



Base de Datos__

Conceptos básicos
Diseño de Base de datos

Base de Datos

¿dato = información?

Para debatir en clase

Conceptos fundamentales

¿Qué es un dato?

Un dato puede ser una letra, número, símbolo o palabra que por sí solo no tiene ningún significado.

¿Qué es la información?

Se define como un conjunto de datos procesados que tienen significado para el usuario.



¿Qué es una base de datos?

Para debatir en clase

¿Qué es una base de datos?

Es un repositorio en el que se almacenarán los datos relacionados entre sí, almacenados sistemáticamente para su posterior procesamiento para entregar información al usuario.

No sólo contiene los datos sino que, también almacena una descripción de dichos datos. Los **metadatos** los cuales se almacenan en el diccionario de datos.

Clasificación de las bases de datos

1 Según la variable de la base de datos

1-1 Bases de datos estáticos

1-2 Bases de datos dinámicas

2 Según el contenido

2-1 Bases de datos bibliográficas

2-2 Bases de datos de datos completo

2-3 Bases de datos de información química o biológica

Tipos bases de datos

- 1 Bases de datos jerárquicas
- 2 Bases de datos de red
- 3 Bases de datos transaccionales
- 4 Bases de datos relacionales
- 5 Bases de datos multidimensionales
- 6 Bases de datos orientada a objetos
- 7 Bases de datos documentales
- 8 Bases de datos deductivas
- 9 NoSQL

**Vamos a
trabajar**

¿SGBD = Base de Datos?

Enumera ejemplos



ORACLE[®]
DATABASE



Para debatir en clase

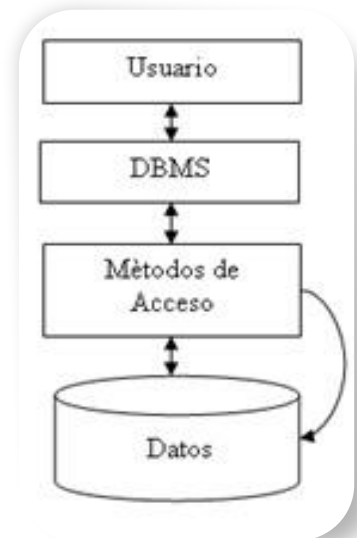
¿Qué es un Sistema de Gestión de base de datos (DBMS)?

Es un software que permite la creación y administración de Base de Datos y además actúa como un intermediario entre los usuarios, las aplicaciones y la propia base de datos.



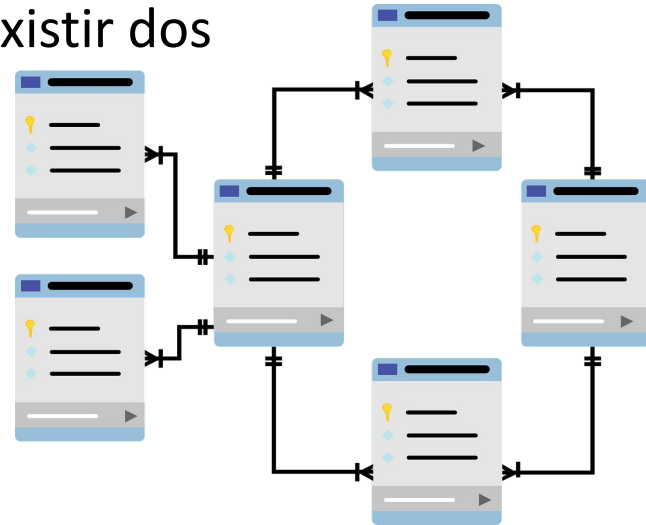
El **DBMS** proporciona los siguientes servicios que permiten:

- Definir la base de datos;
- La inserción, actualización, eliminación y consulta de los datos;
- El acceso controlado a la base de datos;
- Creación de copias de seguridad;
- Exportación e importación de datos;
- Creación de procedimientos almacenados;
- Creación de vistas; Etc.



