

# Introducción a Base de Datos



# Que es una Base de Datos (BD)

- **Definición 1**

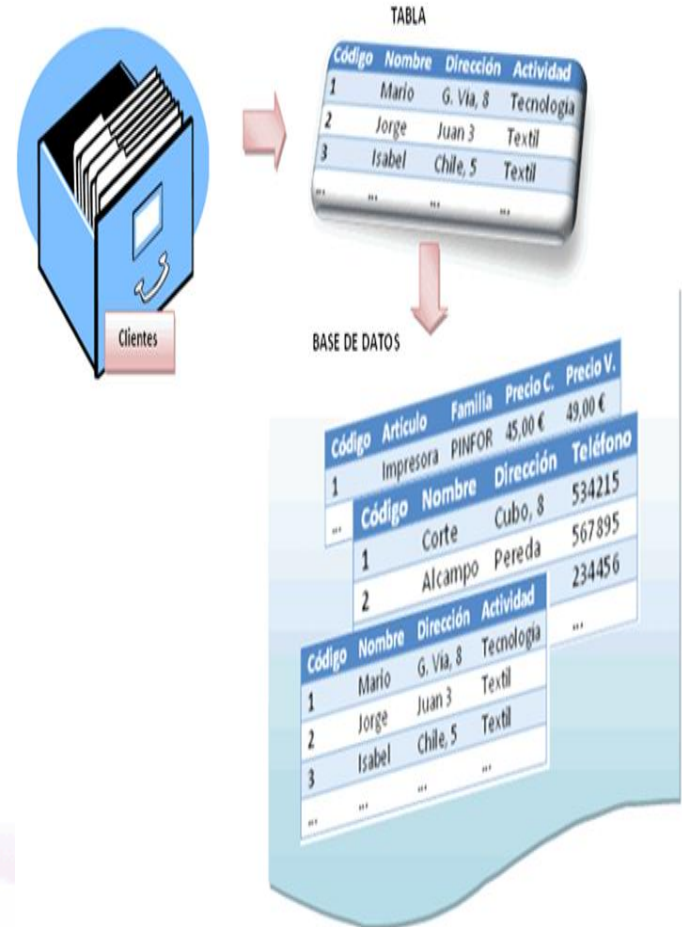
Desde un punto de vista muy simple una base de datos es básicamente un sistema para archivar información en una computadora cuyo propósito general es mantenerla disponible cuando se solicite. La información en general puede ser cualquier cosa que se considere importante para un individuo u organización.

- **Definición 2**

Es un conjunto de información almacenada en una computadora, dicha información puede ser manipulada con operaciones como agregar, consultar, eliminar entre otras. Y puede utilizarse en aplicaciones tan sencillas como llevar la información como una pequeña agenda telefónica hasta la información de una gran corporación.

- **Definición 3**

Una base de datos es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite. Una base de datos es un sistema de archivos electrónico. ( <http://www.masadelante.com/faqs/base-de-datos> )



# Porque utilizar una BD

Las ventajas de un sistema de BD sobre los métodos tradicionales de mantener registros en papel serán quizá más evidentes a continuación:

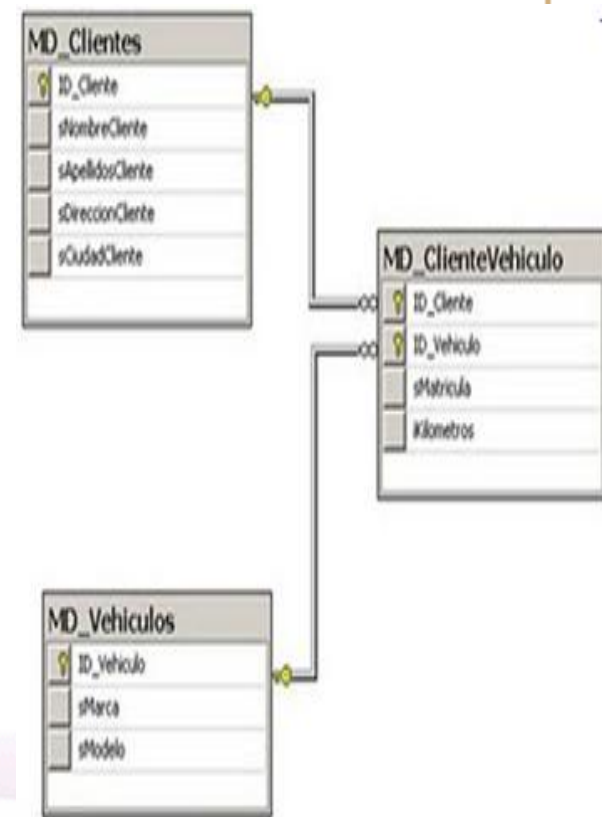
- **Es compacto:** No hacen falta archivos de papeles que pudieran ocupar mucho espacio.
- **Es rápido:** La máquina puede obtener y modificar datos con mucha mayor velocidad que un ser humano.
- **Es menos laborioso:** Se elimina gran parte del tedio de mantener archivos a mano.
- **Es actual:** Se dispone en cualquier momento de la información precisa y al día.



