

## Capítulo 3

# Creación de la base de datos

Errors using inadequate data are  
much less than those using no data  
at all.

---

Charles Babbage

El proceso de diseño de la base de datos ha sido descompuesto en tres diferentes etapas: el diseño conceptual, el diseño lógico y el diseño físico. La primera fase consiste en la producción de un esquema conceptual, que es independiente de todas las consideraciones físicas. Este modelo se refina después en un esquema lógico eliminando las construcciones que no se pueden representar en el modelo de base de datos escogido (relacional, orientado a objetos, etc.). En la tercera fase, el esquema lógico se traduce en un esquema físico para el SGBD<sup>1</sup> escogido.

### 3.1. Diseño conceptual

El primer paso en el diseño de la base de datos es la producción del esquema conceptual. El objetivo del mismo es describir el contenido de la información a almacenar en la base de datos y no las estructuras de almacenamiento que se necesitarán para manejar esta información. Esta etapa es independiente del SGBD concreto que se vaya a utilizar y de cualquier otra consideración física.

Para la realización del diseño conceptual, la primera cuestión a resolver es determinar de manera clara cuales son las prestaciones requeridas por el sistema,

---

<sup>1</sup>Un Sistema Gestor de Bases de Datos (SGBD) o DBMS (DataBase Management System) es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

es decir, cual es la información con la que trabajaremos y que datos necesitaremos almacenar en nuestra base de datos. En este aspecto, en nuestro sistema, es posible hacer una clara distinción entre dos bloques de información a tratar:

- **Gestión del tiempo.** Toda la información relativa a los turnos de acceso a la aplicación, la reserva de los mismos, las fechas a partir de las cuales se permite la reserva de turnos y el acceso a la aplicación y el registro de acceso a las redes.
- **Aplicación.** Incluirá todas las configuraciones creadas y almacenadas por los usuarios durante el transcurso de las prácticas.

Partiendo de las necesidades que deberá satisfacer la base de datos, diseñamos el esquema conceptual mediante el modelo Entidad-Relación (ER), una técnica de diseño de bases de datos gráfica que incorpora información relativa a los datos y la relación existente entre ellos.

Para la elaboración de este modelo debemos desglosar la información de la que partimos en bloques e identificar las distintas entidades y sus atributos así como las relaciones entre ellas.

Las entidades que tenemos son básicamente éstas:

- **Usuario.** Nos interesa saber su nombre y contraseña de acceso y si tiene permisos de administración.
- **Red.** Identificada por su id correspondiente.
- **Turno.** Definido por el día de la semana y las horas de inicio y fin del mismo.
- **Periodo de acceso.** Fijado para la reserva de turnos y el acceso a la aplicación y que abarca el tiempo comprendido entre unas determinadas fechas de inicio y fin.
- **Configuración.** Posee un determinado nombre y fecha de última modificación abarcara la configuración de los distintos elementos de la red:
  - **PC.** Almacenaremos diferentes parámetros de configuración de red como son la IP, máscara de red, puerta de enlace y los servidores DNS.
  - **Conmutador.** Necesitaremos conocer la ruta del archivo de configuración.

Y las relaciones que hay entre estas entidades son:

- Un usuario deberá ser capaz de realizar una *reserva*, y únicamente una, de una determinada red a lo largo de un turno en concreto.
- Una red podrá o no estar *siendo utilizada* por un usuario, y solo uno, en un cierto instante.
- Una red podrá ser *configurada*.
- Un usuario *creará* una o más configuraciones, las cuales *contendrán* las configuraciones de cada uno de los PCs y conmutadores que conforman la red con la que se encuentren realizando las prácticas.

Una vez identificados todos los conceptos, es posible dibujar el diagrama entidad-relación correspondiente. Podemos ver el resultado en la figura 3.1.

## 3.2. Diseño lógico

En esta etapa, se transforma el esquema conceptual en un esquema lógico que utilizará las estructuras de datos del modelo de base de datos en el que se basa el SGBD que se vaya a utilizar, como puede ser el modelo relacional, el modelo de red o el modelo jerárquico. En nuestro caso el SGBD escogido es MySQL, un manejador de base de datos relacionales. Tendremos pues que seguir una serie de pasos para la transformación de nuestro modelo ER en un modelo relacional.

El elemento fundamental según el modelo relacional es lo que se conoce como relación, aunque más habitualmente se le llama tabla. Se trata de una estructura formada por filas y columnas que almacena los datos referentes a una determinada entidad. Como el modelo relacional no distingue entre entidades e interrelaciones, ambos conceptos deben representarse mediante tablas. Debemos por lo tanto analizar cada una de las entidades y relaciones dentro del modelo ER y realizar la conversión a tablas.

Todas las entidades regulares se transforman en tablas en el modelo relacional, manteniendo el número y tipo de los atributos, así como las claves primarias. Las entidades débiles también se convierten en tablas en el modelo relacional, manteniendo el número y tipo de los atributos, formando su clave primaria mediante la composición de la actual con la clave primaria de la entidad regular de la cual depende. Tendremos así las siguientes tablas: **usuario**, **turno**, **red**, **acceso**, **configuracion**, **pc** y **conmutador**.

