



## CICLO 2

[FORMACIÓN POR CICLOS]

# Diseño básico de bases de datos **RELACIONALES**




Ingeni@  
Soluciones TIC



UNIVERSIDAD  
DE ANTIOQUIA

Facultad de Ingeniería



En una lectura anterior abordamos la importancia del diseño y del modelado en la programación orientada a objetos. El modelado de una aplicación, y principalmente de su dominio, se realiza generalmente usando diagramas de clases UML. Este tipo de diagramas es, de hecho, el estándar para dicho modelado. Recordemos que el modelado es un proceso importante en el desarrollo de *software*, el cual nos permite abstraer los aspectos relevantes del dominio de un problema dejando por fuera todo lo demás, de tal manera que podamos darle una solución apropiada al mismo.

Al momento de crear una base de datos relacional, es importante definir, en primer lugar, su estructura. La estructura define desde el principio la información que se va a guardar, en qué tablas se va a distribuir, los campos que tendrá cada tabla y otros aspectos relevantes.

Una forma de definir la estructura mencionada es mediante su modelado. Por lo general, dicho modelado se realiza con un diagrama gráfico que sigue unas reglas específicas, a las cuales se las denomina **notación**. La notación indica las figuras a utilizar, cómo se plasman con dichas figuras los aspectos relevantes de lo que queremos modelar y las restricciones a las que nos sometemos.

Para el modelado de bases de datos relacionales se han propuesto diferentes diagramas. Cuál diagrama usar depende mucho de en qué etapa del diseño de la base de datos nos encontramos. Por ejemplo, al momento de hacer el diseño conceptual de la base de datos, se suele usar, entre otros, el modelo entidad-relación (E-R). Este es un tipo de modelo básico para el que se han propuesto diferentes notaciones, así como modificaciones a dichas notaciones.

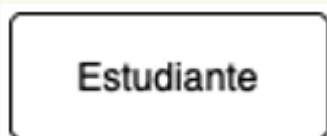


## Modelo entidad-relación

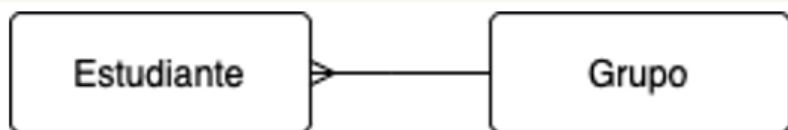
Un **modelo entidad-relación** (diagrama entidad-relación, E-R o ERD) es un tipo de diagrama usado, entre otros propósitos, para el modelado de bases de datos relacionales, y que consta de un conjunto de entidades que se relacionan entre sí. Una **entidad** representa un conjunto de personas, instituciones, objetos y, en general, conceptos susceptibles a tener atributos. Intuitivamente, son similares a las clases en el diseño orientado a objetos y, por lo tanto, en los diagramas de clases UML. Se puede decir que las entidades sirven como plantillas abstractas de una o varias instancias, de manera similar a las clases en la programación orientada a objetos. Una **instancia** sería una persona, institución u objeto en particular. Las **relaciones** entre entidades se representan mediante líneas que incluyen figuras geométricas que indican la naturaleza de la relación. A continuación, vamos a hacer un repaso de los elementos de un modelo entidad-relación usando una de sus notaciones más usadas.

### Entidades

Los elementos clave dentro de un diagrama entidad-relación son, precisamente, las entidades. Una entidad representa, además de lo definido anteriormente, algo de lo cual vamos a guardar información en la base de datos. En un modelo entidad-relación, una entidad se representa mediante un rectángulo ovalado, como el que apreciamos a continuación.



La entidad anterior representa el concepto de estudiante, es decir, un conjunto (abstracto) de estudiantes. Sin embargo, por lo general una base de datos no consta de un solo concepto. Por esto, podemos tener en un solo diagrama varias entidades relacionadas entre sí. A continuación, podemos ver cómo la entidad Estudiante se relaciona con otra entidad, en este caso la entidad Grupo.