# Ventajas del enfoque de BD

A continuación se explica algunas de las ventajas específicas que emanan del concepto de control centralizado:

- Es posible disminuir la redundancia
- Es posible evitar la inconsistencia
- Es posible compartir los datos
- Es posible hacer cumplir las normas
- Es posible aplicar restricciones de seguridad
- Es posible mantener la integridad



# Base de datos Relacionales

Antes de explicar que es una base relacional es importante algunos conceptos Básicos.

- **Campo:** Es la unidad más pequeña a la cual uno puede referirse en un programa. Desde el punto de vista del programador representa una característica de un individuo u objeto. Ejemplo

Si quisiéramos almacenar la información de un estudiante los campos serían las características asociadas a este estudiante tales como:

- Nombre
- Fecha de Nacimiento
- Teléfono etc.

- **Registro:** Es una colección de campos de iguales o de diferentes tipos.

Siguiendo con el ejemplo diríamos entonces que cada campo representa a un estudiante, el cual está formado por el grupo de campos o características individuales.

Carnet	Nombre	Apellido
1590-1001	Jorge	López
1590-1002	Luis	Córdova
1590-1003	Maria	Pérez

- **Archivo:** Es una colección de registros almacenados siguiendo una estructura homogénea. .



# Base de Datos Relacionales

- **Base de Datos:**

Podemos decir entonces que una base de datos es una colección de archivos interrelacionados y cuyo contenido engloba a la información concerniente(almacenadas en archivos) de una organización, de tal manera que los datos estén disponibles para los usuarios.

Los tres componentes principales de un sistema de base de datos:

- El hardware
- El software DBMS
- Los datos a manejar (así como el personal encargado del manejo del sistema).



# Base de datos Relacionales

- **ENTIDAD:**

En una base de datos relacional conoceremos como entidad a un objeto real o abstracto acerca del cual se almacenan datos, el cual se diferencia de otros objetos a partir de atributos.

Puede ser de diversa naturaleza:

- Tangible: Persona, vehículo, material.
- Intangible: Proyecto, materia, etc.
- Eventual: Compra, venta, vuelo, etc.
- Rol: Cliente, vendedor, etc.
- Localización: Sucursal, bodega, etc.

- **ATRIBUTO:**

Elemento o dato que describe o identifica cada entidad dándole características propias. Normalmente los atributos son virtualmente ilimitados para una entidad pero deben seleccionarse los de mayor relevancia para el problema que se está analizando. Ejemplo

Para una entidad llamada "Persona", se podría tener varios atributos como Nombre, Apellido, Peso, Estatura etc.





# Base de datos Relacionales

- Habiendo explicado algunos conceptos básicos discutiremos el termino *Base de datos Relacional*. **Una relación** es una asociación entre los datos guardados en diferentes archivos o tablas, por ejemplo.
  - Un estudiante se asocia con los cursos asignados en los diferentes semestres
  - Un cliente se asocia con la emisión de una factura de compra.
  - Un curso se asocia a un semestre en particular.

Un aspecto importante de las relaciones entre las tablas es la cardinalidad. **La Cardinalidad** se refiere a la cantidad de asociaciones que pueden existir entre dos tipos de registros de datos. Dependiendo del tipo de cardinalidad que tenga la relación esta puede denominarse de las siguientes maneras:

- 1) Relación de uno a uno
- 2) Relación de uno a muchos
- 3) Relación de muchos a muchos





# Relación uno a uno

Una registro de una Entidad A se relaciona únicamente con un registro de una tabla B y viceversa. Este tipo de relación no es común en el mundo de las bases de datos. Ejemplos

- Supongamos que tenemos una tabla llamada vehículo y otra llamada Matriculas, la relación que se haría entre ambas tablas sería la siguiente: un vehículo solo puede estar asociado a una sola matricula y viceversa.



- Supongamos que almacenamos la información de un empleado en una tabla y en otra tabla guardamos los currículum vitae de los empleados. La relación en este caso es de uno a uno ya que un empleado puede tener solo un currículum vitae y viceversa.



# Relación Uno a Mucho

Una registro de una Entidad A se relaciona con varios registros de una tabla B. Ejemplos

- Un Departamento tiene varios municipios, y un municipio solo pertenece a un departamento



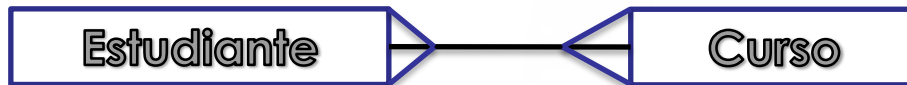
- Un estudiante Municipio tiene muchos ciudadanos, pero un ciudadano solo puede pertenecer a un municipio.



# Relación Muchos a Muchos

Cuando un registro de una entidad A puede estar relacionado con más de un registro de la entidad B y viceversa. En este caso las dos tablas no pueden estar relacionadas directamente, se tiene que añadir una tabla entre las dos que incluya los pares de valores relacionados entre sí. Ejemplo

- Un estudiantes puede asignarse varios cursos, pero un curso también puede tener varios estudiantes asignados.



- Sin embargo esto no puede ser reflejado de esta forma si no que es necesario añadir una nueva tabla para poder registrar la información en una Base de Datos



# Ejemplos

- Explicación de relaciones para un sistema de facturación: (clientes, artículos, factura, detalle de factura)
- Explicación de relaciones entre automóvil, partes.



