributo Clave es aquel atributo cuyo valor no se repite y no puede estar vacío para todos los elementos de una entidad.

Los Atributos Clave distinguen a cada elemento de una entidad.

Por ejemplo, en el caso de la entidad Clientes, el atributo clave será el CIF, ya que no puede haber dos clientes con CIF iguales, y todos los clientes tendrán CIF. El CIF permite distinguir a un cliente de otro.

Es conveniente asignar un atributo clave para cada entidad. En algunos casos podemos elegir uno de los atributos de la entidad como atributo clave, en otros casos ninguno de los atributos de la entidad podrá ser atributo clave y entonces tendremos que añadir uno nosotros.

Veamos los atributos clave de cada entidad:

Entidad Clientes: El atributo clave será el CIF.

Entidad Trabajadores: Se puede elegir como atributo clave el DNI.

Entidad Coches: Se puede elegir como atributo clave la matrícula.

Entidad Servicios.

En el caso de la entidad Servicios, tenemos que ninguno de los atributos que posee puede ser atributo clave. Observa:

- Dos servicios distintos pueden ser realizados en la misma fecha, por lo que dos servicios pueden tener fechas repetidas: la fecha no puede ser atributo clave.
- Dos servicios distintos pueden ser del mismo tipo, por lo que dos servicios pueden tener el mismo tipo: el atributo tipo no puede ser atributo clave.
- Dos servicios distintos podrían tener el mismo comentario, o no tener comentario en absoluto: el atributo comentario no puede ser atributo clave.
- Dos servicios distintos podrían haber costado lo mismo, por lo que el atributo cantidad no puede ser atributo clave.

No existe para la entidad Servicios un atributo que identifique cada servicio. En estos casos se suele añadir un atributo que sea atributo clave, es decir, un "código" o "identificador".

En nuestro ejemplo, añadiremos un atributo "número" que será el número de servicio realizado. Cada servicio tendrá un número distinto, por lo que el atributo número será el atributo clave.

Así pues, los atributos de la entidad Servicios serán los siguientes:

Entidad Servicios

Atributos: fecha, tipo, cantidad, comentario, número.

Y el atributo número será su atributo clave.

5. **Relaciones** entre entidades.

Las entidades se relacionan entre sí. Es necesario distinguir en el supuesto cuales son estas relaciones. Veamos:

Un Trabajador realiza Servicios. Por lo tanto, las entidades Trabajadores y Servicios se relacionan.

Un Servicio se realiza a un Cliente. Por lo tanto las entidades Servicios y Clientes se relacionan.

Un Coche lo usa un Trabajador. Por lo tanto las entidades Coche y Trabajadores se relacionan.

Estas son las relaciones:

Coches --- Trabajador --- Servicios --- Clientes

6. Cuando se establecen las relaciones, hay relaciones que están implícitas. Por ejemplo, no se ha establecido una relación entre Trabajadores y Clientes:

“Un Trabajador hace trabajos a Clientes.”

No está establecida una relación directa entre ambas entidades, pero sí se relacionan a través de la entidad Servicios:

Trabajador --- Servicios --- Clientes

Así pues, en realidad, trabajadores y clientes se relacionan, a través de la entidad servicios.

7. **Cardinalidad** de las relaciones o **Tipo** de relaciones.

Según la forma en que se relacionen dos entidades, tenemos tres tipos de relaciones:

Relaciones 1 a 1

En este tipo de relación, un elemento de la primera entidad se relaciona con un elemento de la segunda entidad, y viceversa.

En nuestro ejemplo, la relación entre Coches y Trabajadores es del tipo 1 a 1, ya que un coche pertenece a un solo trabajador, y ese trabajador solo usa ese coche.

Relaciones 1 a Muchos

En este tipo de relación, un elemento de la primera entidad se relaciona con muchos elementos de la segunda, pero un elemento de la segunda, se relaciona con uno solo de la primera.

En nuestro ejemplo, la relación Trabajadores – Servicios es del tipo 1 a Muchos. Observa: un trabajador realiza a lo largo de su vida laboral muchos servicios, pero un servicio es realizado solo por un trabajador (por lo que nos dice el supuesto).

La relación entre Clientes y Servicios es también del tipo 1 a Muchos. Observa: un cliente solicita a lo largo de su existencia muchos servicios a la empresa, pero cada servicio se realiza a un solo cliente (por lo que nos dice el supuesto).

Relaciones Muchos a 1

Este tipo de relación es el mismo que el anterior. Basta con “darle la vuelta” a la relación.

Por ejemplo, se ha visto que la relación Trabajadores – Servicios es del tipo 1 a Muchos. Por tanto, la relación Servicios – Trabajadores es del tipo Muchos a 1.

