**Modelo Entidad/Relación.**

Este modelo se obtiene en tiempo de diseño de la base de datos. Fue propuesto por Peter Chen en 1976 y desde entonces se viene utilizando de una forma muy global.

Se caracteriza por utilizar una serie de símbolos y reglas para representar los datos y sus relaciones.

Con este modelo conseguimos representar de manera grafica la estructura lógica de una base de datos.

Los principales elementos del modelo entidad-relación son las entidades con sus atributos y las relaciones entre entidades.

Las bases de datos son un gran pilar de la programación actual, ya que nos permiten almacenar y usar de forma rápida y eficiente cantidades ingentes de datos con cierta facilidad. En la actualidad se usa de forma mayoritaria las bases de datos relacionales (dominadas por distintos gestores a través del lenguaje SQL, en gran medida).

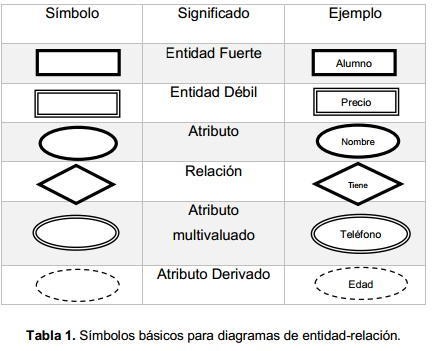
Pero ahora vamos a dar un pequeño repaso a lo más esencial del modelo entidad- relación, que es y ha sido durante años la mejor forma de representar la estructura de estas bases de datos relacionales (o de representar sus esquemas).

# ¿Qué es el modelo entidad-relación?

Este modelo es solo y exclusivamente un método del que disponemos para diseñar estos esquemas que posteriormente debemos de implementar en un gestor de *BBDD* (bases de datos). Este modelo se representa a través de diagramas y está formado por varios elementos.

Este modelo habitualmente, además de disponer de un diagrama que ayuda a entender los datos y como se relacionan entre ellos, debe de ser completado con un pequeño resumen con la lista de los atributos y las relaciones de cada elemento.

# Elementos del modelo entidad-relación



## Entidad

Las entidades representan *cosas* u *objetos* (ya sean reales o abstractos), que se diferencian claramente entre sí.

Para poder seguir un ejemplo durante el artículo añadiré ejemplos sobre un taller mecánico, donde se podría crear las siguientes entidades:

* Coches (objeto físico): contiene la información de cada taller.
* Empleado (*objeto* físico): información de los trabajadores.
* Cargo del empleado (*cosa* abstracta): información de la función del empleado. Estas entidades se representan en un diagrama con un rectángulos, como los siguientes.

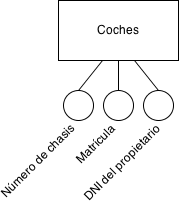
ntidades

## Atributos

Los atributos definen o identifican las características de entidad (es el contenido de esta entidad). Cada entidad contiene distintos atributos, que dan información sobre esta entidad. Estos atributos pueden ser de distintos tipos (numéricos, texto, fecha...). Siguiendo el ejemplo de antes podemos analizar los atributos de nuestra entidad "Coches", que nos darán información sobre los coches de nuestro supuesto taller.

Unos posibles atributos serían los siguientes: *número de chasis*, *matrícula*, *DNI del propietario*, *marca*, *modelo* y muchos otros que complementen la información de cada coche.

Los atributos se representan como círculos que descienden de una entidad, y no es necesario representarlos todos, sino los más significativos, como a continuación.



En un modelo relacional (ya implementado en una base de datos) una ejemplo de tabla dentro de una *BBDD* podría ser el siguiente.

|  |  |  |
| --- | --- | --- |
| Número de chasis | Matrícula | DNI del propietario |
| 5tfem5f10ax007210 | 4817 BFK | 45338600L |
| 6hsen2j98as001982 | 8810 CLM | 02405068K |
| 5rgsb7a19js001982 | 0019 GGL | 40588860J |

Este ejemplo es con tres atributos, pero un coche podría tener cientos (si fuese necesario) y seguirían la misma estructura de columnas, tras implementarlo en una *BBDD*.

La mayoría de los atributos tienen un **valor simple**; estos atributos reciben el nombre de sencillos. Por ejemplo la fecha de nacimiento es el valor sencillo para una persona. En algunas ocasiones un atributo puede tener un grupo de valores para el mismo registro -por ejemplo, un atributo de Color para un automóvil, o un atributo Grado para una persona. Los automóviles con un color tienen un valor sencillo, mientras un automóvil con dos tonos tiene dos valores para Color. De forma similar una persona puede no tener un grado, otra persona persona puede tener un grado y un tercero puede tener dos o más grados; eso quiere decir que, diferentes personas pueden tener diferente número de valores para el atributo Grado; a este tipo de atributo se le llama **multivaluado o de valor múltiple**.

