

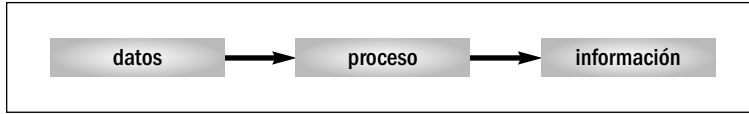
# Conceptos generales

En este capítulo veremos los primeros elementos para aproximarnos al uso de las bases de datos, explicando los diferentes tipos de guardado de los datos (Archivos y Bases de datos). También se indicará cómo se organizan los archivos (secuenciales, indexados y relativos) y se presentarán algunos conceptos, tales como registro, campo y tipo de datos.

<b>Un poco de historia</b>	<b>16</b>
Organización de los archivos	17
Inconvenientes en el uso de archivos	18
<b>¿Qué es una base de datos?</b>	<b>20</b>
Ventajas de utilizar bases de datos	23
<b>Resumen</b>	<b>25</b>
<b>Actividades</b>	<b>26</b>

## UN POCO DE HISTORIA

Lo primero y fundamental para ingresar al apasionante mundo de la programación con bases de datos, es lo siguiente:



**Figura 1.** Etapas del diseño de un programa.

En la figura vemos las tres etapas bien definidas de todo proyecto. La primera es **Datos**, que son los elementos básicos o fundamentales con los que cuento en el momento de inicio de la tarea a realizar, por si solos, ellos no poseen ningún valor agregado. La segunda es **Proceso**, que es un conjunto de acciones o tareas que se realizan en un orden específico con un principio y un fin bien definidos. Por último, la tercera es **Información** y es el resultado que se obtiene como consecuencia de procesar los datos, y el fin primordial que se busca en todo sistema o programa informático, obtener información certera y precisa en el menor tiempo posible.

Si bien los datos y la información son únicos e inconfundibles, pueden existir múltiples procesos para llegar a la información deseada, tomando como punto de partida los datos que poseemos, lo que se busca es que sea el más eficiente de todos los factibles.

Ahora bien queda claro que durante toda la secuencia anteriormente definida hay datos que en determinado momento están en movimiento y en otro determinado momento están en reposo; los datos que están en movimiento es tema para otro libro. Lo que nos importa a nosotros realmente son los datos que están en reposo.

Como es lógico de imaginar, estos datos que se encuentran en reposo hay que guardarlos en algún lugar. Para ello, dentro del mundo de sistemas, los lugares “lógicos” factibles para dicha tarea son: Archivos, Bases de datos, etc..... Cabe aclarar, como detalle significativo, que según cómo se organice lógicamente la información se facilitará o dificultará, esperemos que esto último no nos suceda, el acceso a ella.

### III PERSISTENCIA DE DATOS

Comenzaremos a definir algunos términos usados en el mundo de la programación, como complemento de las nociones teóricas elaboradas. Normalmente en la jerga informática, para referirse a los datos en reposo se suele decir “persistencia de datos”.

Antes de la existencia de las bases de datos, la información lógica (a partir de ahora siempre que digamos información vamos a asumir que es información lógica) se guardaba en archivos. Un archivo estaba compuesto por **registros** y cada registro estaba compuesto por un **conjunto de campos**.

## Organización de los archivos

Es importante que entienda que la organización de los archivos se refiere en este caso a cómo se almacenan y acceden físicamente a los registros del archivo. Existen tres formas de organizar el acceso a la información de los archivos. Según el tipo de organización se definen como:

- Archivos secuenciales
- Archivos indexados
- Archivos relativos o directos

Veamos en qué se diferencian.

### Archivos Secuenciales

En dichos archivos tanto para leer un registro como para escribir un registro, se hace de a uno por vez y un registro detrás de otro. Ej., supóngase que usted tuviera 30 contactos en su agenda guardada en un archivo secuencial y que nuestro amigo Juan Perez estuviera guardado en la posición de registro número 20. Para ver cuál era la dirección de mail de su amigo, usted debería abrir el archivo secuencial para lectura y leer los 19 registros anteriores hasta que al leer el registro 20 accediera al mail de aquel. Imagínese si su lista de contactos fuera mucho más grande y Juan Pérez se encontrara guardado en un registro muy posterior, creo que no hace falta ninguna otra explicación.