Relaciones Muchos a Muchos

En este tipo de relación un elemento de la primera entidad se relaciona con muchos elementos de la segunda entidad, y viceversa.

En nuestro supuesto no tenemos ninguna relación de este tipo, pero supongamos la siguiente relación entre Profesores y Cursos:

“Un profesor puede dar varios cursos, y un curso es impartido por varios profesores”

Esta relación sería del tipo Muchos a Muchos.

8. Averiguar el tipo de relaciones.

Para averiguar el tipo de cada relación, es necesario estudiar bien el enunciado del supuesto, y pedir información al cliente que nos encarga el programa, de lo contrario, tendremos que hacer suposiciones para asignar un tipo a cada relación.

En nuestro supuesto, las relaciones tienen el siguiente tipo:

Coches -1---1- Trabajadores -1---M- Servicios -M---1- Clientes

Relación Coches – Trabajadores. Tipo 1 a 1.

Relación Trabajadores – Servicios. Tipo 1 a Muchos.

Relación Servicios – Clientes. Tipo Muchos a 1.

9. Intercambio de atributos clave entre entidades relacionadas.

Para hacer efectivas las relaciones de la base de datos, es necesario intercambiar los atributos claves entre las entidades relacionadas siguiendo ciertas reglas.

Las reglas son las siguientes:

- Primera Regla. Si dos entidades se relacionan de forma 1 – 1, entonces debe pasarse el atributo clave de una de ellas a la otra, y viceversa.

En nuestro ejemplo, las entidades Coches y Trabajadores se relacionan de forma 1 a 1. Esto implica que es necesario añadir el atributo clave de Coches a Trabajadores, y viceversa.

Así pues, será necesario añadir el atributo *matrícula* a Trabajadores, y el atributo *DNI* a Coches.

Por tanto, las entidades Trabajadores y Coches quedarán así:

Entidad Trabajadores

Atributos: nombre, apellidos, DNI, sueldo, fecha de entrada, matrícula

Entidad Coches

Atributos: matrícula, marca, modelo, año de compra, DNI

- Segunda Regla. Si dos entidades se relacionan de forma 1 – Muchas, entonces, se debe pasar el atributo clave de la entidad de la parte de la relación 1, a la entidad de la parte de la relación Muchas.

En nuestro ejemplo, las entidades Trabajadores --- Servicios se relacionan de forma 1 a Muchas, por tanto, será necesario traspasar el atributo clave de la entidad Trabajadores a la entidad Servicios, con lo que quedarán así:

Entidad Trabajadores

Atributos: nombre, apellidos, DNI, sueldo, fecha de entrada, matricula

Entidad Servicios

Atributos: fecha, tipo, cantidad, comentario, número, DNI.

Por otro lado, las entidades Servicios --- Clientes se relacionan de forma Muchas a 1. Por tanto, será necesario traspasar el atributo clave de la entidad Clientes a la entidad Servicios, con lo que quedarán así:

Entidad Servicios

Atributos: fecha, tipo, cantidad, comentario, número, DNI, CIF.

Entidad Clientes

Atributos: CIF, nombre empresa, dirección, teléfono1, teléfono2.

10. Claves Foráneas.

La razón de traspasar el atributo clave de una entidad a otra es que exista un elemento de unión, en ambas entidades relacionadas.

Cuando una entidad posee un atributo clave de otra entidad, se dice que ese atributo es una **clave foránea**.

Después de realizar estos traspasos, nuestras entidades quedan con los siguientes atributos.

Nota: se subraya con línea continua el atributo clave en cada entidad, y se subraya con línea de puntos las claves foráneas de cada entidad.

Entidad Coches

Atributos: matrícula, marca, modelo, año de compra, DNI

Entidad Trabajadores

Atributos: nombre, apellidos, DNI, sueldo, fecha de entrada, matricula

Entidad Servicios

Atributos: fecha, tipo, cantidad, comentario, número, DNI, CIF.

