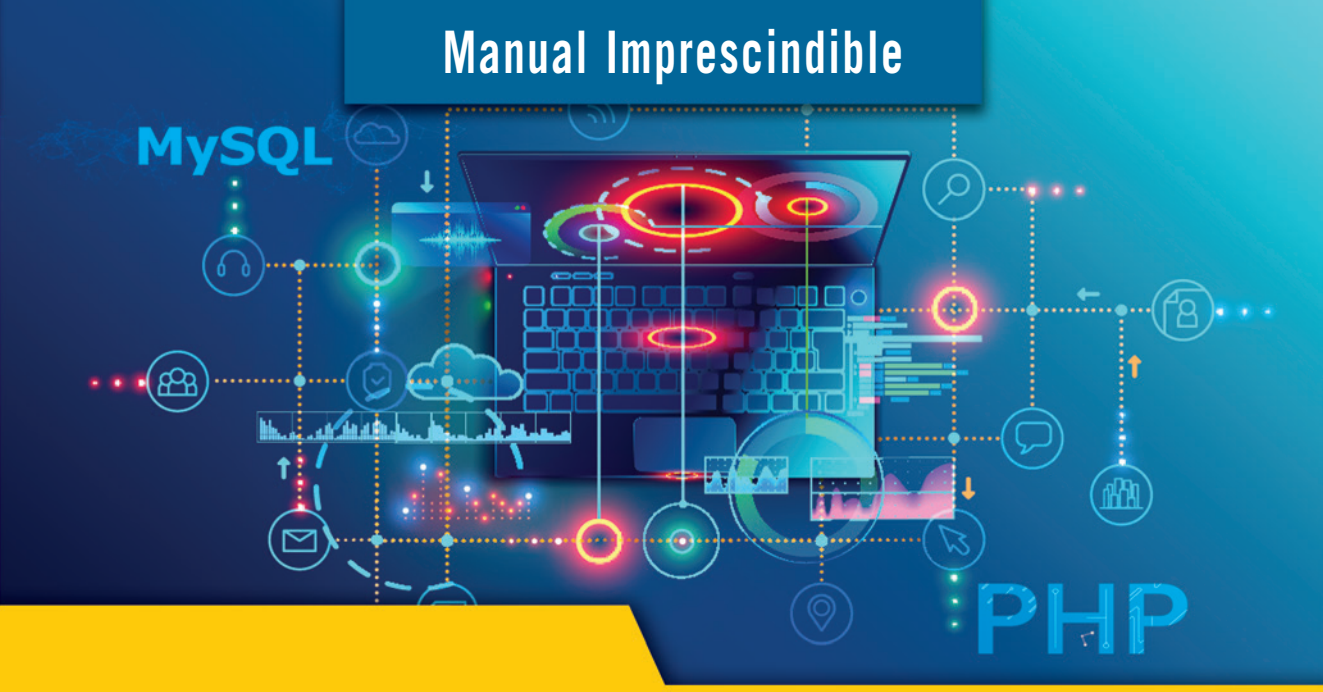
Manual Imprescindible



CURSO DE PHP 8 Y MySQL 8

Luis Miguel Cabezas Granado
Francisco José González Lozano

ANAYA MULTIMEDIA

ndice de contenidos

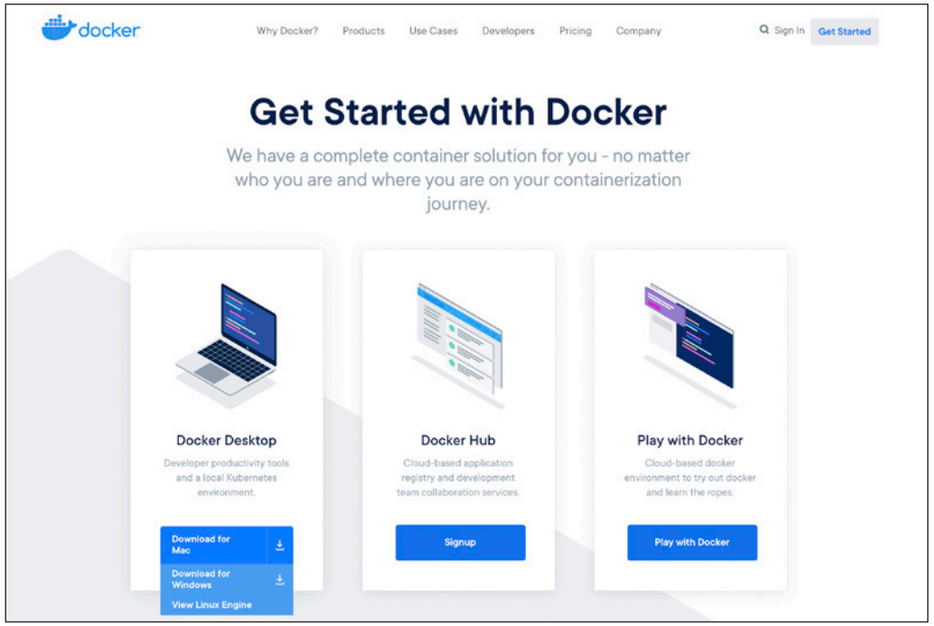


Figura 1.6. Descarga de Docker (fuente: `https://docker.com`).

Para comprobar que Docker está instalado, tendremos que utilizar un terminal, que no es más que una aplicación del sistema operativo que permite escribir comandos. Acostúmbrese a utilizar comandos desde el principio, va a ser muy necesario el día que quiera desplegar aplicaciones en la nube.

En Windows 10 se puede abrir el terminal desde la caja inferior de búsqueda escribiendo **terminal** o **cmd**. En macOS se puede buscar en Spotlight la palabra **terminal**.