Base de datos relacional

Las **bases de datos relacionales** son aquellas que están compuestas por distintas tablas relacionadas entre sí mediante **claves primarias** y **claves foráneas**, en donde no pueden existir dos tablas con el mismo nombre y las mismas se componen de un conjunto de **campos** o **columnas** y **registros** o **filas**.



Las 12 reglas de Codd

Son un conjunto de normas establecidas por Edgar Codd, para que una base de datos del modelo relacional se considere verdaderamente relacional, teniendo en cuenta que, a mayor cantidad de reglas cumplidas, *más relacional* será la base de datos.

Algunas de las reglas de CODD

Regla 0 - Fundación

Regla 1 - Información

Regla 2 - Acceso garantizado

Regla 3 - Tratamiento sistemático de los valores nulos

Regla 4 - Catálogo en línea relacional

Regla 5 - Sublenguaje de datos completo

Regla 6 - Actualización de vistas

Regla 7 - Inserciones, modificaciones y eliminaciones de alto nivel

Regla 8 - Independencia física de los datos

Regla 9 - Independencia lógica de los datos

Regla 10 - Independencia de integridad

Regla 11 - Independencia de distribución

Regla 12 - No subversión

Para ver más sobre el tema

https://es.wikipedia.org/wiki/12_reglas_de_Codd

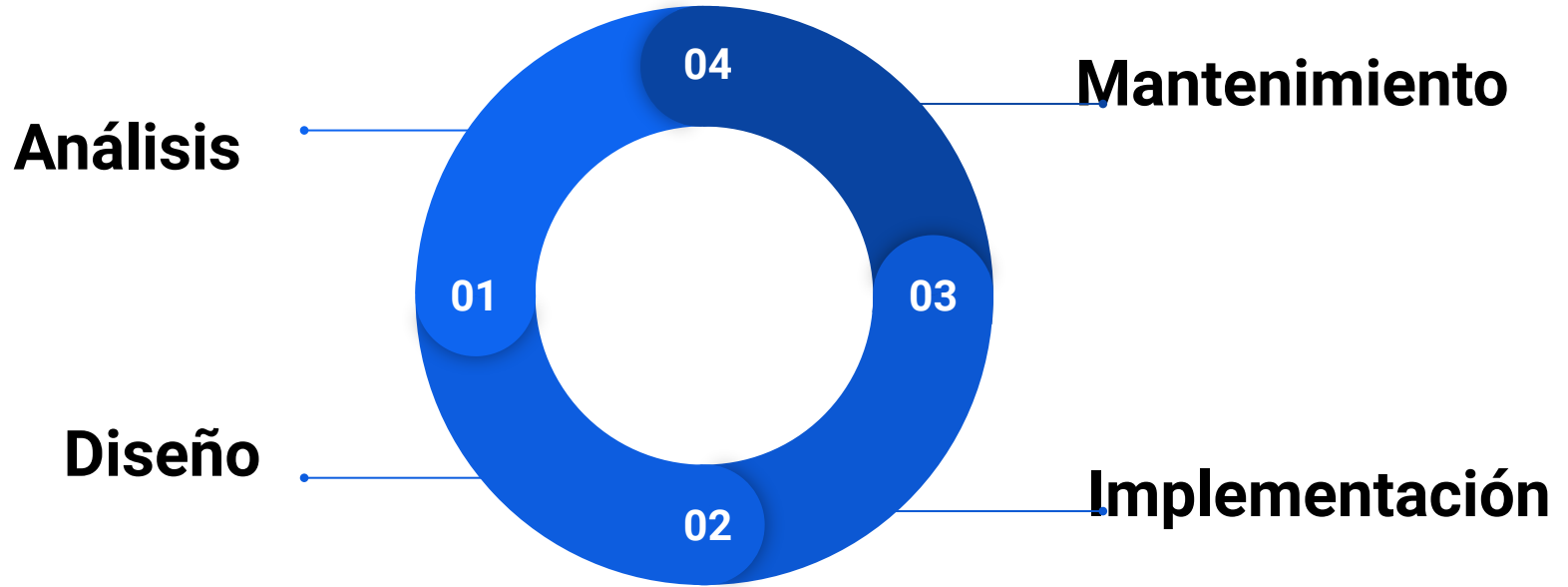
SQL (Structured Query Language/ Lenguaje Estructurado de Consultas)

Es un lenguaje informático de alto nivel que sirve para gestionar y organizar los datos de una base de datos relacional.

