

Desarrollo de
aplicaciones web con
PHP y MySQL



España - México - Colombia - Chile - Ecuador - Perú - Bolivia - Uruguay - Guatemala - Costa Rica

Desarrollo de aplicaciones web con PHP y MySQL

Autor: Manuel Ángel Torres Remon

Derechos reservados © Empresa Editora Macro EIRL

Primera edición: Empresa Editora Macro, Lima – Perú, enero 2015

Primera edición: MARCOMBO, S.A. 2016

© 2016 MARCOMBO, S.A.

«Cualquier forma de reproducción, distribución, comunicación pública o transformación de esta obra sólo puede ser realizada con la autorización de sus titulares, salvo excepción prevista por la ley. Diríjase a CEDRO (Centro Español de Derechos Reprográficos, www.cedro.org) si necesita fotocopiar o escanear algún fragmento de esta obra».

ISBN: 978-84-267-2326-0

D.L.: B-8570-2016

Impreso en Ulzama Digital SL

Printed in Spain

Manuel Ángel Torres Remon

Manuel Ángel Torres Remon es Licenciado en Informática. Se dedica a la consultoría y docencia de cursos de tecnología desde hace 15 años. Su formación académica la llevó a cabo en el Instituto Manuel Arévalo Cáceres, en la Universidad Alas Peruanas y en la Universidad Peruana del Norte.

En la actualidad, se desempeña como docente en el Instituto Superior Manuel Arévalo Cáceres, Cibertec y Unimaster, de la Universidad Nacional de Ingeniería, en los cuales imparte cursos de programación, base de datos y análisis de sistemas.

Entre sus publicaciones se encuentran *Fundamentos de programación con Visual Basic 2012*, *Programación orientada a objetos con Visual Basic 2012*, *Programación Transact con SQL Server 2012* y su última publicación fue dedicada al HTML5 y CSS3.

Para cualquier duda o consulta sobre el material puede escribir al *email*: manuel.torresr@hotmail.com.

Agradecimientos

Escuché alguna vez la frase “la lectura de un libro enriquece de conocimientos y empobrece la ignorancia”, pero hoy en día un libro impreso es difícil de leer, pues son pocas las personas que cargan uno en mano, ya que estos han sido reemplazados por los dispositivos móviles. Por tal motivo, mi primer agradecimiento es para usted amigo lector, por adquirir este material que simboliza sacrificio, tiempo y esfuerzo en comprobar que los códigos expuestos son los correctos.

Asimismo, deseo agradecer a la gran familia de la Editorial Macro por confiar nuevamente en mi persona para el desarrollo del presente libro: *Desarrollo de aplicaciones web con PHP y MySQL*, que cuenta con un fundamento expresado en el libro *Fundamentos de programación con PHP*, que recomiendo revisar si los códigos expuestos en este material resultan un tanto complejos.

Finalmente, quiero agradecer enormemente a mis pequeñas hijas Ángela Victoria y Fernanda Ximena Torres Lázaro, a mi esposa Luz y a mi querida suegra Victoria Menor Bautista que siempre tiene un comentario bueno respecto a mi persona y ello me motiva de gran manera a seguir adelante.

Dedicatoria

A mis pequeñas Ángela Victoria y Fernanda Ximena, que son mi fuente de inspiración, y a mi esposa Luz, por su comprensión en todo lo que me propongo.