Para comprobar que todo está correcto, debemos escribir el comando `docker version` y obtendremos algo parecido a la figura 1.7.

Evidentemente, no debe tener en cuenta las versiones que le aparezcan, Docker se actualiza muy a menudo y seguramente no sean iguales a las de la figura 1.6.

Conceptualmente, un **contenedor** es un entorno aislado donde se pueden ejecutar aplicaciones. Se les suele comparar con máquinas virtuales (VirtualBox, WMWare), pero las máquinas virtuales necesitan un sistema operativo completo para funcionar. Un contenedor tan solo necesita un pequeño núcleo (*kernel*) y una capa de aislamiento, el resto de recursos los obtiene del sistema operativo anfitrión; por lo tanto, los **contenedores** suelen ser muy livianos y ligeros.

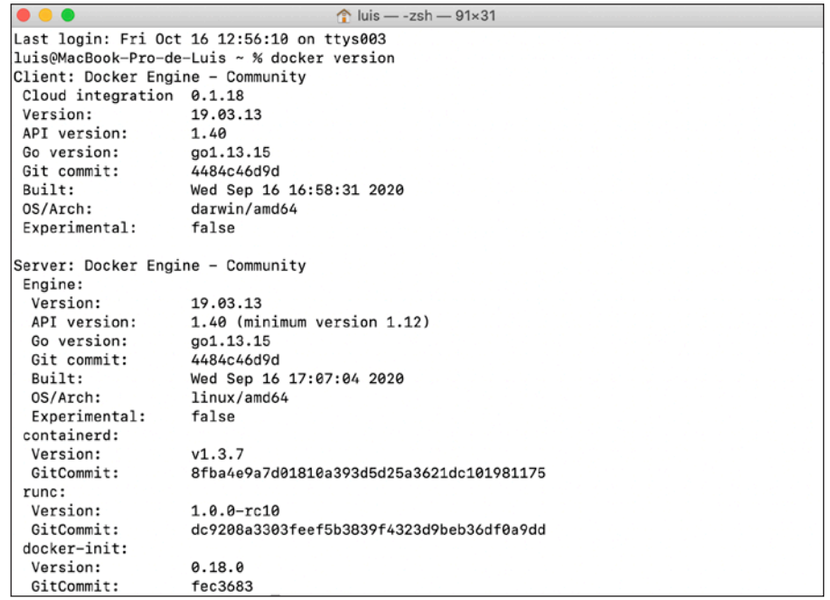


Figura 1.7. `docker version`.