# Que es una Llave Primaria

Una **llave primaria** es aquel atributo (o atributos) que consideramos clave para la identificación de los demás atributos que describen a la entidad. Ejemplo

- Por ejemplo, si consideramos la entidad ALUMNO este podría tener los siguientes atributos: Carnet, Nombre, Dirección, Teléfono, de todos estos atributos el que podremos designar como llave primaria es el número de carnet, ya que es diferente para cada alumno y este los identifica de forma única dentro de la universidad.
- Otro ejemplo puede ser la entidad CURSO, el cual puede contener los atributos: Código, Nombre, Créditos. Aquí el atributo Código es al que podemos designar como llave primaria.

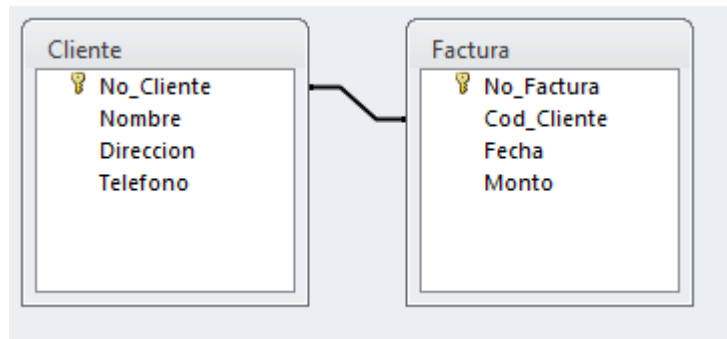


# Que es una llave Foránea

Es uno o mas campos de un tabla que hacen referencia al campo o campos de llave primaria de otra tabla, una llave foránea indica como esta relacionadas las entidades.

Los datos en los atributos de llave primaria y llave foránea deben coincidir en el tipo de datos y tamaño que almacenan, aunque los nombres de los atributos no sean los mismos.

Ejemplo en la entidad cliente la llave primaria es No\_Cliente y en factura la llave Foránea se llama Cod\_Cliente, estos datos se relacionan entre sí de la forma en que en una factura se almacena el código del cliente al que pertenece dicha factura



# Arquitectura para Sistemas de Base de Datos

Los usuarios no tienen porque conocer como están organizados y almacenados los datos, es por ese motivo que una base de datos debe presentar los datos de forma que el usuario pueda interpretarlos y modificarlos según las tareas que éste tenga asignados. Evidentemente esto no lo podemos aplicar a un informático que necesite saber donde se encuentran físicamente los datos para poder tratarlos.

Según la visión y la función que realicen los usuarios sobre la BD podemos dividir en tres niveles principales :

- **Nivel Interno:** es el nivel más cercano al almacenamiento físico de los datos. Permite escribirlos tal y como están almacenados en la computadora. En este nivel se diseñan los archivos que contienen la información, la ubicación de los mismos y su organización, es decir se crean los archivos de configuración.





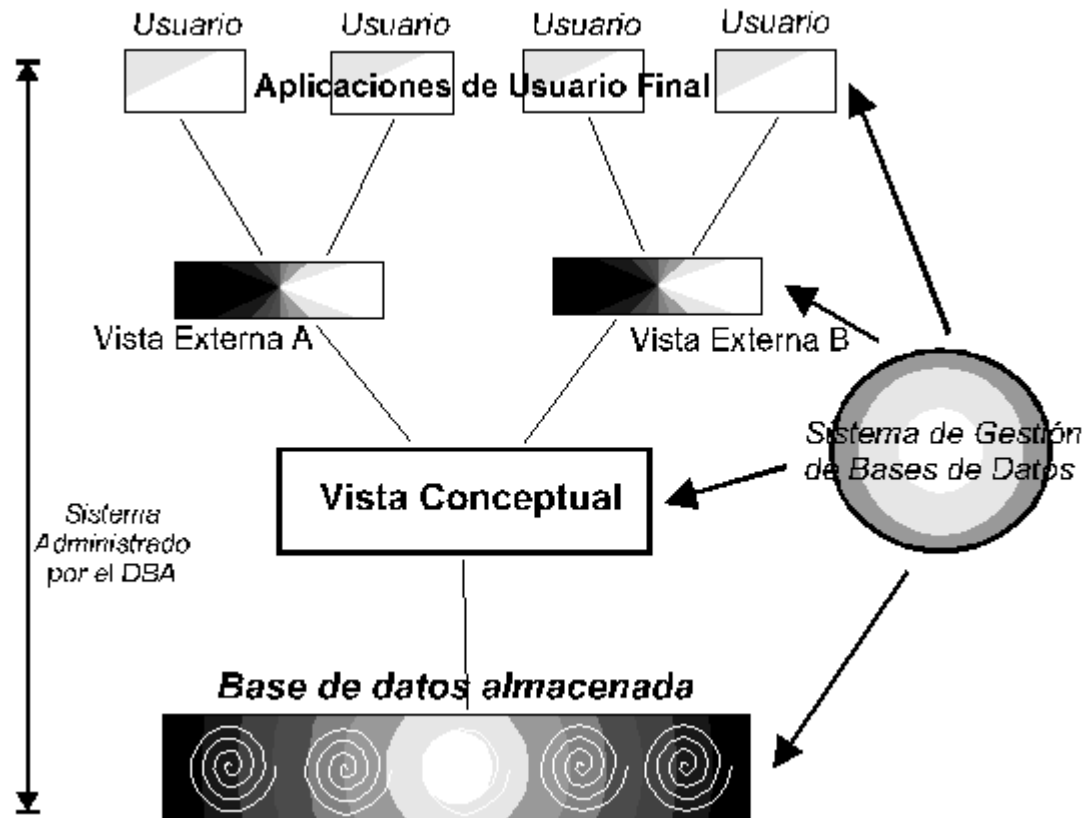
# Arquitectura para Sistemas de Base de Datos