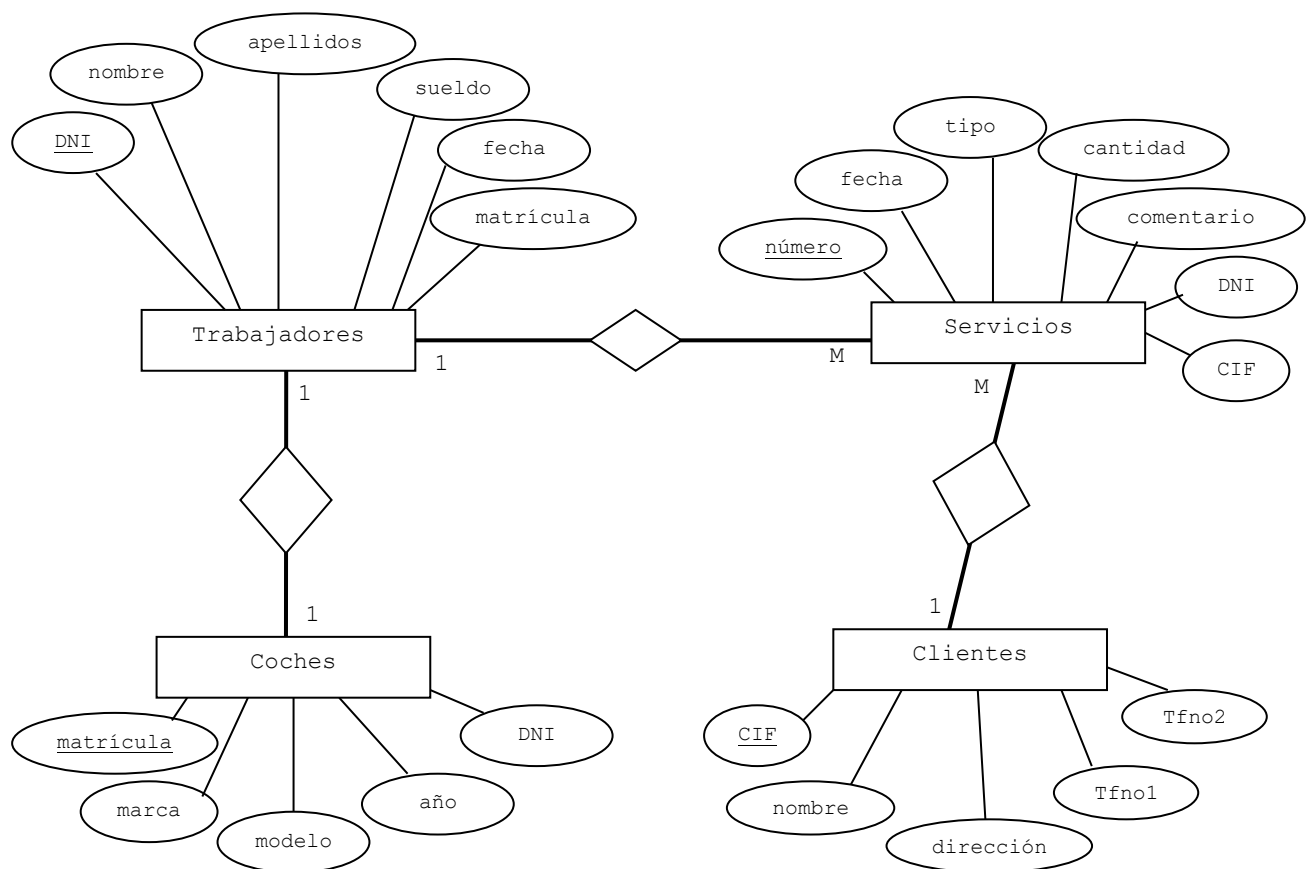
Entidad Clientes

Atributos: CIF, nombre empresa, dirección, teléfono1, teléfono2.

No confundas el concepto de atributo clave, con el concepto de clave foránea.

11. Modelo E-R.

Una vez definidas las entidades, los atributos, atributos claves, relaciones, tipos de relaciones, y claves foráneas, todo esto se puede representar de forma gráfica como sigue:



El gráfico anterior es lo que se llama un Modelo Entidad Relación y es una representación de la base de datos que acabamos de diseñar.

En este gráfico observarás lo siguiente:

- Las entidades se representan en rectángulos.
- Los atributos de cada entidad se representan en óvalos.
- El atributo clave de cada entidad se subraya.
- Las relaciones entre entidades se representan usando líneas con rombos.
- Se indica el tipo de relación (1 a 1, o 1 a Muchos) añadiendo a la línea de la relación un 1 o una M para indicar el tipo de relación.

CONCLUSIÓN

Las Bases de Datos son elementos vitales en la creación de aplicaciones para empresas.

Para diseñar una Base de Datos hay que especificar los siguientes elementos:

- **Entidades cuya información queremos almacenar.**
- **Atributos de dichas entidades.**
- **Atributos claves para dichas entidades.**
- **Relaciones entre entidades.**
- **Definir el tipo de relaciones.**
- **Asignar las claves foráneas según el tipo de relación.**

Una vez especificados estos elementos, todo el diseño se puede representar gráficamente a través de un modelo gráfico llamado Modelo Entidad-Relación.