Podemos pensar que un **contenedor** es una especie de caja donde guardar y ejecutar aplicaciones, pero esas aplicaciones debemos sacarlas antes de algún sitio. El otro concepto básico en Docker son las **imágenes**. Una imagen es un paquete de software que sirve como base para un contenedor. Hay imágenes de todo tipo: PHP, MySQL, Apache, WordPress, etc. Como ejemplo puede probar a instalar una imagen creada para este libro con el siguiente comando:

```
docker run -d hazardco/anaya-php8:v1.0.1
```

Docker intenta localizar esa imagen dentro de su disco duro. Si no la encuentra, la busca en una base de datos de imágenes llamada dockerhub (`https://hub.docker.com`) y la descarga. Cuando ya está la imagen en el disco, procede a ejecutarla en segundo plano gracias al parámetro `-d`. Puede ver el resultado en la figura 1.8.

Después de ejecutar la imagen de PHP 8, lo que obtenemos es un contenedor que está iniciado, pero que no es visible *a priori*. Además del contenedor que está ejecutándose, Docker guarda de forma local una copia de la imagen descargada. Se puede obtener un listado de imágenes con el comando:

```
docker image list
```

ADVERTENCIA:

No hay que confundir el operador de asignación (=) con el operador de comparación, doble igual (==), sobre todo en bucles de control, donde sería algo complicado encontrar el error.

3.4. Operadores aritméticos

Este tipo de operador forma parte de la aritmética básica. Le resultará familiar porque son símbolos muy utilizados en el aprendizaje de las matemáticas.

Tabla 3.1. Operadores aritméticos.

Ejemplo	Nombre	Resultado
`$a + $b`	Suma	Suma dos valores
`$a - $b`	Resta	Resta dos números
`$a * $b`	Multiplicación	Producto de variables
`$a / $b`	División	Cociente entre dos variables
`$a % $b`	Módulo	Resto de la división de `$a` entre `$b`

NOTA:

PHP ignora los espacios en blanco entre las variables y los operadores. Aunque `$a + $b` *es equivalente a* `$a+$b`*, es preferible utilizar la primera forma, porque es más legible.*

3.5. Operadores de comparación

En algunos ejemplos del capítulo anterior puede ver que se utiliza la estructura de control `if ... else`. Como verá más adelante, esta estructura compara dos valores y elige el camino a seguir dentro del código. El resultado siempre es `true` o `false`.

Tabla 3.2. Operadores de comparación.

Ejemplo	Nombre	Resultado
`$a == $b`	Igual	Verdadero si los dos son iguales
`$a === $b`	Identidad	Verdadero si son iguales y del mismo tipo
`$a != $b`	Distinto	Verdadero si las dos variables son distintas
`$a < $b`	Menor que	`true` si `$a` es menor que `$b`
`$a > $b`	Mayor que	`true` si `$a` es mayor que `$b`
`$a <= $b`	Menor o igual	`true` si `$a` es menor o igual a `$b`
`$a >= $b`	Mayor o igual	`true` si `$a` es mayor o igual a `$b`

El código siguiente es un típico ejemplo de utilización de operadores de comparación. En la figura 3.1, puede ver el resultado.

```
<?php
$a = 23; // Asignación de los valores
$b = 75;
if ( $a >= $b ) { // La condición no se cumple. El resultado es false
  echo "Esta parte no se ejecuta";
} else {
  echo "La comparación es FALSE porque $a es menor que $b";
}
```

Figura 3.1. Operador de comparación.

3.6. Operadores lógicos

Durante el desarrollo de su proyecto, puede encontrarse con situaciones en las que necesite hacer varias comparaciones seguidas para que se cumpla una determinada condición. PHP permite unir todas las comparaciones en una mediante el uso de operadores lógicos. Es posible encadenar varias comparaciones juntas mediante este tipo de operadores.

Tabla 3.3. Operadores lógicos.

Ejemplo	Nombre	Resultado
`expresión1 and expresión2`	Y	`true` si las dos expresiones son verdaderas
`expresión1 && expresión2`	Y	`true` si las dos expresiones son verdaderas
`expresión1 or expresión2`	O	`true` si una de las expresiones es verdadera

5 Funciones

En este capítulo aprenderás:

- Qué es y para qué sirve una función
- Funciones de fecha
- Definir funciones
- Ámbito de variables
- Cómo se pueden definir argumentos de función
- Qué son las funciones anónimas
- Recursividad

5.1. Funciones

Las funciones son grupos de instrucciones independientes que tienen un propósito determinado. De esta manera, tenemos funciones que calculan la raíz cuadrada de un número, nos dan un número aleatorio o permiten contar los caracteres de una palabra.