- **Nivel conceptual:** En este nivel se representan los datos que se van a utilizar sin tener en cuenta aspectos como lo que representamos en el nivel interno. A grandes rasgos este nivel debe ser un panorama de los datos tal como son y no como por fuerza los perciben los usuarios debido a las limitantes del lenguaje o el equipo específico utilizado.
- **Nivel externo:** es el más cercano al usuario. En este nivel se describen los datos o parte de los datos que más interesan a los usuarios.

Por ejemplo para un usuario final será o bien un lenguaje de consulta, o algún lenguaje de aplicación especial, quizá manejado por una forma, reporte o menú adaptados a ese usuario.



# Arquitectura para Sistemas de Base de Datos



# Administrador de Base de Datos (DBA)

- El administrador de la base de datos es la persona que toma las decisiones estratégicas y de políticas con respecto a la información de la base de datos



# Responsabilidad del DBA

La responsabilidad general del DBA es facilitar el desarrollo y el uso de la Base de Datos dentro de las guías de acción definidas por la administración de los datos.

El DBA es responsable primordialmente de:

- Definir el esquema conceptual
- Definir el esquema interno
- Definir procedimientos de respaldo y recuperación
- Administrar la estructura de la Base de Datos
- Administrar la actividad de los datos
- Administrar el Sistema Manejador de Base de Datos
- Establecer el Diccionario de Datos
- Asegurar la confiabilidad de la Base de Datos
- Confirmar la seguridad de la Base de Datos
- Entre otras



# Sistema de Gestión de Base de Datos (SGBD)

Es el conjunto de programas que maneja todo acceso a la base de datos. Conceptualmente, lo que sucede es lo siguiente:

- Un usuario solicita acceso, empleando algún sublenguaje de datos determinado (por eje. SQL).
- El SGBD interpreta esa solicitud y la analiza.
- El SGBD inspecciona, en orden, el esquema externo de ese usuario, el esquema conceptual y la definición de la estructura almacenada.
- El SGBD ejecuta las operaciones necesarias sobre la base de datos almacenada. (ver fig. diapositiva 16)

Las funciones del SGBD por lo menos son las siguientes:

- Definición de datos
- Manipulación de datos
- Seguridad e integridad de datos
- Recuperación y concurrencia de los datos.
- Diccionario de datos
- Desempeño



# Lenguaje de Base de Datos SQL

El Lenguaje SQL (structured Query Language) utilizado para base de datos fue desarrollado entre 1974 y 1975 en IBM, en esos años se le conocía como llamaba SEQUEL (Structured English Query Language) y se utilizaba como interfaz para un sistema experimental de base de datos llamado SYSTEM R.

SQL fue introducido por Oracle por primera vez en una base de datos comercial en el año 1979 y pronto se convirtió en un estándar ANSI e ISO dando lugar de allí en adelante a varias versiones mejoradas del lenguaje.

El lenguaje se utiliza para definir toda las estructuras de la base de datos: crear entidades, atributos, y todas su operaciones, crear, eliminar, modificar y por supuesto consultar la información almacenada en ellas.



# Lenguaje de Base de Datos SQL

El SQL incluye un componente de lenguaje de definición de datos llamada **DDL (Data Definition Language)** el cual se encarga de realizar la función de definición de datos del SGBD, pudiendo realizar la creación, eliminación y modificación de los objetos de la base de datos (es decir los metadatos).

Los objetos que pueden manipularse a través del DDL son: tablas, vistas, índices y otras objetos relacionados con la definición de la base de datos.

En principio sólo los administradores y los usuarios propietarios pueden acceder a cada objeto, salvo que se modifiquen los privilegios del objeto para permitir el acceso a otros usuarios. Algunas de las instrucciones que son parte del DDL son: create table, drop table, create index, etc.





# Lenguaje de Base de Datos SQL

El **DML (Data Management Language)** lenguaje de manipulación de datos, la cuales cuenta con una gama amplia de sentencias que sirven para recuperar, insertar, borrar, modificar los datos almacenados en las estructuras creadas por el DDL: algunas de las instrucciones que forman parte del DML son: Select, insert, update, delete.

El **DCL (Data Control Language)** es la parte menos conocida del SQL , siendo la finalidad controlar el acceso a datos denegando y otorgando privilegios sobre los objetos existentes. Las instrucciones que forman parte del DCL son: Commit, Rollback, Savepoint, save transaction.