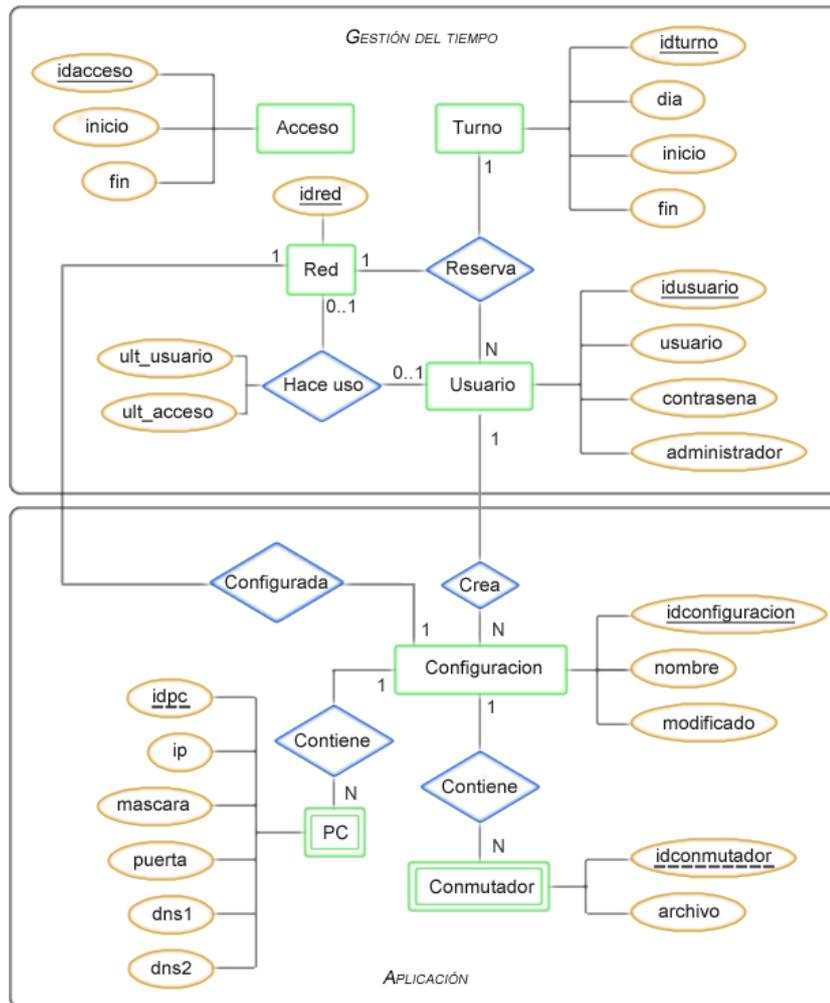


Figura 3.1: Diagrama Entidad-Relación

En cuanto a las relaciones presentes en nuestro diseño, se presenta la necesidad de crear una nueva tabla, **reserva**, que tendrá los atributos correspondientes a la relación y cuya clave estará formada por la composición de los identificadores de las entidades que participan en la misma. Por otro lado, los atributos de la relación 'Hace uso' pasarán a formar parte de la tabla red.

Una vez realizada esta conversión, el siguiente paso consiste en la normalización del resultado obtenido. A través de este proceso logramos que los datos obtenidos tengan una estructura óptima para su implementación, gestión y explotación. Una de las principales ventajas de llevar a cabo la normalización es que la información no estará duplicada innecesariamente dentro de las estructuras: habrá mínima re-

dundancia. Este proceso se basa en ir comprobando que cada tabla cumple una serie de reglas centradas en la clave primaria y las dependencias funcionales. Cada regla que se cumpla aumenta el grado de normalización. En caso contrario, la relación se debe descomponer en varias relaciones que sí la cumplan. El modelo relacional sólo requiere que las tablas se hallen en lo que se conoce como primera forma normal (1NF). Las restantes formas normales son opcionales, sin embargo, es recomendable llegar al menos a la tercera forma normal (3NF). Nos aseguraremos pues que todas las tablas de nuestro modelo relacional estén estructuradas de acuerdo a la 3NF.

El resultado final obtenido se encuentra representado en la figura 3.2.