La sintaxis básica de una función es la que se describe a continuación:

```
function nombre_función(parámetro1, parámetro2, parámetro3...parámetro_n)
```

Las funciones pueden ser llamadas con varios parámetros o con ninguno, dependiendo de su definición. Cuando PHP encuentra en el código la llamada a una función, primero evalúa cada argumento y los utiliza como parámetro de entrada. Después, ejecuta la función y devuelve el valor solicitado o realiza alguna acción sin enviar ningún valor de salida.

El siguiente ejemplo contiene un conjunto de llamadas a funciones con 0, 1 o 2 parámetros de entrada:

```
<?php
echo "Marca de tiempo: ".time() . "<br/>";
echo "Raíz cuadrada de 9: ".sqrt(9) . "<br/>";
echo "Número aleatorio entre 10 y 20: ".rand(10, 20) . "<br/>";
echo "Número pi: ".pi();
```

5.1.1. Valores de las funciones

Cada función en PHP se considera como una expresión. Se pueden utilizar las funciones para lo mismo que se utilizan las expresiones, como por ejemplo almacenar su resultado en una variable o formar parte de una expresión más compleja. El ejemplo siguiente llama a la función `rand()`, que devuelve un número aleatorio que esté entre los dos números que se pasan como parámetro.

```
<?php
$numero_aleatorio = rand(1, 100);
echo $numero_aleatorio;
```

Si crea un fichero con este código y lo ejecuta, obtendrá un valor entero entre 1 y 100. Cada vez que actualice el navegador, obtendrá un número distinto.

No todas las funciones devuelven un valor numérico. También pueden devolver caracteres, *arrays* o `TRUE`/`FALSE` si la función ha tenido éxito o no. Este tipo de funciones se utiliza para conectar a bases de datos, escribir en ficheros de texto, manipular imágenes, etcétera.

Una base de datos MySQL contiene una o más tablas, cada una de las cuales contiene registros o filas. Dentro de estas filas, hay varias columnas o campos que contienen los datos en sí. Como ejemplo, la siguiente tabla muestra el contenido de una base de datos de ejemplo de cinco publicaciones que detallan el autor, el título y el año de la publicación.

Tabla 12.1. Ejemplo de una base de datos.

Autor	Título	Año
Robert C. Martin	*Código limpio*	2012
Steve Krug	*No me hagas pensar*	2015
Aurélien Géron	*Aprende Machine Learning*	2020

Cada fila de la tabla es igual a una fila de una tabla MySQL, cada columna corresponde a una columna en MySQL y cada elemento dentro de una fila es igual a un campo MySQL.

En este capítulo utilizará un tipo de base de datos relacional. Este tipo de base de datos es gestionado por un software que se conoce como **Sistema gestor de base de datos relacional** (SGBDR), que es el responsable de controlar las bases de datos, sus componentes, mantener la integridad de los datos, los usuarios que acceden, de recibir las consultas de petición, creación y actualización de la información y muchas otras funcionalidades.

El SGBDR que utilizaremos se llama MySQL, que está disponible como código abierto bajo licencia GPL. MySQL es un sistema de administración de bases de datos relacionales cliente/servidor, es decir, existe como una aplicación de software instalada y funcionando en una o múltiples máquinas en una red, que sirve información a través de esa red a aplicaciones de software instaladas en otras máquinas que solicitan datos o los envían para ser almacenados. Por lo tanto, lo más habitual es tener la versión servidor (MySQL Server) en una máquina en la red, que será donde almacenamos nuestros datos.

Es necesario tener una versión cliente que acceda al servidor para enviar y recibir datos. Como verá más adelante, con PHP ya viene incluida una extensión para trabajar como cliente de MySQL.

12.2. MySQL Workbench

En el capítulo 1 aprendimos a instalar PHP y MySQL; por lo tanto, ya tiene disponible el servidor de MySQL en su sistema. También vimos cómo instalar una aplicación cliente de escritorio que nos servirá para gestionar las bases de datos de nuestros proyectos: MySQL Workbench.