Es importante que se entienda que en este tipo de organización, salvo el primer registro, todos los demás registros tienen un registro que los precede y salvo el último registro, todos los demás tienen un registro que los sucede. El orden físico de los registros se corresponde con el orden lógico.



## ALMACENAMIENTO FÍSICO Y LÓGICO

El almacenamiento "lógico" se refiere a la estructura lógica de cómo se organiza la información. Ej.: Uno podría tener los datos de sus contactos dentro de a) Una agenda de contactos como la del Outlook, b) Una tabla dentro de una base de datos o c) Una planilla de Excel. La información lógica podría estar almacenadas físicamente en uno o más archivos de un disco rígido,

## Archivos Indexados

Para evitar el problema que se generaba con los archivos secuenciales, se creó una estructura que anexa al archivo de datos un índice según un orden y criterio predefinido, con lo cual se agiliza el acceso a los datos organizados por dicho índice.

Un ejemplo de esto son los índices de los libros. Si estos no existieran usted estaría obligado a leer el libro página por página, pero con el índice Ud. puede acceder al tema que le interesa directamente sin tener que leer todas las páginas anteriores. El orden físico de los registros se corresponde con el orden lógico.

## Archivos Relativos

En este tipo de organización el orden físico de los registros no necesariamente se corresponde con el orden lógico de ellos. El contenido de los datos del registro determina su posición física, de manera que, conociendo el registro se puede hacer referencia a dicha posición. Se puede acceder a cada uno de los registros sin necesidad de leer los que le preceden. La relación entre el registro y su posición física se obtiene mediante una fórmula.

## Inconvenientes en el uso de archivos

En general los inconvenientes en el uso de archivos para la persistencia de los datos pueden ser los siguientes:

- Redundancia de datos.
- Problema de consistencia e integridad de los datos.
- Dificultad para modificar su estructura lógica.

## Redundancia de datos

Por lo general, para las aplicaciones que persisten sus datos en archivos, cada aplicación está asociada a un archivo. Por ejemplo, si los sistemas de una empresa están soportados sobre archivos es muy probable que el área de recursos humanos posea

---

## REGISTRO Y CAMPOS

Un registro es la mínima unidad disponible a la cual se puede acceder en un archivo. Tomemos como ejemplo el caso de la agenda de contacto y supongamos que tomamos como datos relevantes el Nombre, DNI, la dirección, el teléfono, el mail y la fecha de nacimiento. A estos datos los llamaremos campos. Es decir que nuestro registro está compuesto por 6 (seis) campos.

un archivo con los datos más relevantes de todos sus empleados, así el registro contendría los siguientes campos: Número de Legajo, Nombres, Apellido, DNI, dirección, teléfono, mail, fecha de nacimiento, fecha de ingreso (por citar los más relevantes, podría tener más atributos o campos).

A su vez, el departamento de ventas también podría tener un archivo propio en el cual se persistieran los datos de la venta realizadas por cada uno de los empleados, con el siguiente diseño de registro: Legajo Empleado, Nombre del Empleado, Apellido del Empleado, Número de Venta, Importe Venta.

Ahora bien, veamos un par de detalles. Está claro que cada departamento (ventas, y RRHH) deberán cargar los números de legajos, nombres y apellido de los empleados una vez cada uno con lo cual dichos datos estarán cargados dos veces (duplicados) en dos archivos diferentes (**esto es redundancia de datos**). Supóngase que la empresa posee empleadas solteras y que alguna de ellas se casa y desea que la registren con su apellido de casada. Por lo que citamos anteriormente, cada departamento estaría obligado a actualizar el apellido de esta.

### **Consistencia e integridad de datos**

En el caso anterior de la empleada que se casó y actualizó su apellido con su apellido de casada, si el departamento de RRHH realiza la actualización del dato y el de Ventas no, la información sobre la empleada en cuestión quedaría inconsistente y como consecuencia se perdería la integridad de los datos, ya que para un mismo legajo tendríamos dos personas diferentes.

Además, sucede algo mucho más grave que muestra la pérdida de integridad: cómo Recursos Humanos realizó la actualización del apellido y Ventas no, las ventas registradas en el archivo de ventas no pertenecen a ninguna empleada actual de la empresa. Al no coincidir los apellidos, no podría liquidarse la comisión para dicha empleada. La integridad de los datos también se puede perder por fallas físicas o lógicas al realizar una operación de entrada /salida (lectura /escritura) sobre el archivo con el que se está operando.