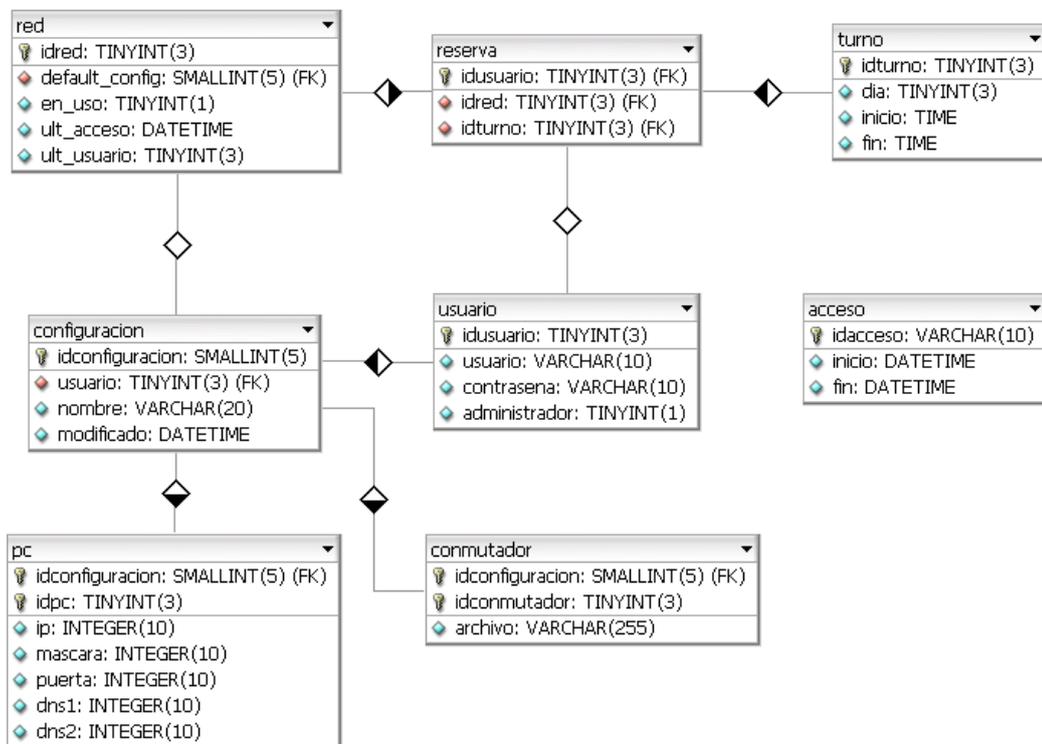


Figura 3.2: Modelo relacional

### 3.3. Diseño físico

El diseño físico de la base de datos especifica la configuración física de la base de datos en los medios de almacenaje. Describe las relaciones base, la organización de los archivos y de los índices utilizados para conseguir un acceso eficiente a los

datos y cualesquiera restricciones de integridad asociadas y medidas de seguridad implementadas. El esquema físico se expresa haciendo uso del lenguaje de definición de datos del SGBD objetivo, en nuestro caso es SQL (Structured Query Language).

El código resultante para la definición de nuestra base de datos puede verse en el apéndice B.1.

### 3.4. El acceso a la base de datos

En esta etapa se diseñan los programas de aplicación que usarán y procesarán la base de datos. Esta etapa y el diseño de la base de datos, son paralelas. En la mayor parte de los casos no se puede finalizar el diseño de las aplicaciones hasta que se ha terminado con el diseño de la base de datos. Este proceso está compuesto por dos actividades: el diseño de las transacciones y el diseño de la interfaz de usuario.

El objetivo de analizar las distintas transacciones que se realizarán con la base de datos es el de optimizar el tiempo de respuesta. Una vez analizadas las posibles consultas se lleva a cabo la creación de los índices. Los índices son usados para encontrar rápidamente los registros que tengan un determinado valor en alguna de sus columnas. Sin un índice, MySQL tiene que iniciar con el primer registro y leer a través de toda la tabla para encontrar los registros relevantes. Aún en tablas pequeñas, de unos 1000 registros, es por lo menos 100 veces más rápido leer los datos usando un índice, que haciendo una lectura secuencial.

En lo referente a la interfaz de usuario, en nuestro caso, ésta se encuentra incluida dentro de la página web creada para este proyecto. Desde ella se controlan las vistas y la modificación de los datos almacenados en nuestra base de datos. Haciendo uso de los lenguajes HTML y JavaScript en el lado del cliente se ofrece al usuario una interfaz simple a través de diversos menús y formularios con el cual poder consultar la información almacenada en el servidor y cambiar aquella sobre la que se tenga permisos de modificación.

Las consultas y transacciones permitidas están limitadas en función de si el usuario está definido o no como administrador del sistema. De esta manera la interacción de un alumno (al que no se le han concedido los derechos de administración) con la base de datos está limitada a la consulta de las fechas de acceso y horario de los turnos, así como la realización de la reserva de los mismos.

Por otro lado el administrador del sistema tendrá un control total a la hora de establecer, editar o borrar cuentas de usuario, turnos de prácticas, reservas o fechas de acceso.

Para la comunicación con la base de datos se ha creado una clase PHP a la que hemos denominado `Database`. Esta clase busca ampliar las capacidades de la clase `mysqli` aplicándole el patrón singleton además de dotarla de la capacidad de lanzar excepciones propias. La definición de esta clase se encuentra dentro del archivo `func_comunes.php` cuyo contenido completo se puede observar en el apéndice B.2.