# El Modelo Entidad-Relación (E/R)

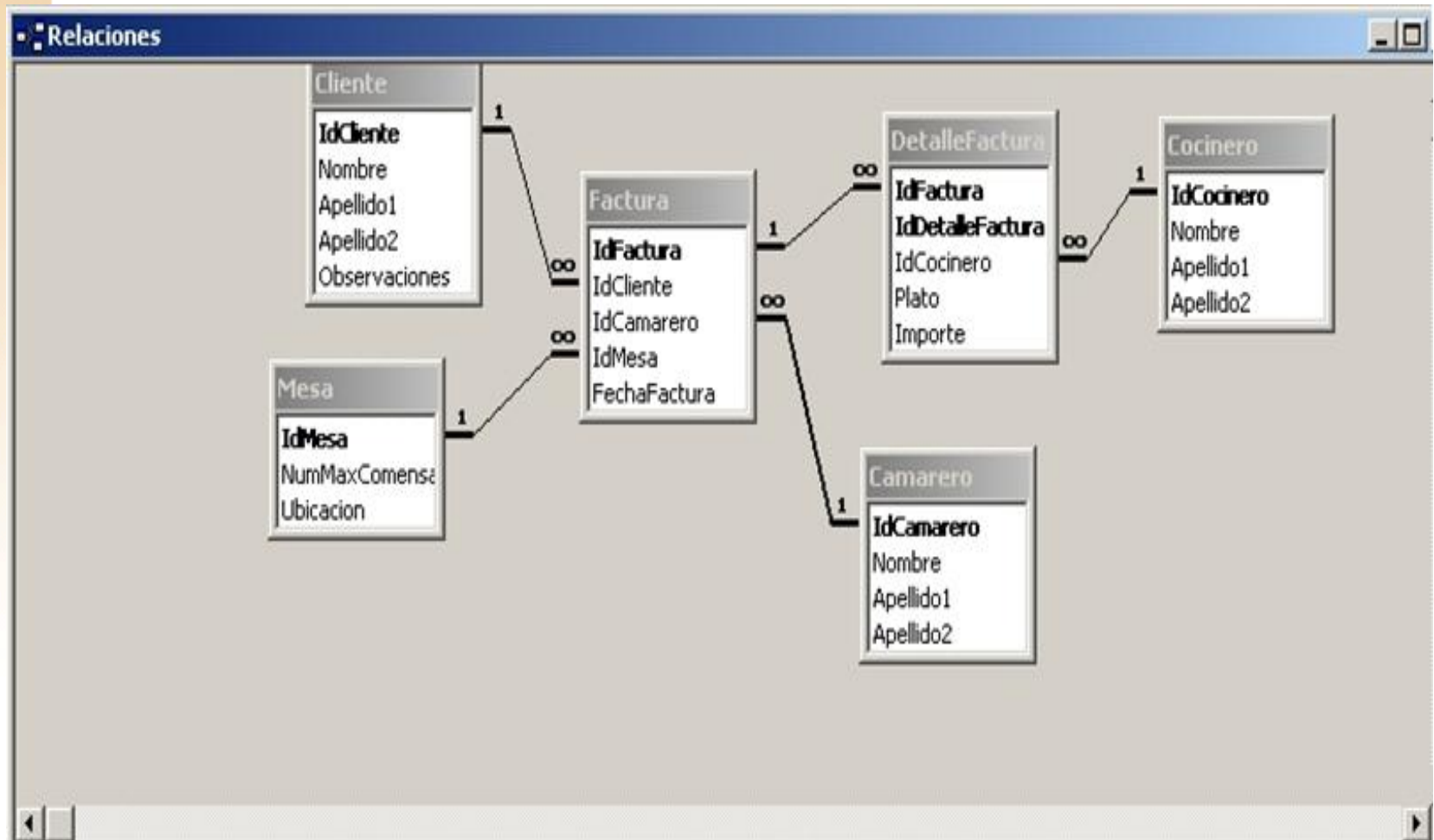
Cuando se utiliza una base de datos para gestionar información, se está plasmando una parte del mundo real en una serie de tablas, registros y campos ubicados en un computador; creándose un modelo parcial de la realidad. Antes de crear físicamente estas tablas en el ordenador se debe realizar un modelo de datos.

Se suele cometer el error de ir creando nuevas tablas a medida que se van necesitando, haciendo así el modelo de datos y la construcción física de las tablas simultáneamente. El resultado de esto acaba siendo un sistema de información parcheado, con datos dispersos que terminan por no cumplir adecuadamente los requisitos necesarios.