En este capítulo utilizaremos esa aplicación para conocer las principales características de MySQL. En la figura 12.1 puede observar la pantalla inicial de MySQL Workbench, que incluye la conexión al servidor MySQL que configuró en el capítulo 1.

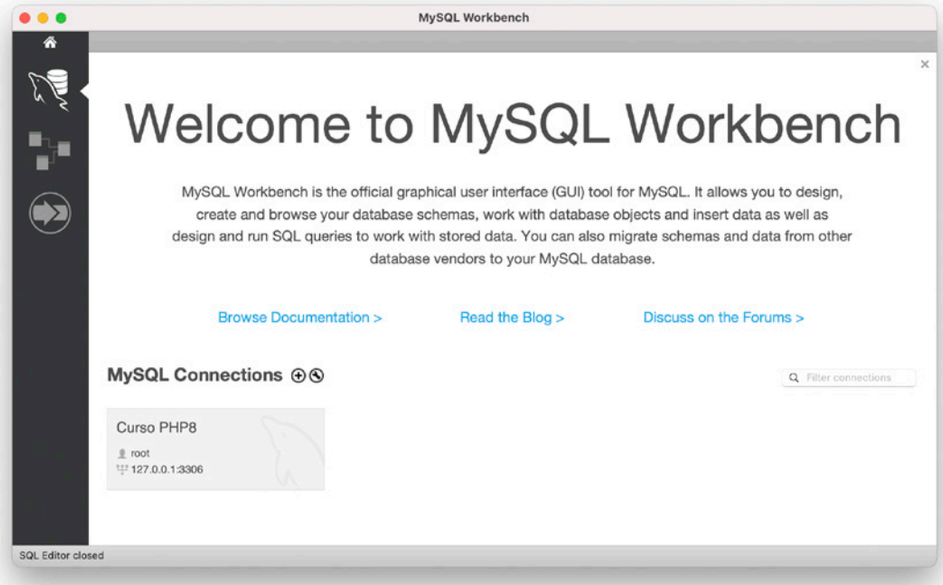


Figura 12.1. Vista inicial de MySQL Workbench.

En la vista de esta figura puede observar la distribución de los diferentes elementos que componen el entorno de trabajo de MySQL Workbench. En la zona de la izquierda, hay una columna con varios iconos con los que interactuar para acceder rápidamente a algunas funcionalidades. El que muestra una casa nos lleva a la pantalla inicial. Justo debajo, el icono del delfín nos permite acceder a las diferentes conexiones. A continuación, el icono de diagrama muestra una zona donde crear modelos, es decir, modelar gráficamente la base de datos utilizando diagramas. Y, por último, la flecha encerrada en un círculo nos lleva a un apartado para poder realizar migraciones utilizando un asistente. Una migración nos permite transformar una base de datos de otro gestor de bases de datos relacionales como PostgreSQL hacia MySQL y viceversa. En el apartado de conexiones aparece la conexión que creamos en el capítulo 1, Curso PHP8. Haga clic en ella para abrir el editor de MySQL que tiene el aspecto mostrado en la figura 12.2.

Se comprobará de nuevo el acceso al esquema seleccionado, haga clic en Continue y finalmente le mostrará la vista para confirmar la importación de las tablas de ese esquema. También ofrece la posibilidad de escoger qué tablas desea que sean importadas al modelo utilizando la opción Show Filter. Haga clic en el botón Execute para realizar la importación. A continuación, se le mostrará el resultado de ese proceso, haga clic en Continue para llegar a la vista de resumen completo del proceso y cierre la ventana haciendo clic en Close. Puede ver el resultado en la figura 12.17. Tiene la tabla Contactos dibujada como un rectángulo con el nombre de la tabla y la lista de campos con su tipo.