Funciona con un tipo específico de base de datos, llamado *base de datos relacional*.



Al trabajar en la bases de datos también aplicamos



Especificación de requerimientos

DISEÑO CONCEPTUAL

Esquema Conceptual (DER)

DISEÑO LÓGICO

Normalización

Esquema Lógico (Modelo Relacional)

DISEÑO FÍSICO

Esquema físico

El **diseño** conceptual surge a partir de las especificaciones de requisitos del usuario, de allí se obtiene el **esquema** conceptual de la base de datos. Este esquema define la estructura de la base de datos, independientemente de la tecnología o el sistema de gestión de base de datos que se utilice. Distinto al esquema, el **modelo** conceptual es el lenguaje que se utilizará para describir el esquema conceptual.

El objetivo de este diseño es describir la información de la base de datos en sí y no de cómo se almacenará la información en ésta.

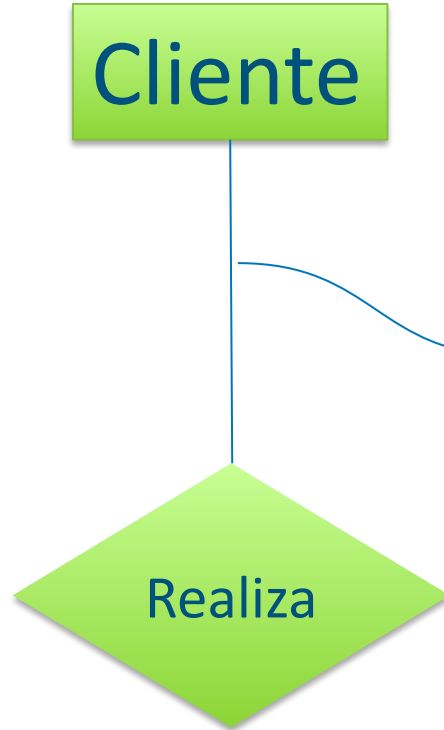
El resultado del diseño lógico se sitúa ya en el mundo de las representaciones.

Diagrama Entidad Relación

Representa la relación que tienen las entidades entre sí. Una entidad se puede decir que es una unidad que tiene atributos y contiene información que conforma una base de datos, siendo una representación de objeto, persona, cosa, etc.

En el DER, la entidad se representa con un rectángulo al que conectamos a sus atributos y a las relaciones mediante una línea.

Cada relación se representa con un rombo y así, dos o más entidades se pueden conectar mediante una misma relación en común indicando la *cardinalidad*. En cambio los atributos se representan mediante un círculo.



Entidad.

Tipo de objeto sobre el cual se obtiene información: una cosa, una persona, etc.

Sólo pueden aparecer una vez en el esquema.