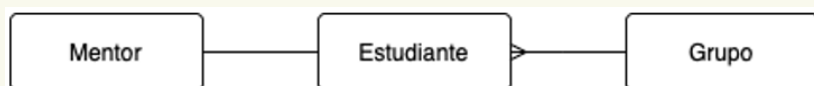


## Relaciones

Notemos que la relación consta de una línea que conecta ambas entidades, pero en uno de sus extremos se dibuja con tres líneas adicionales que simulan una “pata de gallina” (y así se las denomina). Eso quiere decir que se trata de una relación **uno-a-muchos**, que se puede leer en dos direcciones, así:

- Un estudiante debe estar matriculado en un grupo.
- Un grupo debe tener matriculados varios estudiantes.

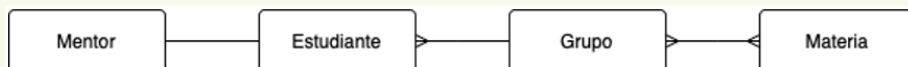
También tenemos relaciones **uno-a-uno**, en las que basta con una sola línea, como a continuación:



En el caso anterior, hay una relación uno-a-uno entre Estudiante y Mentor, la cual se puede leer también en dos direcciones, así:

- Un mentor asesora a un estudiante.
- Un estudiante cuenta con un mentor.

Así también, contamos con relaciones **muchos-a-muchos**. En este caso, como podremos imaginar, la pata de gallina se presenta en ambos sentidos, como podemos apreciar a continuación:

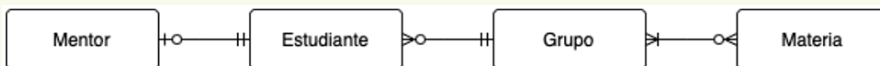




En el anterior diagrama podemos apreciar que hay una relación muchos-a-muchos. Es decir, que un grupo se relaciona con muchas materias, y que una materia se relaciona con muchos grupos. Para nuestro caso, podríamos leer esa parte del diagrama, en dos direcciones, así:

- En un grupo se imparten varias materias.
- Una materia es impartida en varios grupos.

Partiendo del diagrama anterior, es posible que las relaciones sean más expresivas. Es decir, que una relación entre dos entidades suministre más información usando figuras geométricas adicionales. Para ilustrar lo anterior, vamos a modificar dicho diagrama agregando un círculo o una línea perpendicular, según sea el caso, a los extremos de las relaciones:



El modelo inmediatamente anterior es similar al que ya hemos analizado antes. Sin embargo, las relaciones entre las entidades tienen figuras geométricas que proporcionan información adicional, así:

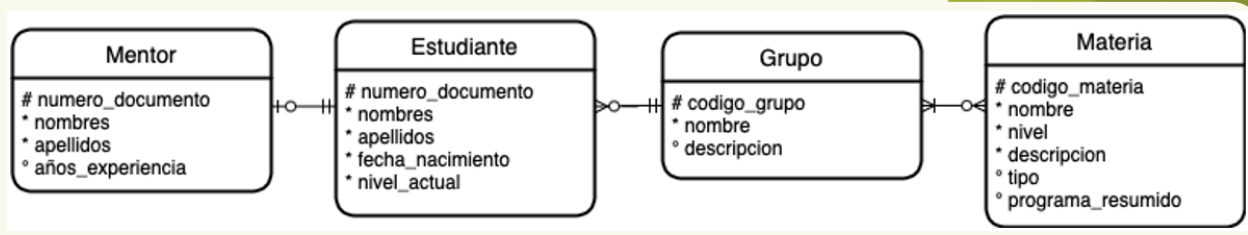
- Si la relación termina en “pata de gallina”, como observamos antes, se trata de una relación de “muchos”.
- Si la relación termina con una o dos líneas perpendiculares, se trata de una relación **obligatoria**.
- Si el final de la relación incluye un círculo, se trata de una relación **condicional**.

Lo anterior puede variar dependiendo de la notación que se elija para realizar el modelado. Así que es común encontrar diagramas entidad-relación con símbolos similares que varíen en su significado o símbolos totalmente diferentes, así como elementos semánticos adicionales, como la cardinalidad.

Observando detalladamente el diagrama anterior, y teniendo en cuenta los símbolos adicionales, este se podría leer de la siguiente manera:

- Un mentor debe asesorar a un estudiante (relación obligatoria).
- Un estudiante puede contar con un mentor (relación opcional).
- Un estudiante debe pertenecer a un grupo (relación obligatoria).
- Un grupo puede tener entre cero y múltiples estudiantes (relación opcional).
- En un grupo se pueden impartir cero o varias materias (relación opcional).
- Una materia se debe impartir en uno o varios grupos (relación obligatoria).