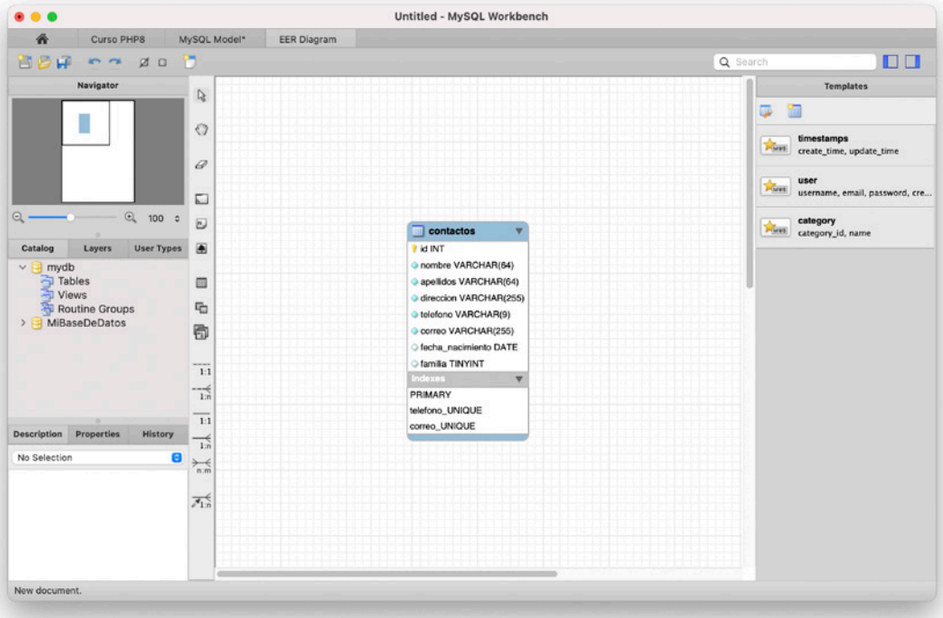


Figura 12.17. Importación de nuestro esquema a un modelo.

Para añadir la tabla Reuniones, haga clic en el icono de tabla del menú que se encuentra junto al borde del diagrama y después haga clic en cualquier zona del diagrama, como se ve en la figura 12.18.

Se creará una nueva tabla con el nombre table 1. Haga doble clic en ella para ver sus propiedades y así poder cambiarlas. El proceso de modificar el nombre y los campos de la tabla es el mismo que vimos cuando creamos la tabla Contactos. En la figura 12.19 puede ver el resultado. Observe que no hemos creado el campo `contactos_id`, lo creará a continuación utilizando el diagrama.

Figura 12.18. Añadir una tabla al diagrama.

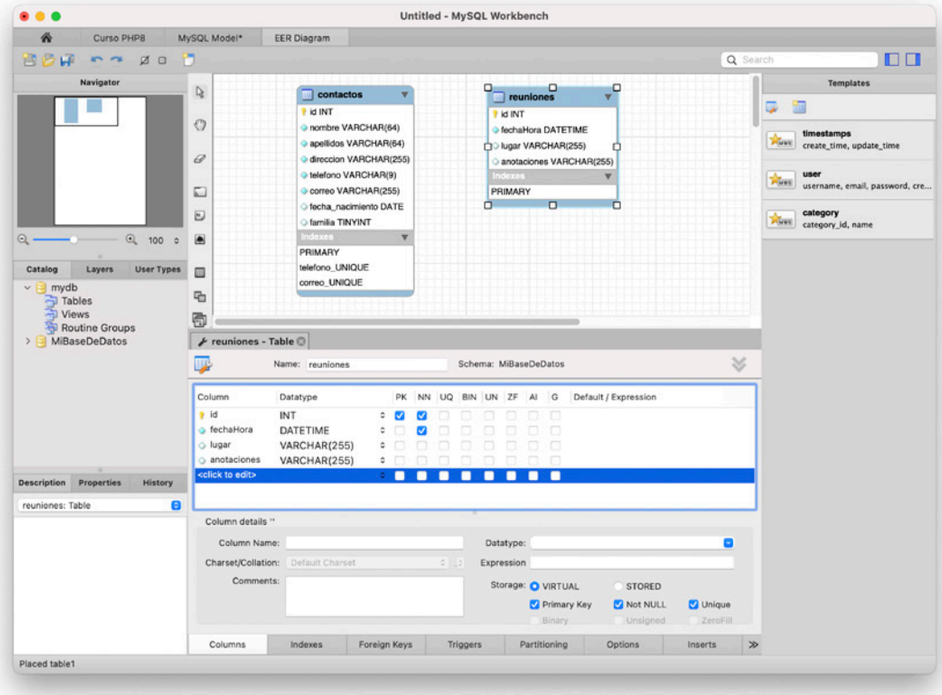


Figura 12.19. Configuración de la tabla Reuniones.