Introducción	11
---------------------------	----

Capítulo 1

Introducción al MySQL

1.1 Introducción	15
1.2 Características	15
1.3 Versiones de MySQL.....	16
1.4 Descargar MySQL	17
1.4.1 Comprobar con WAMP Server	17
1.4.2 Comprobar con XAMPP	18
1.5 Uso de comandos básicos desde MySQL Console.....	19
1.6 MySQL Workbench.....	21
1.6.1 Descargar MySQL Workbench	21
1.6.2 Instalar MySQL Workbench	22
1.6.3 Pantalla inicial de MySQL Workbench y conexión a MySQL	24
1.7 MySQL-Front	25
1.7.1 Descargar MySQL-Front	25
1.7.2 Instalar MySQL-Front	26
1.7.3 Crear una nueva conexión a MySQL	29
1.8 Sentencias MySQL para el objeto de base de datos.....	30
1.8.1 Crear una base de datos	30
1.8.2 Listar las bases de datos registradas en el servidor	31
1.8.3 Activar la base de datos	31
1.8.4 Mostrar la base de datos activa	32
1.8.5 Eliminar la base de datos	32
1.9 Sentencias MySQL para el objeto tabla.....	33
1.9.1 Motor de almacenamiento	33
1.9.2 Tipos de datos	34
1.9.3 Crear una tabla	36
1.9.4 Agregar columna a la tabla	38
1.9.5 Eliminar columnas de una tabla.....	38
1.9.6 Implementar columnas con valores nulos	39
1.9.7 Implementar columnas con valores por defecto	40
1.9.8 Verificar la estructura de una tabla.....	42
1.9.9 Restringir la duplicidad de valores.....	43
1.9.10 Crear tablas relacionales.....	44
1.9.11 Asignar la llave primaria.....	45
1.9.12 Visualizar la estructura de la tabla desde MySQL-Front y MySQL Workbench	47
1.9.13 Asignar la llave foránea.....	50
1.9.14 Implementar un esquema de base de datos	51
1.10 Sentencia de Manipulación de Datos (SMD)	56
1.10.1 Sentencia INSERT	56
1.10.2 Sentencia UPDATE.....	58
1.10.3 Sentencia DELETE.....	60

1.10.4 Sentencia TRUNCATE	61
1.11 Consulta básica con la sentencia SELECT.....	61
1.11.1 Selección de datos sin restricciones	61
1.11.2 Selección de datos limitando las columnas de la tabla.....	62
1.11.3 Restringir la muestra de datos repetidos.....	63
1.11.4 Ordenar de forma ascendente o descendente los registros de una tabla	63
1.11.5 Limitar los registros de la consulta.....	64
1.12 Combinación de tablas con INNER JOIN	64
1.12.1 Combinación de tablas con LEFT JOIN	66
1.12.2 Combinación de tablas con RIGHT JOIN	67
1.13 Recuperación de datos agrupados	68
1.13.1 Cláusula GROUP BY	68
1.13.2 Función agregada COUNT	69
1.13.3 Función agregada SUM	71
1.13.4 Función agregada MAX	72
1.13.5 Función agregada MIN	72
1.13.6 Función agregada AVG	73
1.13.7 Cláusula HAVING.....	74

Capítulo 2

Integración MySQL con PHP

2.1 Introducción	77
2.2 Conectar MySQL a PHP.....	77
2.2.1 Función de conexión al servidor MySQL.....	77
2.2.2 Función de cierre del servidor	79
2.2.3 Función de selección de base de datos desde el servidor	80
2.2.4 Función que ejecuta una sentencia de consulta MySQL en PHP	82
2.2.5 Función que determine el total de registros devueltos por la función mysql_query usando la sentencia SELECT de MySQL.....	83
2.2.6 Función que muestre los registros de una consulta mysql_query en PHP	85
2.2.7 Casos desarrollados	91
Caso desarrollado 1: Listado de clientes por distrito.....	91
Caso desarrollado 2: Listado de boletas por cliente	97
Caso desarrollado 3: Listado de clientes con imágenes	101
Caso desarrollado 4: Listado de clientes paginado con imágenes	105
2.2.8 Uso de registros mediante arreglos de índice	110
2.2.9 Uso de registros mediante arreglos asociativos	112
2.2.10 Uso de registros mediante arreglos asociativos y foreach.....	115
2.2.11 Uso de registros mediante procedimientos almacenados	117
Caso desarrollado 5: Consulta de productos por código	120

Capítulo 3

Mantenimiento de registros

3.1 Inserción de registros	127
3.1.1 Casos desarrollados	132
Caso desarrollado 1: Registro del nuevo cliente.....	132
Caso desarrollado 2: Registro del nuevo cliente adjuntando su imagen.....	142
3.2 Actualización de registros.....	152
Caso desarrollado: Actualización de datos del cliente.....	159
3.3 Eliminación de registros	169
Caso desarrollado: Eliminando registro de cliente	173
3.4 Casos desarrollados.....	184
Caso desarrollado 1: Mantenimiento de registro de clientes.....	184
Caso desarrollado 2: Mantenimiento de registro de clientes paginado	195