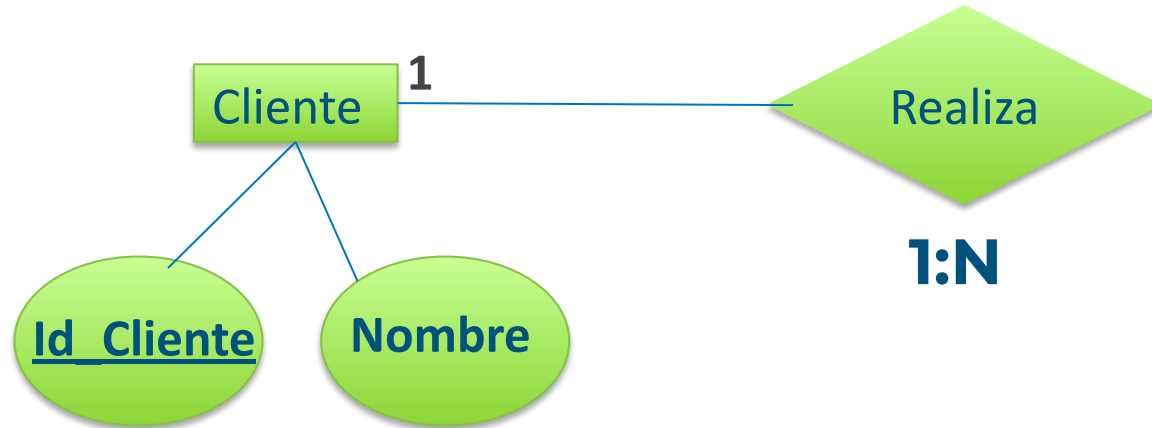
Unión

Relación

Relación o asociación entre dos o más entidades.

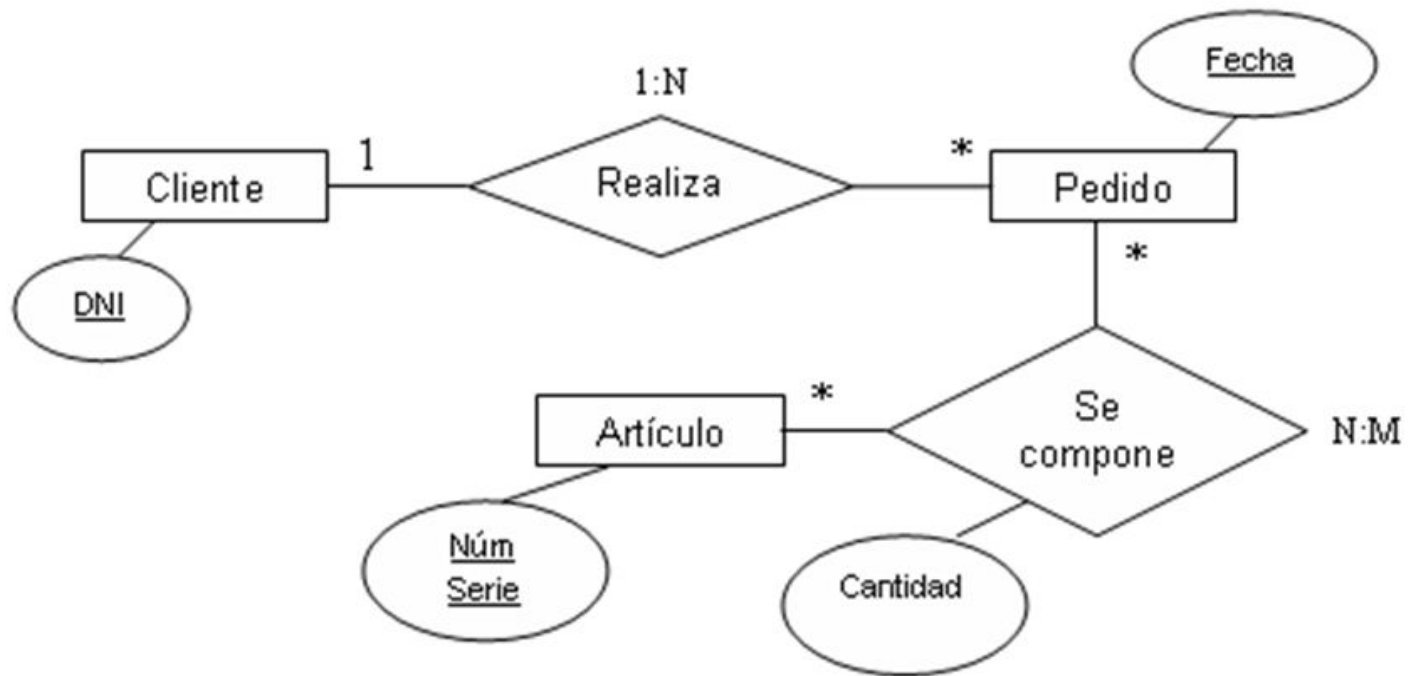
La *cardinalidad* o *multiplicidad* indica la cantidad de elementos de una entidad que se relacionan con una instancia de otra entidad.