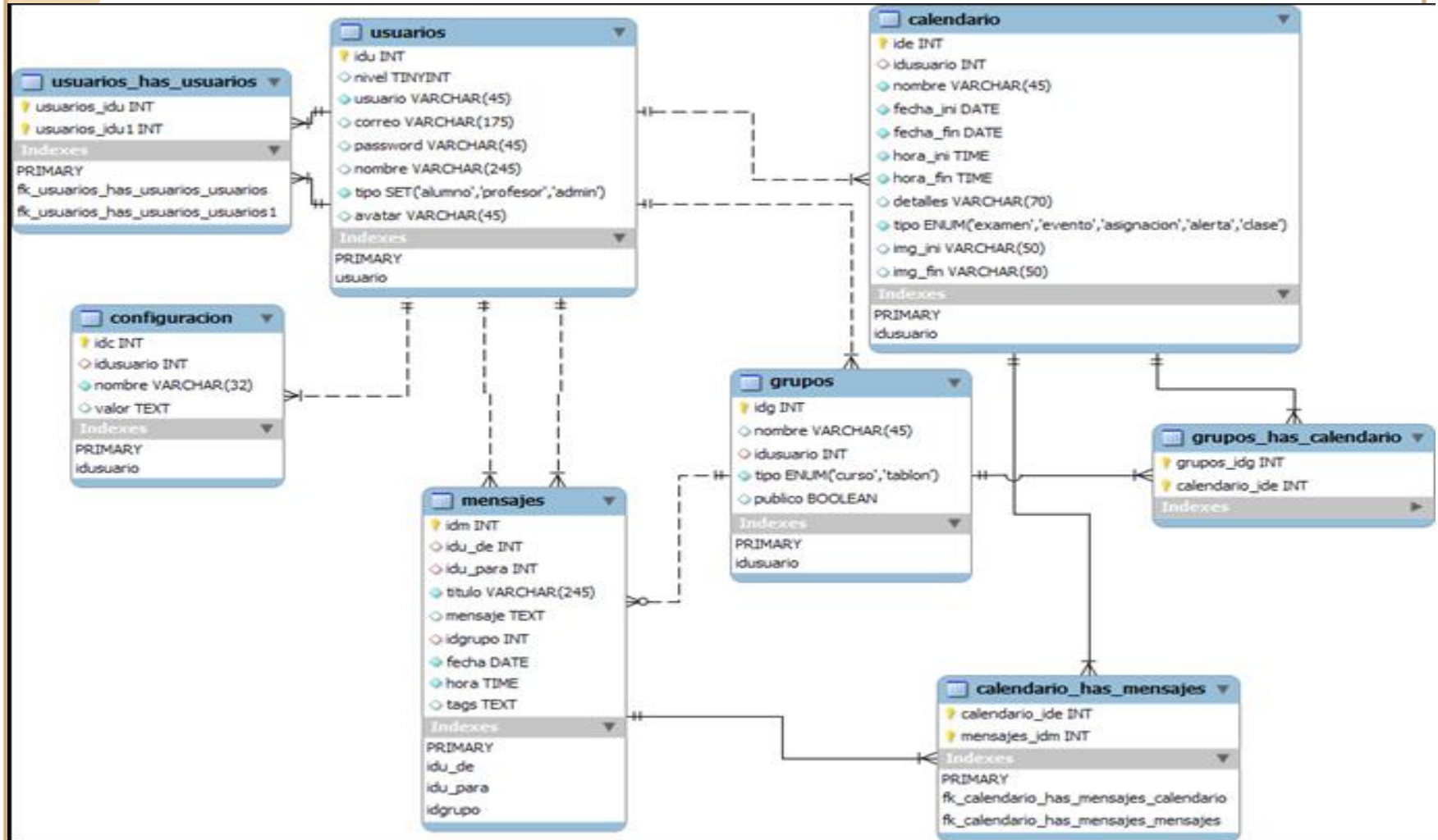
Para asimilar fácilmente un diseño de datos cuando se emplea el modelo E/R se utilizan elementos gráficos que permiten representar el modelo de datos tal como se muestra a continuación:



# Ejemplo de un Entidad-Relación de un Restaurante

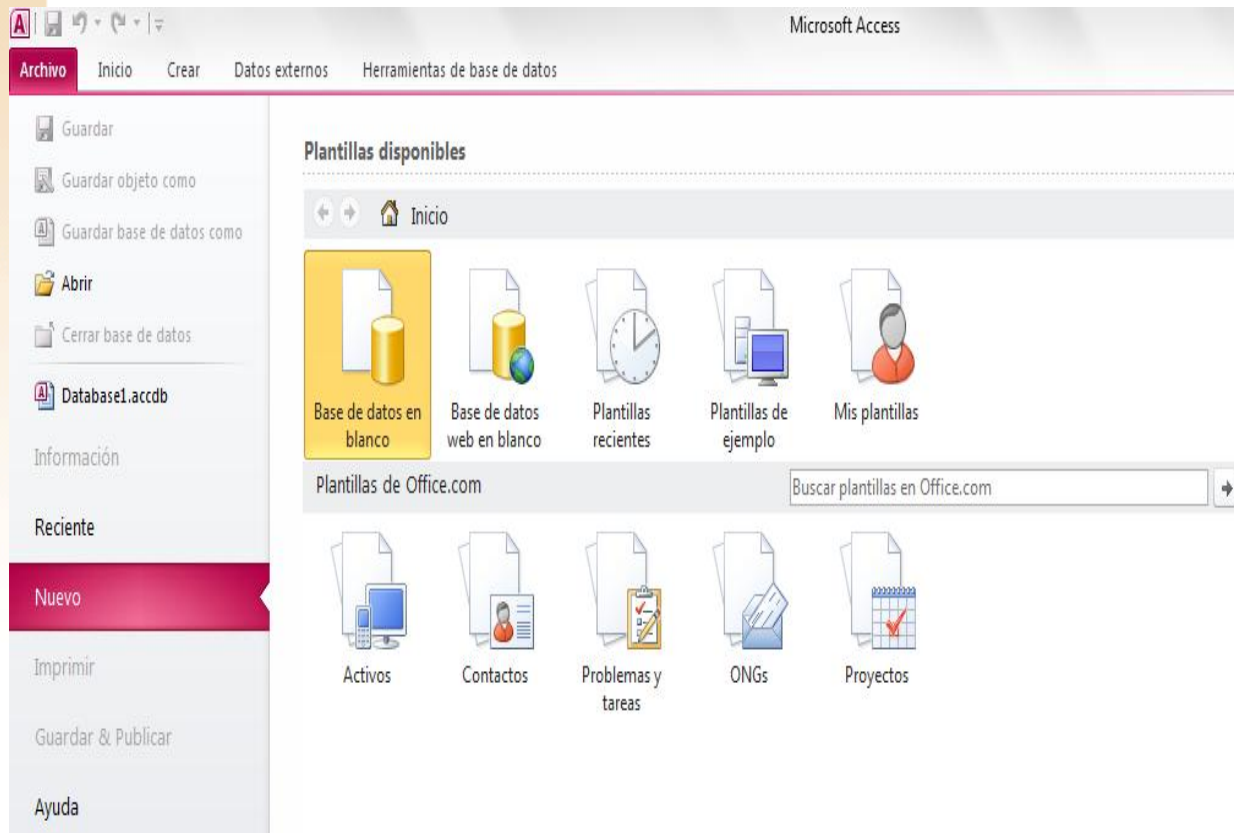


# Ejemplo de E/R de mensajería para usuarios de un sistema



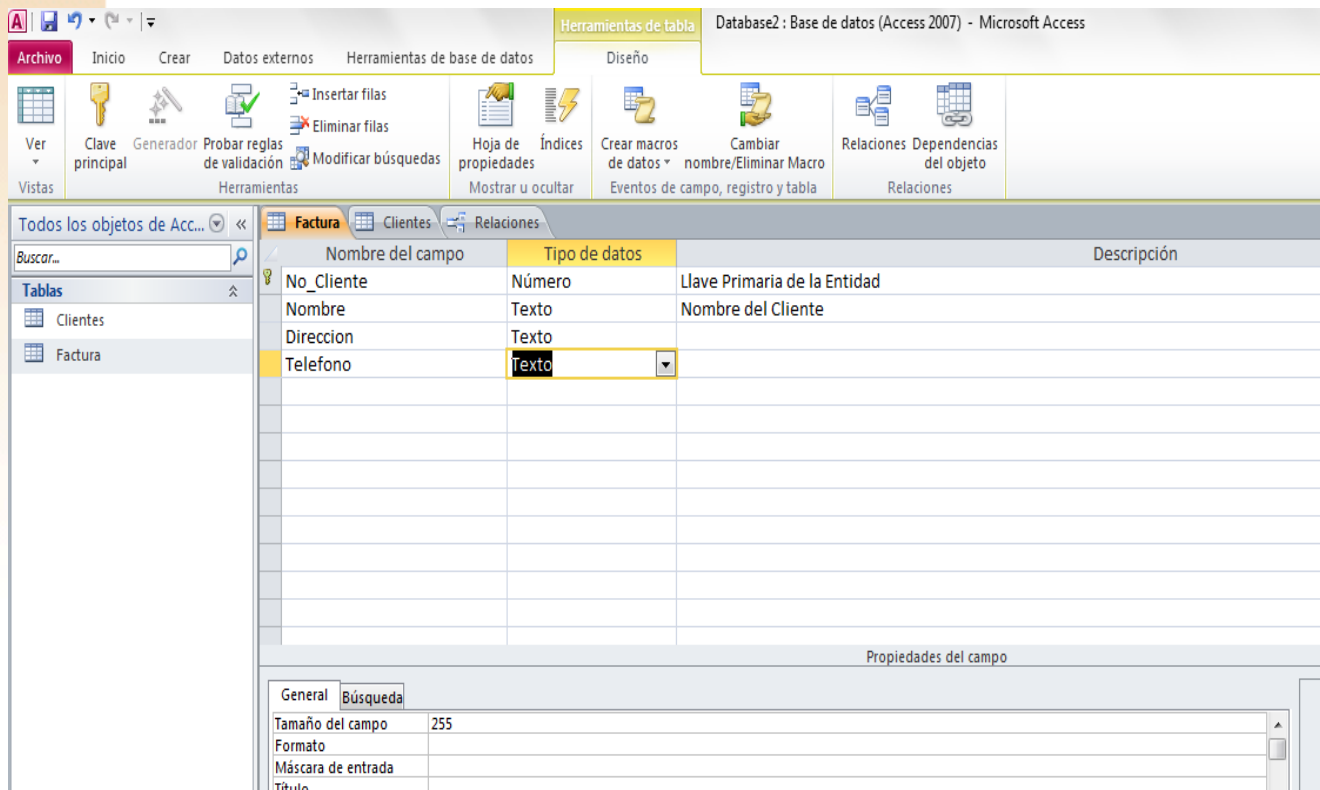
# Como crear una Base de Datos En Access

- Lo primero que debe hacerse es crear la Base de datos que contendrá las entidades, atributos y relaciones que almacenarán los datos