## **III TABLAS**

La tabla es un objeto de una base de datos que almacena datos en una colección de filas y columnas. La tabla se define por medio de las columnas que contiene. Cada columna representa a un campo diferente y cada fila a un registro diferente. El concepto de campo y registro es el mismo que para los archivos. Una base de datos puede contener o albergar una o más tablas.

## Modificar la estructura lógica

Ahora bien supongamos que el departamento de ventas quisiera registrar, además, la fecha en que se realizó la venta pues surgió la necesidad de relevar la cantidad de ventas realizadas por cada empleado por mes. Esto implicaría modificar la estructura del registro del archivo de ventas, que quedaría como la siguiente: Legajo Empleado, Nombre del Empleado, Apellido del Empleado, Número de Venta, Importe Venta, Fecha de Venta, y lo que es peor, habría que modificar todo el código del programa para que aceptara este nuevo diseño de registro.

## ¿QUÉ ES UNA BASE DE DATOS?

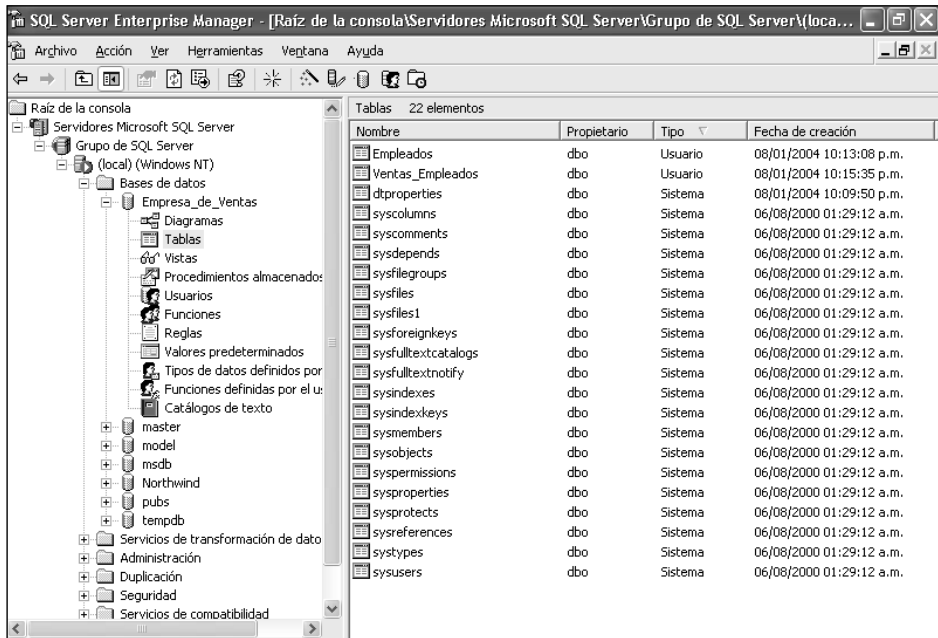
Antes hablamos que en este libro nos íbamos a ocupar de la persistencia de datos (datos en reposo). Pues bien: una base de datos es el lugar donde se almacenan los datos en reposo. Si uno quisiera hilar más fino y ser un poco más exquisito, podría decir que: “Una base de datos es el lugar donde se guardan los datos en reposo y al cual acceden las diferentes aplicaciones (sistemas o programas) de una organización (tomemos como organización una definición macro en la cual quedan incluidas las empresas, ONG, organizaciones sin fines de lucro, administración pública, etc...) dada”.

Retomemos el ejemplo anterior de nuestra empresa que tenía un departamento de RRHH y otro de Ventas. En este caso toda la información estaría guardada en una única base de datos que llamaríamos **Empresa\_de\_Ventas**, la cual contendría dos tablas (**Figura 2**); una tabla para registrar los empleados de nombre **Empleados** con el siguiente diseño de registro: **Número\_Legajo, Nombres, Apellido, DNI, dirección, teléfono, mail, fecha\_nacimiento, fecha\_ingreso** (**Figura 3**) y otra tabla con el nombre **Ventas\_Empleados** para registrar las ventas de los empleados con el siguiente diseño de registro: **Legajo\_Emp, Número\_Venta, Importe\_Venta, Fecha\_Venta** (**Figura 4**).