La **cardinalidad** de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ocurrencia de la entidad o relación a la que pertenece. El valor por omisión es (1,1)



Atributo

Característica de interés sobre una entidad o relación.



Trabajo en Grupos

Diseño Conceptual - DER

Especificación de requerimientos

DISEÑO CONCEPTUAL

Esquema Conceptual (DER)

DISEÑO LÓGICO

Normalización

Esquema Lógico (Modelo Relacional)

DISEÑO FÍSICO

Esquema físico

El **diseño lógico** es parte del esquema conceptual y da como resultado la estructura de la base de datos, definiendo las estructuras de datos que pueden soportar los sistemas gestores de bases de datos. A este esquema se lo llama **esquema lógico**.

El *modelo* lógico es el lenguaje que determina el esquema lógico.

En cuanto al diseño lógico, se determina según el SGBD que se vaya a utilizar sin depender del producto concreto.

Diseño Lógico

Luego de obtener nuestro esquema conceptual y teniendo en cuenta los puntos anteriormente mencionados, podremos obtener nuestro diseño lógico, haciendo la transformación del DER a un modelo relacional siguiendo las 12 reglas de Cood y las Reglas de normalización.

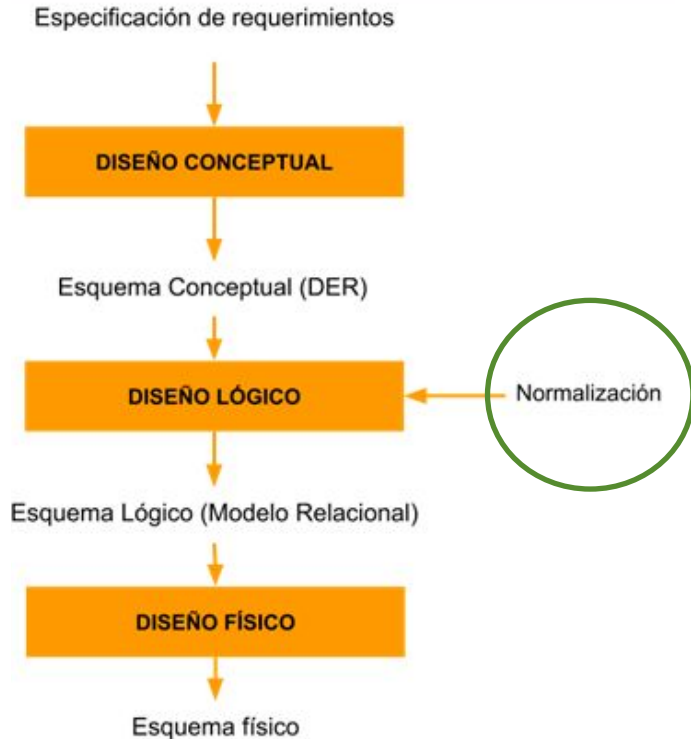
Diseño Lógico

¿Cómo paso del Diagrama Entidad - Relación al Modelo Relacional?

Para ello, seguir los siguientes pasos:

1. Toda entidad se transforma en una tabla.
2. Todo atributo se transforma en una columna dentro de la tabla a la que pertenece.
3. El identificador de la entidad se convierte en la **clave primaria** de la tabla. Si no existe, se crea una. Ej. Id_estudiante.
4. Toda relación N:M (muchos a muchos) se convierte en una tabla que tendrá como clave primaria la **dos claves primarias** de las entidades que se asocian.
5. En las relaciones 1:N (uno a muchos) la clave primaria de la entidad con cardinalidad 1 pasa a la tabla de la entidad cuya cardinalidad es N (lo que sería una **clave foránea**).
6. En las relaciones 1:1 (uno a uno) las dos tablas en cuestión se fusionan en una sola. Se deberá definir un identificador como clave primaria común a ambas.