Capítulo 4

Reportes

4.1 Clase FPDF().....	209
4.2 Principales funciones de la librería FPDF.....	210
4.2.1 Objeto FPDF	210
4.2.2 Añadir una nueva página PDF	211
4.2.3 Definir el tipo de letra y tamaño del texto	212
4.2.4 Definir una celda para un elemento textual	213
4.2.5 Definir la posición de un elemento textual.....	214
4.2.6 Definir el color de un elemento	215
4.2.7 Definir una línea recta	216
4.2.8 Definir un rectángulo	217
4.2.9 Implementar una imagen en el documento PDF	218
4.2.10 Definir la salida de información	218
4.2.11 Casos desarrollados	219
Caso desarrollado 1: Reporte simple de clientes PDF.....	219
Caso desarrollado 2: Reporte de clientes con <i>banner</i>	221
Caso desarrollado 3: Reporte de clientes con imágenes	224
Caso desarrollado 4: Generación de fotocheck	227
4.3 Clase PHPLOT.....	232
4.3.1 Tipos de gráficos	232
4.3.2 Caso desarrollado: Generación de gráficos estadísticos	233

Capítulo 5

Uso de imágenes

5.1 Introducción	239
5.1.1 Casos desarrollados	239
Caso desarrollado 1: Gráfico básico-Listado de clientes por distrito	239
Caso desarrollado 2: Gráfico con PhpLot-Reporte de boletas por mes-Año 2015....	243
Caso desarrollado 3: Gráfico con PHPLot-Listado de boletas por cliente	248
Caso desarrollado 4: Gráfico con Google Charts-Listado de boletas por cliente	258

Capítulo 6

Programación orientada a objetos

6.1 Introducción	267
6.2 Clases en PHP	267
6.3 Encapsulamiento de datos	270
6.4 Objetos en PHP.....	271
6.5 Método constructor en PHP.....	271
6.6 Casos desarrollados.....	272
Caso desarrollado 1: Clase/objeto-Venta de productos	272
Caso desarrollado 2: Método constructor-Pago de trabajadores.....	280
Caso desarrollado 3: Método constructor-Listado de productos con MySQL	286
Caso desarrollado 4: Mantenimiento de productos orientado a objetos.....	291
Caso desarrollado 5: Mantenimiento de productos con MVC.....	305

Capítulo 7

Objetos de datos PHP (PDO)

7.1 Introducción	331
7.2 Conexión PDO.....	331
7.3 Casos desarrollados.....	332
Caso desarrollado 1: Listado de clientes.....	332
Caso desarrollado 2: Consulta de datos de cliente por código.....	336
Caso desarrollado 3: Listado de clientes por distrito.....	339

Bibliografía	343
---------------------------	------------

Introducción

El libro *Desarrollo de aplicaciones web con PHP y MySQL* le permitirá crear aplicaciones web usando base de datos y, además, le orientará en la programación de objetos.

Para cumplir con dicho objetivo se ha dividido el libro en siete capítulos que son distribuidos de la siguiente manera: el capítulo uno presenta una breve introducción sobre las sentencias MySQL, este capítulo es la base del desarrollo del libro, pues al comprender dichas sentencias podrá generar aplicaciones sin ningún problema de datos. Se sugiere que repase este tema cuando lo crea conveniente y en cualquier capítulo.

El capítulo dos hace referencia a la integración de PHP con MySQL. Se emplean técnicas de conexión al servidor de la base de datos, así como el permiso de uso de la misma. Con ello se aprenderá a mostrar el listado de registros conectados a la base de datos con casos desarrollados y explicados.

En el capítulo tres se aplican los métodos necesarios para un mantenimiento de registros adecuado, como agregar, eliminar o actualizar los datos de una tabla de base de datos, se presentan casos desarrollados y explicados.

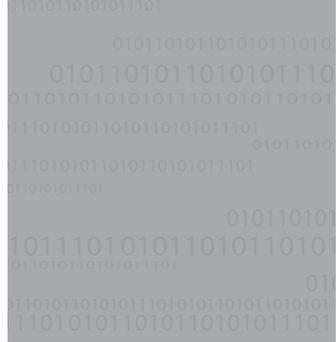
Por otra parte, un punto importante dentro de los proyectos web son los informes o reportes que se pueden producir a partir de los datos de una base de datos, tal es así que en el capítulo cuatro se implementan aplicaciones con reportes, usando la clase FPDF.