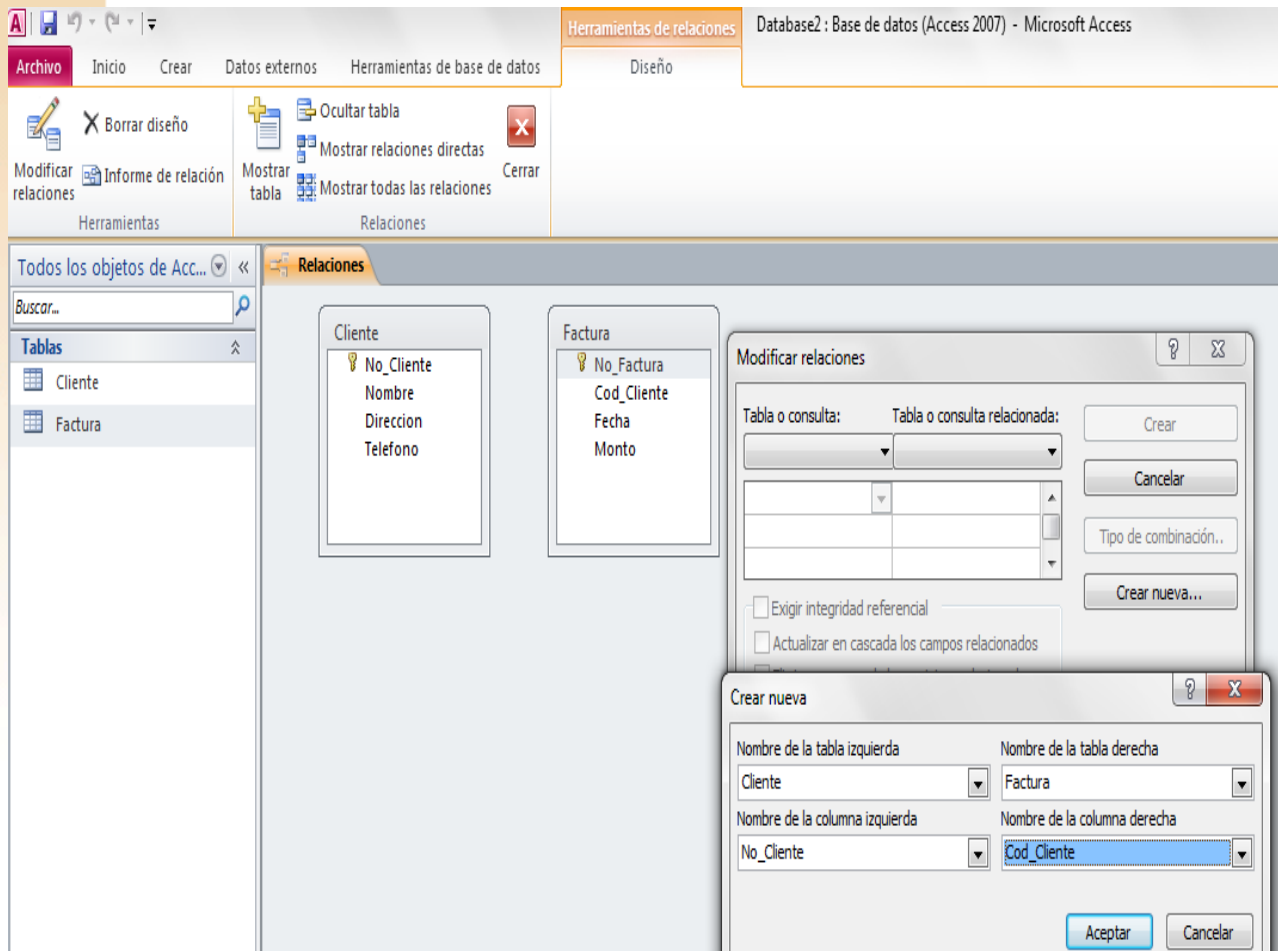
# Creación de Entidades y Atributos

- Una vez creada la base de datos se procede a crear las entidades y atributos de nuestra BD, para posteriormente proceder a crear las relaciones entre ellas.





# Creación de Relaciones





# Creación de Relaciones

The screenshot displays the Microsoft Access interface for creating relationships. The top ribbon shows the 'Herramientas de relaciones' (Relationship Tools) tab. The 'Relaciones' (Relationships) pane on the left shows a diagram with two tables: 'Cliente' and 'Factura'. 'Cliente' has fields 'No\_Cliente', 'Nombre', 'Direccion', and 'Telefono'. 'Factura' has fields 'No\_Factura', 'Cod\_Cliente', 'Fecha', and 'Monto'. A line connects 'No\_Cliente' in 'Cliente' to 'Cod\_Cliente' in 'Factura'. The 'Modificar relaciones' (Modify Relationships) dialog box is open, showing the relationship between 'No\_Cliente' and 'Cod\_Cliente'. The dialog includes options for 'Exigir integridad referencial' (Enforce referential integrity), 'Actualizar en cascada los campos relacionados' (Cascade updates to related fields), and 'Eliminar en cascada los registros relacionados' (Cascade delete of related records). The 'Tipo de relación' (Relationship type) is set to 'Uno a varios' (One to many).



# Investigación

- Investigar que es un diccionario de datos
- Investigación de Normalización.