Observe que en la tabla **Ventas\_Emp** ya no aparecen los campos **Nombre\_Empleado** y **Apellido\_Empleado** como sucedía con los archivos.

## SQL SERVER

SQL Server es un motor de base de datos que comercializa Microsoft. Este posee varias aplicaciones para la administración de las bases de datos. En este libro haremos incapié en dos aplicaciones fundamentales: el **Enterprise Manager** (Administrador de tareas) y el **Query Analyzer** (Analizador de consultas) y lo usaremos para realizar demostraciones prácticas.



**Figura 2.** Vista de la Base de datos **Empresa\_de\_Ventas** en SQL Server con las Tablas de **Empleados** y **Ventas\_Empleados**.

La **Figura 2** muestra la pantalla del Enterprise Manager (Administrador de Tareas) de SQL Server. Observe que en la parte izquierda de la pantalla aparece una solapa de Bases de Datos y dentro de ella hemos creado una base de datos de nombre **Empresa\_de\_Ventas** y además ella tiene seleccionada la solapa **Tablas**. Y si nos fijamos su parte superior derecha, encontraremos las tablas **Empleados** y **Ventas\_Empleados**. La idea de esta figura es que se vaya familiarizando poco a poco con el entorno de trabajo que propone esta aplicación de Microsoft.

Observe que en la **Figura 3** se muestra la **Tabla Empleados** desde la ventana de propiedades, para activarla simplemente hay que presionar el botón derecho del mouse con la tabla seleccionada y elegir la opción **Propiedades**. Fíjese que no se llegan a ver todos sus campos y habría que desplazarse con las barras laterales.



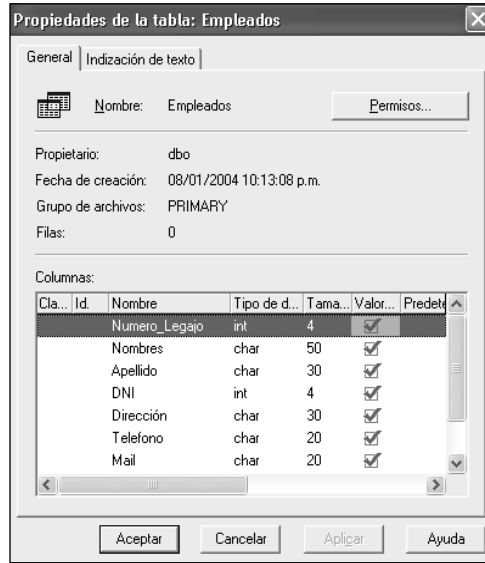
## PROPIEDADES DE LA TABLA

Si hace doble clic sobre la tabla seleccionada se desplegará su ventana de **Propiedades**. Desde ahí, podrá acceder directamente a todas las opciones de la tabla.



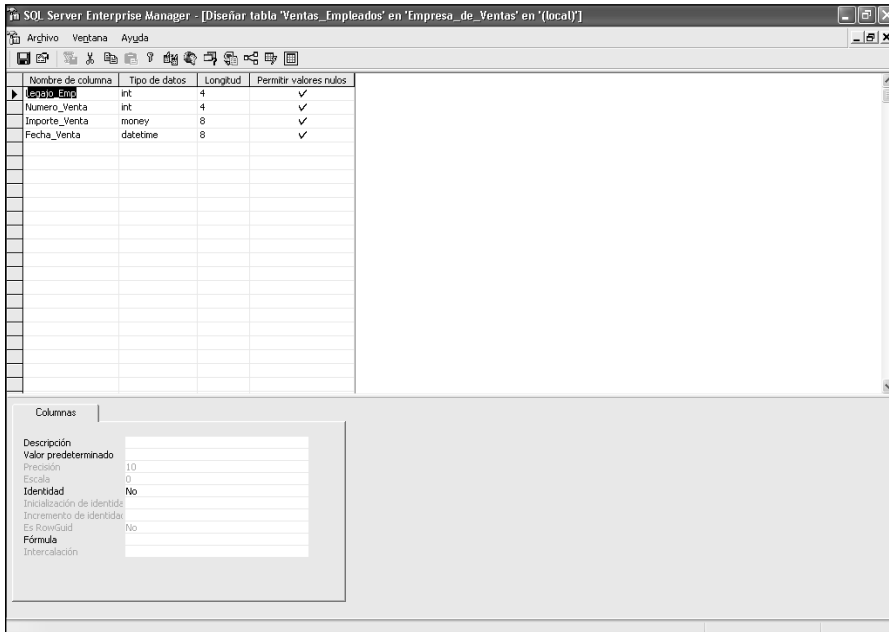
## TRANSACCIONES

El concepto de transacción es definir una unidad básica de trabajo. Esta unidad básica de trabajo está compuesta por un conjunto de tareas o instrucciones a realizar.