Para comprender mejor consultar el libro Bases de Datos de Mercedes Marqués (página 125 a 137)

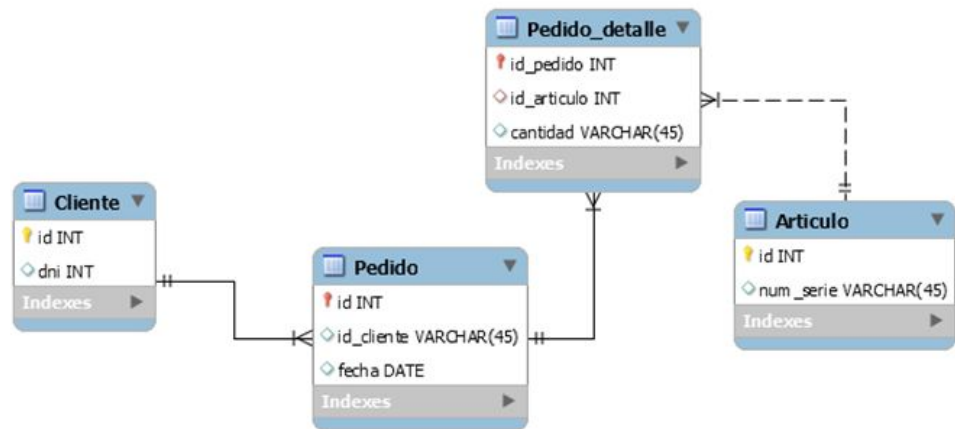
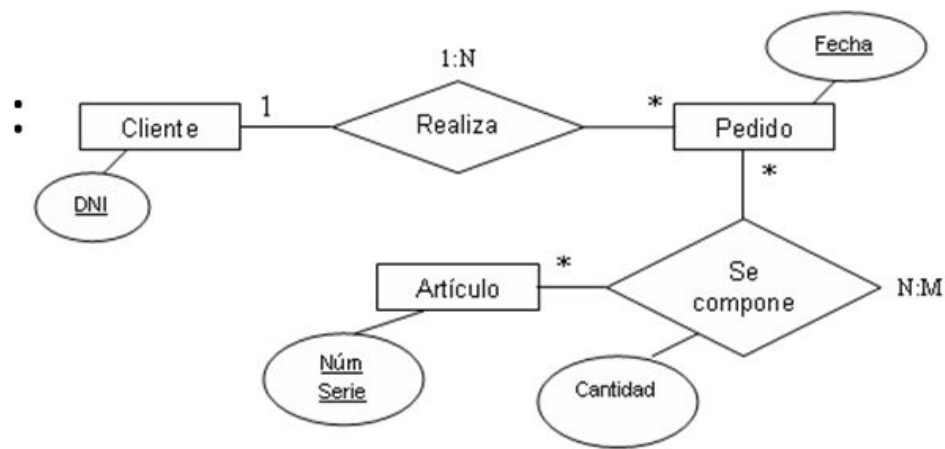


La normalización es la técnica de estandarización y validación de datos que se utiliza para diseñar las tablas y establecer las relaciones entre ellas. Esto nos permite una mayor organización para eliminar la redundancia de datos y proteger la integridad de los mismos.

Sin la normalización, tendríamos datos redundantes los cuales traen aparejados problemas de mantenimiento, además de ocupar espacio en el disco duro, como también, dependencias incoherentes.

Cinco formas de normalización (FN: Forma normal)

- 1FN: Todos los datos son atómicos
- 2FN: Los atributos no claves dependen de la clave
- 3FN: Eliminar columnas que no depende de clave
- 4FN: Aislar Relaciones Múltiples Independientes
- 5FN: Aislar relaciones semánticamente relacionadas múltiples.



1FN

Una tabla está en primera forma normal cuando todos los datos son atómicos (es decir, cada atributo tiene su propia celda) y todas las columnas contienen el mismo tipo de datos. Sirve para eliminar los grupos repetidos.