Otra tarea prioritaria de las aplicaciones web es el manejo de imágenes de forma dinámica, tal es así que el capítulo cinco está dedicado al desarrollo de aplicaciones, se emplean imágenes gráficas referentes a los valores de la base de datos, esto se analizará mediante el uso de las clases PHPLot y Google Charts.

De acuerdo a los supuestos anteriores, se está en la capacidad de crear aplicaciones orientadas a objetos, es decir, se implementará aplicaciones usando la clase y sus objetos a partir de casos desarrollados básicos, como el método constructor, métodos GET/SET y los objetos. Asimismo, se emplea la programación orientada a objetos para conectarnos a una base de datos e implementar aplicaciones web de mantenimiento de registros, usando puras clases.

Finalmente, en el capítulo siete se hace uso de la programación orientada a objetos, mediante la clase PDO de PHP. Además, se muestran casos desarrollados de mantenimiento de registros.

El autor confía en que este libro resulte útil, en particular, a los estudiantes y profesionales para quienes el desarrollo de aplicaciones web con PHP y MySQL sea de su interés.



CAP.

1

Introducción al MySQL

1.1 Introducción

MySQL es considerado como un sistema de gestión de bases de datos relacional, el cual permite administrar los registros, actualización y eliminación de la información tal como lo haría un SQL Server u Oracle.

La característica principal de PHP es el tema de los derechos, debido a que cuenta con una licencia dual, por una parte es considerado como *software* libre bajo la licencia GPL (licencia pública general) y por otra es necesaria la compra de la licencia.

Se atribuye el costo de licencia a la empresa sueca MySQL AB desde los primeros meses del 2008, esta a su vez es una subsidiaria de la empresa Sun Microsystems y esta de Oracle Corporation a partir de abril de 2009.

Se considera adquirir la licencia cuando modificamos el código MySQL y queremos redistribuir aplicando un costo por dicha distribución. Dicho de otro modo, lo que es prohibido es distribuir algo que hemos obtenido bajo la licencia GPL. Eso quiere decir que podemos usar el servidor MySQL pero no incluirlo dentro del *software* desarrollado, ya que eso distribuiría a MySQL y, por tanto, rompe las reglas GPL. Se recomienda que el *software* que se desarrollará también se distribuya bajo la misma licencia.

A continuación, algunas políticas que se pueden aplicar bajo la licencia GPL:

- Implementar un sistema de información en una organización que solo use al servidor MySQL.
- Se puede modificar el código fuente de MySQL las veces que crea necesaria mientras no distribuya a MySQL con dichos cambios.
- Se permite la distribución de MySQL a pesar de sus modificaciones, pero siempre bajo la licencia GPL.
- Si el sistema de información solo usa a MySQL y no lo distribuye, el sistema puede ser privativo y beneficiarse de él.

Como se mencionó, MySQL fue creada por la empresa MySQL AB establecida en Suecia en el año 1995 con el objetivo principal de cumplir con el estándar SQL (Lenguaje de consultas estructurado), sin sacrificar la velocidad, fiabilidad o usabilidad demostrada por otros SGBD.

Se sostiene que el nombre «MySQL» es heredado de su modelo anterior llamado mSQL, que presentaba algunas deficiencias y al cual le incorporaron API para su mejoramiento, denominándolo así MySQL. Otros señalan que las letras iniciales «My» son colocadas en honor a la hija de uno de los creadores. En el caso del logotipo, el nombre del delfín de MySQL es «Sakila» y fue asignado así por los fundadores de la empresa MySQL AB, proponiendo un concurso para dicho nombre llamado *Name the Dolphin*.