**Figura 3. Ventana de propiedades de la tabla Empleados.**

Observe que la ventana de la **Figura 4** muestra la vista de diseño de la tabla denominada **Ventas\_Empleados** donde, del lado izquierdo de la pantalla aparece el nombre del campo con el cual se lo identificará de aquí en adelante y del lado derecho, el tipo de dato que guardará ese campo (**Char[n]**, {n cantidad de posiciones}, **int** {número entero}, **money** {moneda}, **datetime** {día y hora}, etc.).



**Figura 4. Vista de diseño de la tabla Ventas\_Empleados.**

## Ventajas de utilizar bases de datos

Vamos a describir brevemente algunas de las principales ventajas de utilizar base de datos para la persistencia de los datos:

- Normalizar los datos de la base de datos.
  - Evitar la redundancia de datos.
  - Evitar la inconsistencia de datos.
  - Garantizar la integridad de los datos.
  - Garantizar la seguridad de los datos.
  - Compartir los datos.
  - Facilidad de modificar los datos.
- 
- **Normalizar los datos:** Es una metodología (ya lo veremos con profundidad en próximos capítulos) que permite minimizar entre otras cosas la redundancia de datos, y agiliza y garantiza la actualización de estos.
  - **Evitar redundancia de datos:** Como política de las bases de datos, estos se tratan de guardar en un único lugar y cuando existe la necesidad de acceder a ellos, se hace por medio de relaciones entre los mismos. A veces se sacrifica el tema de no ser redundante en pos de la performance.
  - **Evitar inconsistencias de datos:** Dado que las bases de datos utilizan transacciones se puede garantizar prácticamente la inexistencia de inconsistencias de datos.
  - **Garantizar la integridad de los datos:** Como se evita la redundancia de datos y la inconsistencia y se manejan transacciones, la integridad está garantizada. El concepto básico de integridad es que la información obtenida de la base de datos es correcta en todo momento.
  - **Garantizar la seguridad de los datos:** Debido a que los accesos a la base de datos tanto para usuarios como para aplicaciones están dados por medio de permisos, si estos últimos están bien definidos nadie podrá ingresar a ningún lugar que no le esté permitido ni acceder a un conjunto de datos que no le esté permitido. Existen además otros tipos de objetos de la base de datos, como las vistas (que las veremos en capítulos siguientes) que también colaboran en el tema de seguridad y restricciones de acceso a los datos.
  - **Compartir los datos:** Dado que todos los datos de una base de datos se almacenan en un mismo lugar lógico (la base de datos) estos se pueden compartir sin problema entre distintos usuarios y aplicaciones.
  - **Facilidad de modificar los datos:** Como se evita la redundancia de datos y estos se guardan en un único lugar es más fácil poder realizar modificaciones sobre ellos.

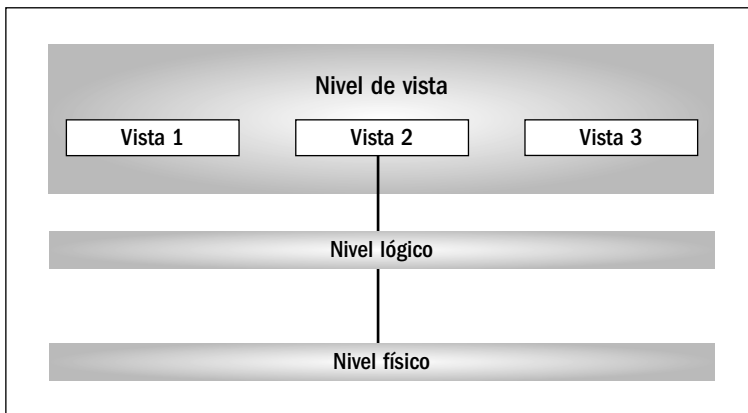
## Visión de los datos

Un SGBD (sistema gestor de base de datos) es una colección de archivos interrelacionados o un conjunto de programas que permiten a los usuarios acceder y

modificar estos archivos. El propósito principal de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los datos. Es decir, el sistema esconde ciertos detalles desde cómo se almacenan y mantienen los datos.