2FN

Para estar en la segunda forma normal, a las condiciones de la primera forma normal se le debe agregar que los atributos que no forman parte de ninguna clave, su funcionamiento depende de la clave primaria. Ésta sirve para eliminar los datos redundantes.

3FN

Para cumplirse la tercera forma normal, a las dos condiciones anteriores hay que agregarle que los atributos que no son clave no pueden depender de manera transitiva de una clave candidata. Aquí se deben eliminar las columnas que no dependen de la clave principal.

Id Cliente	Apellido	Nombre	Barrio	Calle	Numero	CP	Localidad	Provincia
235	López	Celia		San Martín	550		Esquel	Chubut
458	Rosales	Daniel	Don Basco	Antártida Arg.	450		Esquel	Chubut
789	González	Josefina		Av. Colón	458	5000	Córdoba	Córdoba
899	González			Av. Alem	4258		Santa María	Córdoba

Id Cliente	Teléfono
235	2945-452888
235	2945-588778
789	351-415236
899	351-569987
899	351-556877

Tablas normalizadas 1FN

Id Cliente	Nombre	Domicilio	Teléfono
235	López Celia	San Martín 550 Esquel	2945-452888 2945-588778
458	Rosales Daniel	B° San Martín, Antártida Arg. 450 Esquel Chubut	2945-400587
789	González Josefina	Av. Colón 458 cp.5000 Córdoba	351-415236
899	González	Av. Alem 4258	351-569987 351-556877

Id Cliente	Nombre	Domicilio	Teléfono 1	Teléfono 2
235	López	San Martín 550	2945-452888	2945-588778
458	Rosales	B° Don Basco, San Martín 456	2945-400587	
789	González	Av. Colón 458 cp.5000	351-415236	
899	González	Av. Alem 4258	351-569987	351-556877

La **primera forma normal** significa que los datos están en un formato de entidad, lo que significa que se han cumplido las siguientes condiciones:

- Eliminar grupos repetidos en tablas individuales
- Crear una tabla independiente para cada conjunto de datos relacionados
- Identificar cada conjunto de relacionados con la clave principal

Id Cliente	Apellido	Nombre	Id Localidad	Id Provincia
235	López	Celia	1	1
458	Rosales	Daniel	1	1
789	Anns	Josefina	2	2
899	Roman	José	3	2

Id Localidad	Nombre	Código Postal
1	Esquel	9200
2	Córdoba	5000
3	Santa María	5236

Id Provincia	Nombre
1	Chubut
2	Córdoba

Id Cliente	Apellido	Nombre	Localidad	Código Postal	Provincia
235	López	Celia	Esquel	9200	Chubut
458	Rosales	Daniel	Esquel	9200	Chubut
789	Gonzales	Josefina	Córdoba	5000	Córdoba
899	Gonzales	José	Santa María	5236	Córdoba

Tablas normalizadas 2FN

La **segunda forma normal** asegura que cada atributo describe la entidad. Por ello, se debe crear tablas separadas para el conjunto de valores y los registros múltiples, estas tablas se deben relacionar con una clave externa.

Los registros no deben depender de otra cosa que la clave principal de la tabla, incluida la clave compuesta si es necesario.

La **tercera forma normal** comprueba las dependencias transitivas, eliminando campos que no dependen de la clave principal. Por ello, los campos que no pertenecen a la clave principal hay que colocarlos en una tabla aparte y relacionen ambas tablas por medio de una clave externa.

Para la **cuarta forma normal** las dependencias multivaluadas son aquellas donde la existencia de dos o más relaciones N:M (Muchos a muchos) causa redundancia. Para evitar eso, utilizamos la 4FN.

La **quinta forma normal** existe, pero rara vez se consideran en el diseño práctico.

El no tener en cuenta estas dos reglas de normalización adicionales puede resultar en un diseño de base de datos menos perfecto pero no debería afectar a la funcionalidad

La normalización de base de datos es un punto muy importante que deberíamos de tomar muy en serio para establecer cimientos sólidos sobre los cuales podemos construir aplicaciones robustas que en el futuro no presenten problemas de base de datos difíciles de solucionar.