Logotipo oficial de MySQL

1.2 Características

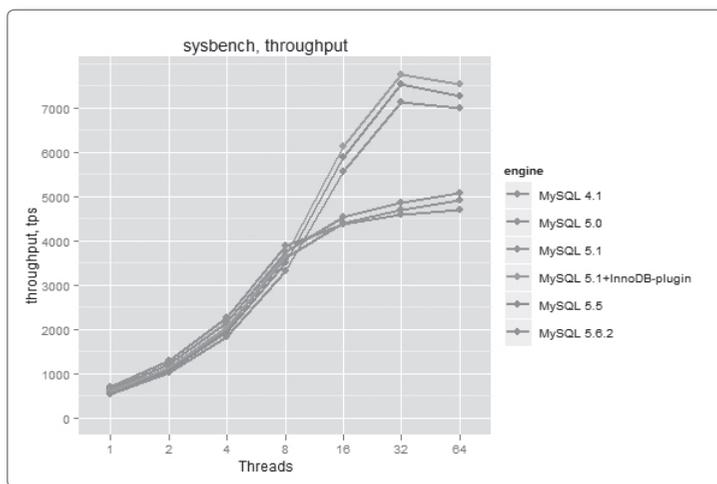
Como ya se había mencionado, uno de los puntos principales de MySQL es la licencia dual que mantiene para su uso, pero, además, presenta las siguientes características:

- MySQL fue escrito en su totalidad en el lenguaje C y C++.
- Es considerado como multiplataforma, ya que se puede ejecutar en diferentes sistemas operativos. Nosotros lo implementaremos en Windows.

- Es considerado como un sistema portable de licencia GPL, ya que usa GNU Automake (herramienta de programación que produce programas makefiles portables usado en la compilación de *software* e implementando especialmente en Linux y Unix), Autoconfig y Libtool.
- Contiene una interfaz de programación de aplicaciones (API) que permite usar MySQL en lenguajes de programación como C, C++, Java, PHP, Perl y otros lenguajes.
- Proporciona un sistema de almacenamiento transaccional y no transaccional con un sistema de reserva de memoria muy rápido basado en hilos y multihilos.
- Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- Cuenta con un soporte completo para las sentencias SELECT y WHERE, tal como lo maneja SQL Server.
- Cuenta con un sistema de privilegios y contraseñas que es muy flexible y seguro, suficiente para una gestión de la información. Las contraseñas resultan ser totalmente seguras, ya que estas son cifradas cuando se conecta con un servidor.
- Cuenta con un soporte importante de almacenamiento de base de datos, se han probado con más 60 000 tablas y cerca de 5 000 000 000 000 de registros.
- Los clientes de MySQL pueden conectarse al servidor usando sockets del TCP/IP.

1.3 Versiones de MySQL

Todas las versiones de MySQL cumplen de manera correcta el objetivo del SGBD. En este material trabajaremos con la versión 5.5 que resulta la versión más estable.



Fuente: <<http://www.genbetadev.com/bases-de-datos/mysql-ha-mejorado-el-rendimiento-al-subir-de-version>>

1.4 Descargar MySQL

Entre en la siguiente URL: <http://dev.mysql.com/downloads/>

MySQL Community Downloads

MySQL Community Server (GPL)
 (Current Generally Available Release: 5.6.20)
 MySQL Community Server is the world's most popular open source database.

[DOWNLOAD](#)

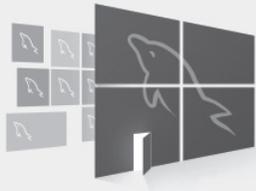
De la imagen, seleccione **MySQL Community Server (GPL)**. Desde aquí podrá elegir la versión adecuada y especialmente escoger el tipo de sistema del cliente que puede ser 32 o 64 bits.

Recommended Download:

MySQL Installer 5.6 for Windows

**All MySQL Products. For All Windows Platforms.
In One Package.**

Starting with MySQL 5.6 the MySQL Installer package replaces the server-only MSI packages.



Download

Windows (x86, 64-bit), MySQL Installer MSI

Other Downloads:

Windows (x86, 32-bit), MSI Installer	5.6.20	44.8M	Download
<small>(mysql-5.6.20-win32.msi) MD5: 59abb64af27634abd0f65a60204b18ab Signature</small>			
Windows (x86, 64-bit), MSI Installer	5.6.20	47.3M	Download
<small>(mysql-5.6.20-winx64.msi) MD5: 503dc2840c6732ae3e5dc80a3022f1a7 Signature</small>			
Windows (x86, 32-bit), ZIP Archive	5.6.20	337.6M	Download
<small>(mysql-5.6.20-win32.zip) MD5: 28528dd2cecd79d33dec1a2e1b7058c Signature</small>			
Windows (x86, 64-bit), ZIP Archive	5.6.20	342.9M	Download
<small>(mysql-5.6.20-winx64.zip) MD5: 08028c89f892534114550c75f57f3453 Signature</small>			