Para que el sistema sea útil, debe recuperar los datos eficientemente. Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadoras, los desarrolladores esconden la complejidad a los usuarios a través de varios niveles de abstracción para simplificar la interacción de los usuarios con el sistema:

- **Nivel físico:** el nivel más bajo de abstracción describe cómo se almacenan realmente los datos. Se presentan en detalle las estructuras de datos complejas de bajo nivel.
- **Nivel lógico:** el siguiente nivel más alto de abstracción describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos. Los administradores de bases de datos, que deben decidir la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.
- **Nivel de vistas:** el nivel más alto de abstracción describe solo parte de la base de datos completa. El sistema puede proporcionar muchas vistas para la misma base.



**Figura 5.** Niveles de abstracción que posee la base de datos y la interrelación entre ellos.

En el **nivel físico**, un registro se puede describir como un bloque de posiciones almacenadas consecutivamente (por ejemplo, palabras o bytes). El compilador del lenguaje esconde este nivel de detalle para los programadores.

En el **nivel lógico** cada registro de este tipo se describe mediante una definición de tipo, y se define la relación entre estos tipos de registros. Los programadores, cuando usan un lenguaje de programación, trabajan en este nivel de abstracción. En forma análoga, los administradores habitualmente trabajan en este nivel de abstracción. Finalmente, en el **nivel de vistas**, los usuarios de computadoras ven un conjunto de programas de aplicación que esconden los detalles de los tipos de datos. Análogamente, en el nivel de vistas se definen varias vistas de una base de datos, y los

usuarios de la base de datos ven esas vistas. Además de esconder detalles del nivel lógico de la base de datos, las vistas también proporcionan un mecanismo de seguridad para evitar que los usuarios accedan a partes de la base de datos.

## Ejemplares y Esquemas

La colección de información almacenada en la base de datos en un momento particular se llama un ejemplar de la base de datos. El diseño completo de la base de datos se llama **esquema de la base de datos**. Los esquemas son raramente modificados, si es que lo son alguna vez.

Un esquema de base de datos corresponde a una definición de tipo en un lenguaje de programación. Una variable de un tipo dado tiene un valor particular en un instante de tiempo. Así, el valor de una variable en lenguajes de programación corresponde a un ejemplar de un esquema de bases de datos.

Los sistemas de bases de datos tienen varios esquemas divididos, de acuerdo con los niveles de abstracción que se han discutido. En el nivel más bajo, está el esquema físico; en el nivel intermedio está el esquema lógico, y en el nivel más alto está el subesquema. En general, los sistemas de bases de datos soportan un esquema físico, un esquema lógico y varios subesquemas.

## RESUMEN

En este capítulo vimos cómo se guardan los datos en reposo de los diferentes sistemas o programas. Explicamos los diferentes tipos de guardado de los datos (Archivos y Bases de datos). Dimos una pequeña explicación de las diferentes formas en que se organizan los archivos (secuenciales, indexados y relativos) y explicamos algunos conceptos básicos de archivos y bases de datos de forma elemental, como ser registro, campo y tipo de datos.

Además definimos de forma elemental conceptos como redundancia, integridad, inconsistencia de datos y transacciones.



### TEST DE AUTOEVALUACIÓN

- 1 ¿A qué tipo de datos nos abocaremos a lo largo de este libro? ¿Por qué?
- 2 ¿De qué forma podemos guardar los datos?
- 3 ¿Cuáles son los tipos de organización de archivos conoce?
- 4 ¿Cuál es la diferencia que existe entre un registro y un campo?
- 5 ¿Qué es una base de datos?
- 6 ¿Cómo se organizan los datos dentro de una base de datos?
- 7 ¿Con qué motor de base de datos vamos a trabajar?
- 8 ¿Cuáles son las ventajas de utilizar una base de datos?
- 9 ¿A qué nos referimos al hablar de la normalización de una base de datos?
- 10 ¿Cuáles son los niveles de abstracción de una base de datos?