A la hora de modelar una base de datos, no solo las entidades y sus relaciones son relevantes, sino también los atributos de dichas entidades. El cómo se especifican dichos atributos en el diagrama depende también de la notación elegida para el modelado. Usando la notación con la que venimos trabajando en la presente lectura, podríamos agregar atributos a las entidades de la siguiente manera:



El diagrama inmediatamente anterior es el mismo en el que hemos venido trabajando a lo largo de la lectura, pero con aún más detalle. Podemos observar que ahora cada entidad cuenta con atributos. Cada atributo consiste en su nombre, anteponiéndole un símbolo, que puede ser uno de los siguientes:

- Si se antepone un signo número (#), quiere decir que ese atributo es el **identificador único** de la entidad. Es decir, que una instancia de dicha entidad se puede distinguir de otra diferente por ese atributo, que será diferente para todas las instancias. Un ejemplo de esto es el documento de identidad de una persona o la referencia de un producto.





- Si se antepone un asterisco (\*) o un punto (.), quiere decir que ese atributo siempre deberá ser conocido para cualquier instancia de la entidad. Es decir, que es **obligatorio**. Por ejemplo, en una base de datos de clientes, por lo general el nombre del cliente suele ser un atributo obligatorio.

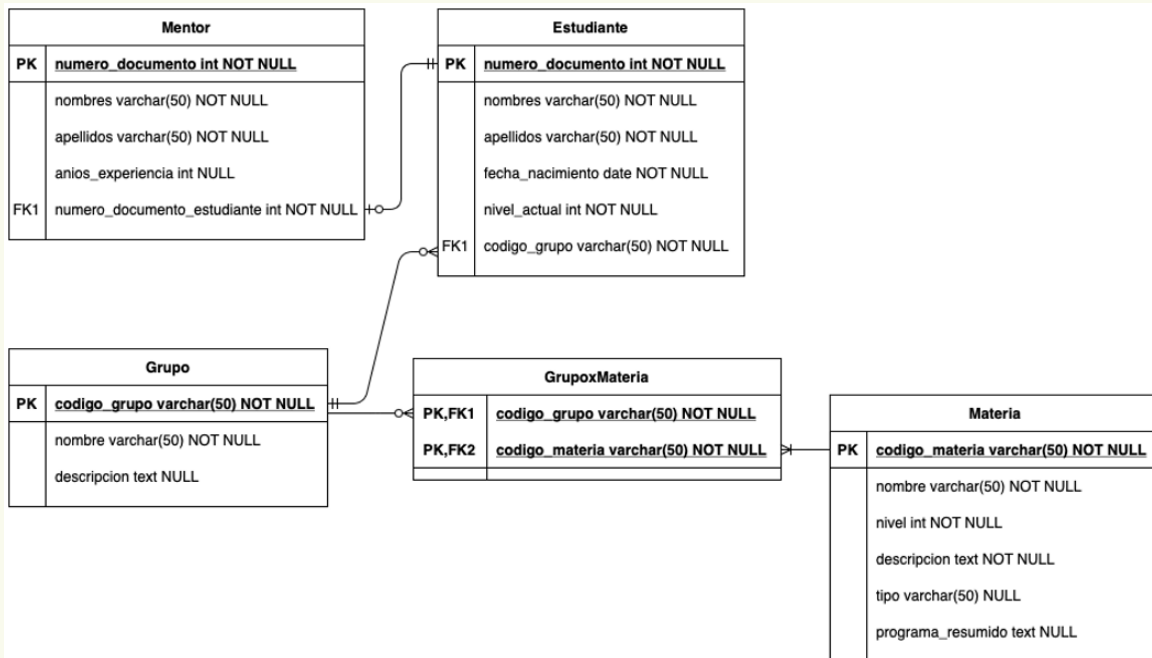
- Si se antepone un círculo vacío (°), quiere decir que ese atributo no siempre se debe conocer y, por lo tanto, es **opcional**. Por ejemplo, en una base de datos de proveedores podríamos querer guardar la descripción de algunos proveedores, pero no de otros, por lo que ese atributo pasa a ser opcional.