Ha llegado al paso más importante, el de la creación de la relación entre ambas tablas. Para ello, tiene que decidir en primer lugar qué tipo de relación va a establecer. Las tablas se pueden relacionar de muchas maneras. De una forma general, sin entrar en detalles de registros que no se relacionan con ninguno de otras tablas, una clasificación resumida de las relaciones puede ser:

Tabla 14.8. Tabla Clientes en tercera forma normal.

Id	Nombre	Dirección	Código Postal
1	Javier Arroyo Arias	C\Felipe Sanz, 34	28001
2	Francisco Martínez	Av. Alan Turing, 67	10600
3	Guillermo Ródena	C\Robert C. Martin, 25	46700
4	Gabriel Villena Díaz	C\Marquesa Pinares, 4	06800

Tabla 14.9. Tabla Códigos Postales en tercera forma normal.

Código Postal	Localidad Id
28001	1234
10600	5678
46700	4321
06800	8765

Tabla 14.10. Tabla Localidades en tercera forma normal.

Localidad Id	Nombre	Provincia Id
1234	Madrid	28
5678	Plasencia	10
4321	Gandía	43
8765	Mérida	06

Tabla 14.11. Tabla Provincias en tercera forma normal.

Provincia Id	Nombre	Abreviatura
28	Madrid	MA
10	Cáceres	CC
43	Valencia	VA
06	Badajoz	BA

Bueno, hemos pasado de tratar una sola tabla, la 14.6, a disponer de un total de cuatro. Aunque parece que esto complica la búsqueda, no se abrume. Primero, usaría el código postal en la tabla 14.8 (Clientes) para encontrar el identificador de la localidad en la tabla 14.9 (Códigos Postales). Con esta información, podría buscar el nombre de la localidad en la tabla 14.10 (Localidades) y, finalmente, el código de la provincia, que podría usar en la tabla 14.11 (Provincias) para obtener el nombre de la provincia.

Aunque usar la tercera forma normal de esta manera puede parecer demasiado exhaustivo, puede aprovecharse de ciertas ventajas. Por ejemplo, eche un vistazo a la tabla 14.11, donde se ha podido incluir tanto el nombre de una provincia como su abreviatura de dos letras. También podría contener detalles de población y otros datos demográficos, si lo desea.

En ocasiones, puede ser difícil decidir si usar la tercera forma normal. Para ello, debe tener claro si va a necesitar añadir datos en el futuro. Si es seguro que, en el caso de los clientes, todo lo que va a necesitar son su nombre y su dirección, probablemente querrá dejar fuera esta etapa final de normalización.

Por otra parte, supongamos que está escribiendo una base de datos para una gran organización como el servicio de Correos. ¿Qué pasaría si se cambiara el nombre de una localidad? Con una tabla como la tabla 14.6, habría que realizar una búsqueda global y reemplazar cada instancia de esa localidad. Pero, si tiene su base de datos diseñada para cumplir con la tercera forma normal, tendría que cambiar solo una entrada en la tabla 14.10 para que el cambio se reflejara en toda la base de datos.

Por lo tanto, le sugerimos que se haga dos preguntas que le ayudarán a decidir si debe llegar a normalizar hasta la tercera forma normal en cualquier tabla:

- ¿Es probable que en el futuro haya que añadir muchas columnas nuevas a esta tabla?
- ¿Podría cualquiera de los campos de esta tabla requerir una actualización global en cualquier momento?

Si alguna de las respuestas es afirmativa, probablemente debería considerar la posibilidad de realizar esta etapa final de normalización.

14.2.4. Cuándo no usar normalización

Ahora que ya conoce qué es la normalización y las formas normales, a continuación, le contamos por qué debería tirar estas reglas a la basura cuando se trate de sitios con tráfico alto. Así es, nunca debe normalizar completamente sus tablas en sitios donde esta normalización exhaustiva pueda provocar que MySQL se caiga, es decir, se bloquee y deje de dar servicio.