Un atributo multivaluado debe tener un acotamiento mínimo y uno máximo para limitar el número de valores permitidos para cada registro. Por ejemplo, el atributo Color de un coche puede estar entre uno y tres valores, si asumimos que un automóvil puede tener cuando mucho tres colores.

## Relación

Es un vínculo que nos permite definir una dependencia entre varias entidades, es decir, nos permite exigir que varias entidades compartan ciertos atributos de forma indispensable.

Por ejemplo, los empleados del taller (de la entidad "Empleados") tienen un cargo (según la entidad "Cargo del empleado"). Es decir, un atributo de la entidad "*Empleados*" especificará que cargo tiene en el taller, y tiene que ser idéntico al que ya existe en la entidad "*Cargo del empleado*".

Las relaciones se muestran en los diagramas como rombos, que se unen a las entidades mediante líneas.

elación

Yo, bajo mi punto de vista, entiendo mejor esto en una tabla (de una implementación en una *BBDD*), por lo que voy a poner el ejemplo de como se representaría (resaltada la relación, que posteriormente veremos como se haría).

Empleados

|  |  |  |
| --- | --- | --- |
| Nombre | DNI | Cargo |
| Carlos Sánchez | 45338600L | 001 |
| Pepe Sánchez | 02405068K | 002 |
| Juan Sánchez | 40588860J | 002 |

Cargo del empleado

|  |  |
| --- | --- |
| ID del cargo | Descripción |
| 001 | Jefe de taller |
| 002 | Mecánico |

***Relaciones de cardinalidad***

Podemos encontrar distintos tipos de relaciones según como participen en ellas las entidades. Es decir, en el caso anterior cada empleado puede tener un cargo, pero un mismo cargo lo pueden compartir varios empleados.

Esto complementa a las representaciones de las relaciones, mediante un intervalo en cada extremo de la relación que especifica cuantos *objetos* o *cosas*(de cada entidad) pueden intervenir en esa relación.

Uno a uno: Una entidad se relaciona únicamente con otra y viceversa. Por ejemplo, si tuviésemos una entidad con distintos chasis y otra con matrículas deberíamos de determinar que cada chasis solo puede tener una matrícula (y cada matrícula un chasis, ni más en ningún caso).

elación uno a uno

Uno a varios o varios a uno: determina que un registro de una entidad puede estar relacionado con varios de otra entidad, pero en esta entidad existir solo una vez. Como ha sido en el caso anterior del trabajador del taller.

elación uno a varios

Varios a varios: determina que una entidad puede relacionarse con otra con ninguno o varios registros y viceversa. Por ejemplo, en el taller un coche puede ser reparado por varios mecánicos distintos y esos mecánicos pueden reparar varios coches distintos.

elación varios a varios

Los indicadores numéricos indican el primero el número mínimo de registros en una relación y posteriormente el máximo (si no hay límite se representa con una "*n*").

## Claves

Es el atributo de una entidad, al que le aplicamos una restricción que lo distingue de los demás registros (no permitiendo que el atributo específico se repita en la entidad)

o le aplica un vínculo (exactamente como comentábamos en las relaciones). Estos son los distintos tipos:

***Superclave***: aplica una clave o restricción a varios atributos de la entidad, para así asegurarse que en su conjunto no se repitan varias veces y así no poder entrar en dudas al querer identificar un registro.

***Clave primaria***: identifica inequívocamente un solo atributo no permitiendo que se repita en la misma entidad. Como sería la matrícula o el número de chasis de un coche (no puede existir dos veces el mismo).

***Clave externa*** *o* ***clave foránea***: este campo tiene que estar estrictamente relacionado con la clave primaria de otra entidad, para así exigir que exista previamente ese clave. Anteriormente hemos hablado de ello cuando comentábamos que un empleado indispensablemente tiene que tener un cargo (que lo hemos representado numéricamente), por lo cual si intentásemos darle un cargo inexistente el gestor de bases de datos nos devolvería un error.

### Integridad referencial.

Se denomina integridad referencial al tipo de interrelación que se produce entre tablas mediante un campo clave que deberá contener la cadena alfanumérica exacta al identificador de la tabla auxiliar para poder realizar la relación entre los registros. En caso contrario no se produce la relación. Además, se trata de un mecanismo que evita duplicidades e incorrecciones ya que la propiedad de integridad referencial conmina a que los datos de un usuario además de su identificador ID sean distintos al de los demás. Dicho de otra forma, no pueden existir dos registros iguales con los mismos datos.

### Entidades fuertes.

Lo constituyen las tablas principales de la base de datos que contienen los registros principales del sistema de información y que requieren de entidades o tablas auxiliares para completar su descripción o información. Por ejemplo la tabla usuario es una entidad fuerte en relación a la tabla tipos de usuarios, que es una entidad débil dada su condición auxiliar para clasificar a los usuarios registrados en la biblioteca.

### Entidades débiles.

Son entidades débiles a las tablas auxiliares de una tabla principal a la que completan o complementan con la información de sus registros relacionados. Por ejemplo también

son consideradas entidades débiles las tablas intermedias que sirven para compartir información de varias tablas principales.