De acuerdo con lo anterior, en el diagrama que hemos construido hasta ahora, tenemos lo siguiente:

- Una entidad Mentor cuyo identificador único es el documento de identidad, tiene como atributos obligatorios los nombres y los apellidos, y como atributos opcionales los años de experiencia del mentor.
- Una entidad Estudiante, cuyo identificador único es, de nuevo, el número de documento, y tiene sus cuatro atributos obligatorios: nombres, apellidos, fecha de nacimiento y nivel actual.
- Una entidad Grupo. Esta tiene como identificador único el código del grupo, su nombre como atributo obligatorio y su descripción como atributo opcional.
- Y una entidad Materia. Su código es el identificador único; sus atributos obligatorios son el nombre, el nivel y la descripción; y tiene como atributos opcionales el tipo y el programa resumido de la materia.

Como dijimos anteriormente, el modelado de una base de datos se realiza mediante diferentes modelos o diagramas, y para cada diagrama se pueden usar diferentes notaciones. La selección de la notación a utilizar depende, entre otras cosas, de para qué vayamos a utilizar el diagrama y de la expresividad que necesitemos del mismo. La notación anterior especifica las entidades, sus relaciones y atributos. Sin embargo, podría ser necesario que el diagrama sea más expresivo e incluso más cercano a lo que implementaremos posteriormente en el sistema de gestión de bases de datos (SGBD). A continuación, veamos atentamente y en detalle un ejemplo del mismo diagrama de arriba, pero usando otra notación.






En la notación que podemos observar arriba, las relaciones entre entidades no varían en su significado, pero se pueden apreciar varias diferencias y elementos adicionales.

- La forma de las entidades ya no es rectangular ovalada, sino completamente rectangular.
- Los atributos que son identificadores únicos, que se denominan en los SGBD **claves primarias** (*primary keys*), en esta notación se indican con la sigla PK.
- Las claves primarias estarán, además, en negrita y subrayadas.
- Después del nombre de cada atributo se puede ver su tipo de dato. Las entidades se implementarán posteriormente como tablas, y los atributos se implementarán como campos o columnas. Cada campo deberá ser de un **tipo de dato**, del cual dependerá lo que podamos a guardar en dicho campo. Por lo tanto, podemos observar que a cada atributo se le asignó un tipo de dato:







- o A los atributos numéricos les asignamos el tipo int.
- o A los atributos de cadenas de caracteres cortas les asignamos el tipo varchar, y entre paréntesis el número máximo de caracteres que se podrán almacenar en dicho campo para cada instancia (o registro).
- o A los atributos de cadenas de caracteres largas les asignamos el tipo text.

- Cada SGBD tiene sus tipos de datos propios, y podemos encontrar mucho más adicionales a los tres anteriores.
- Para cada atributo, luego del tipo de dato, se define si será NULL o NOT NULL. Por ahora, solo necesitamos saber que los atributos NOT NULL son los atributos marcados en la otra notación como **obligatorios** (con \*), y los atributos NULL son los atributos marcados en la otra notación como **opcionales** (con °).

- Algunos atributos están marcados con las siglas FK de *foreign key* o clave foránea. La **clave foránea** permite implementar la relación como tal entre dos entidades. Por eso, las **relaciones** entre dos entidades van, por lo general, entre una clave foránea y una clave principal. La relación indica, entonces, mediante qué clave foránea una relación se asocia con otra, y a qué clave principal corresponde. Por ejemplo, el atributo numero\_documento\_estudiante, marcado con FK1, de la entidad Mentor corresponde al atributo numero\_documento de la entidad Estudiante. De hecho, la relación sale directamente de uno de los atributos al otro. Es decir, es el número de documento del estudiante que asesora al mentor en cuestión. Las demás claves foráneas del modelo desempeñan una función similar.

El modelo anterior, en esas dos y otras notaciones, es una herramienta importante para el diseño, la comunicación y el diagnóstico conceptual de una base de datos. Una vez se usa para el diseño de una nueva base de datos nueva, se pasa a la implementación de esta en un SGBD. Obviando una gran cantidad de detalles, por lo general en dicha implementación, entre otros elementos:

- Las entidades pasan a ser **tablas** de la base de datos.
- Los atributos, pasan a ser **columnas** o campos de las tablas, con sus tipos de datos y otras propiedades.
- Las relaciones pasan a definir **claves foráneas**.