Como ha podido comprobar, la normalización requiere diseminar los datos a través de múltiples tablas. Esto implica que es necesario hacer múltiples llamadas a MySQL por cada consulta. En un sitio muy popular, si tiene tablas de tamaño normal, el acceso a su base de datos se ralentizará considerablemente una vez que supere unas pocas docenas de usuarios simultáneos, porque

17 React

En este capítulo aprenderás:

- Qué es React
- Crear y utilizar componentes React
- Algunos efectos interesantes
- Practicar React con algunos ejemplos básicos

17.1. Introducción

Durante mucho tiempo, las aplicaciones escritas en PHP siempre han ido acompañadas de alguna biblioteca JavaScript para ayudar a la parte visual a ser más atractiva y aportar una funcionalidad extra. En nuestro libro anterior, *Desarrollo web con PHP y MySQL. Edición 2018*, describimos cómo utilizar jQuery, una de las bibliotecas más utilizadas en el desarrollo web. A partir de 2018 comenzaron a despuntar una serie de utilidades que planteaban la interacción con el usuario de una forma distinta a como lo venía haciendo JavaScript de forma clásica. También conocimos que uno de los *frameworks* más utilizados de CSS, Bootstrap, abandonaba el uso de jQuery a favor de otra biblioteca más purista. Todo esto nos ha hecho decantarnos por React, una biblioteca JavaScript moderna que potencia la experiencia de uso de la web y que se acopla perfectamente en los *frameworks* modernos de PHP como Laravel.

React está desarrollado por Facebook y es tan extenso como el propio PHP; podríamos escribir un libro entero sobre su funcionamiento. En este capítulo, veremos la parte esencial para que pueda utilizarlo junto a sus proyectos PHP y, sobre todo, para que tenga una base para el siguiente capítulo sobre Laravel.

17.2. Primeros pasos

Al igual que PHP utiliza composer como sistema de paquetes, el ecosistema de JavaScript utiliza npm para gestionar la descarga de nuevas características y utilidades. En la imagen Docker de PHP, ya está todo listo para crear aplicaciones escritas con React. Entre en esta imagen a través de Visual Studio Code haciendo clic con el botón derecho sobre la imagen y eligiendo la opción Attach Shell.

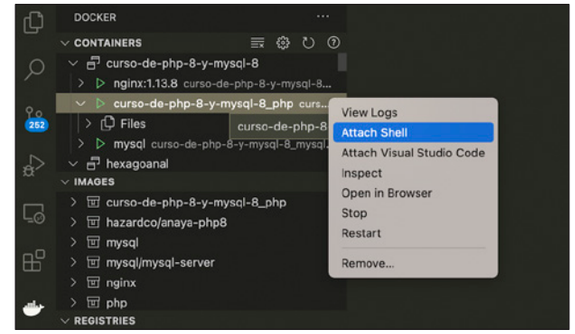


Figura 17.1. Conexión con la imagen Docker de PHP.

CURSO DE PHP 8 Y MySQL 8

Manual Imprescindible

PHP es uno de los lenguajes más utilizados en Internet y está presente en aplicaciones muy conocidas como WordPress, Joomla, Moodle o el propio Facebook. Desde sus inicios fue creado pensando en la sencillez de uso y en facilitar una curva de aprendizaje gradual al programador. MySQL es el sistema de gestión de bases de datos más popular y extendido en la actualidad.

Este libro es ideal para aprender de forma gradual a dominar las nuevas versiones de PHP y MySQL. Cubre todos los aspectos necesarios para aprender PHP 8, una evolución del lenguaje original en la que se han invertido numerosos años en mejorar todos los aspectos tecnológicos del lenguaje, modernizándolo constantemente. También MySQL 8 incorpora muchas novedades como el soporte para UTF8, el formato JSON, funcionalidades GIS y mucho más.

El libro puede dividirse en varios apartados lógicos: una primera parte como introducción a lo que se puede hacer con PHP seguida de una descripción de MySQL como motor principal de bases de datos. Una tercera parte que añade capítulos donde se mezclan PHP y MySQL y una última sobre aspectos avanzados. Además, cuenta con un capítulo introductorio a Laravel 8 para elevar tus conocimientos de PHP a la máxima potencia con este *framework*.

2311362
ISBN 978-84-415-4372-0
9 788441 543720

ANAYA MULTIMEDIA
www.anayamultimedia.es