La **tercera forma normal** comprueba las dependencias transitivas, eliminando campos que no dependen de la clave principal. Por ello, los campos que no pertenecen a la clave principal hay que colocarlos en una tabla aparte y relacionen ambas tablas por medio de una clave externa.

Id Cliente	Apellido	Nombre	Id Localidad
235	López	Celia	1
458	Rosales	Daniel	1
789	Anns	Josefina	2
899	Roman	José	3

Id Cliente	Apellido	Nombre	Id Localidad	Id Provincia
235	López	Celia	1	1
458	Rosales	Daniel	1	1
789	Anns	Josefina	2	2
899	Roman	José	3	2

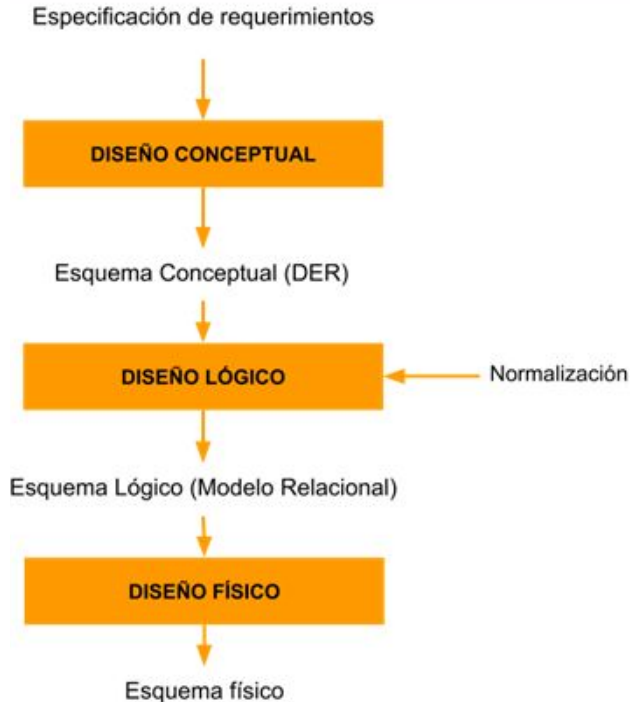
Id Localidad	Nombre	Código Postal	Id Provincia
1	Esquel	9200	1
2	Córdoba	5000	2
3	Santa María	5236	2

Id Provincia	Nombre
1	Chubut
2	Córdoba

**Tablas
normalizadas 3FN**

Trabajo en Grupos

Diseño Conceptual - Modelo Relacional



Diseño Físico

Para obtener nuestro diseño físico, se parte del modelo relacional definido en la fase anterior con el objetivo de conseguir una mayor **eficiencia, performance y la seguridad de los datos** contemplando ahora, los aspectos de **implementación física** que dependen del SGBD. describe la implementación de las estructuras de almacenamiento y los métodos para tener un acceso eficiente a los datos. Este, además, depende del SGBD que se va a utilizar y el esquema físico se expresa mediante su LDD (lenguaje de definición de datos).

Uno de los objetivos principales del diseño físico es almacenar los datos de modo eficiente. Para medir la eficiencia hay varios factores que se debe tener en cuenta:

- **Rendimiento de transacciones.** Es el número de transacciones que se quiere procesar en un intervalo de tiempo.
- **Tiempo de respuesta.** Es el tiempo que tarda en ejecutarse una transacción. Desde el punto de vista del usuario, este tiempo debería ser el mínimo posible.
- **Espacio en disco.** Es la cantidad de espacio en disco que hace falta para los ficheros de la base de datos. Normalmente, el diseñador querrá minimizar este espacio.

Modeladores

Existen modeladores, como los de los link debajo que permiten generar los diagramas de entidad - relación o de clases y convertir a código, por ejemplo sql.

<https://cruise.umple.org/umpleonline/>

<https://umbrello.kde.org/>