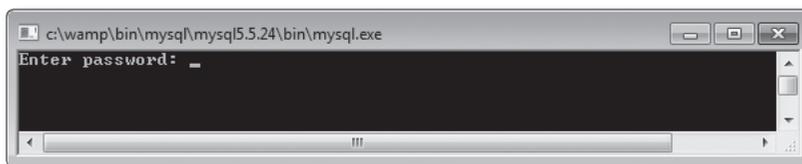
Desde esta ventana se recomienda revisar el tipo de sistema de su ordenador para poder seleccionar la descarga de MySQL adecuada para su sistema.

1.4.1 Comprobar con WAMP Server

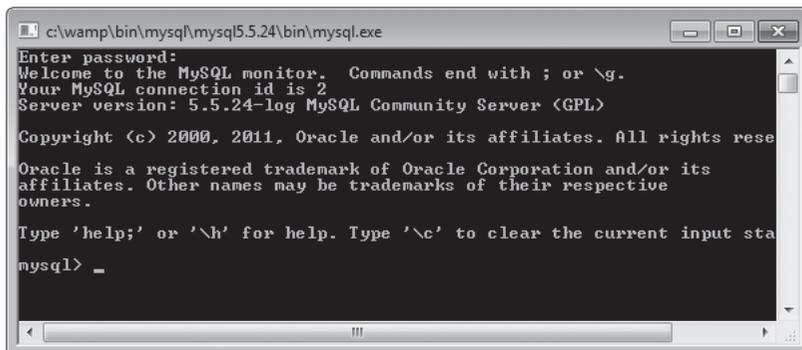
Se debe tener en cuenta que si ha instalado WAMP Server o XAMPP, como servidor de aplicaciones para PHP, entonces ya tiene instalado MySQL. Veamos cómo comprobar con WAMP Server:

- Cargue el servidor WAMP Server.
- Seleccione  residente en las aplicaciones de memoria situada en la barra de tareas de Windows.

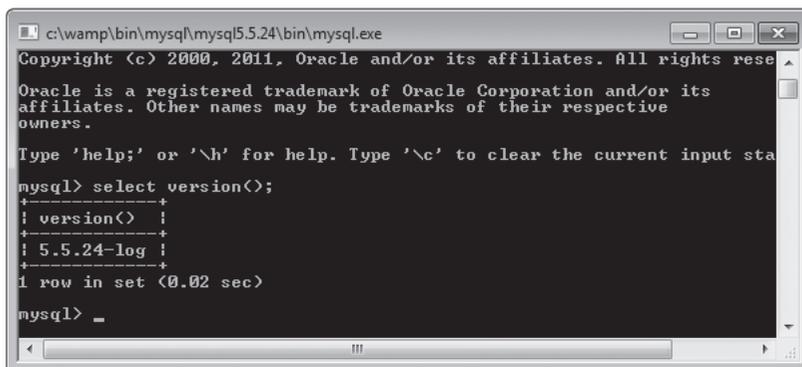
- Seleccione **MySQL > MySQL Console**.



- Desde la ventana anterior, el servidor solicita la clave de acceso, para lo cual solo debe pulsar la tecla **Enter** y visualizará la siguiente ventana:

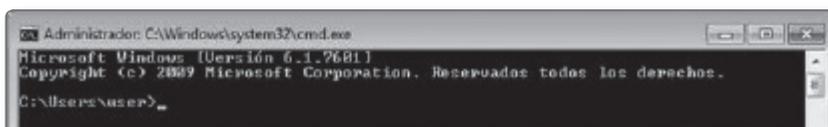


- Para comprobar si todo es correcto, puede introducir la siguiente instrucción: `select version();`



1.4.2 Comprobar con XAMPP

- Asegúrese de que otro servidor web no se encuentre ejecutado en nuestro ordenador.
- Cargue el servidor XAMPP.
- Verifique que en la barra de tareas se encuentre el icono .
- Pulse la combinación de teclas **Windows + R** y coloque la instrucción `CMD`.



- Coloque las siguientes instrucciones en estricto orden para llegar al servidor MySQL.

```

C:\Windows\system32\cmd.exe - mysql -h localhost -u root -p
Microsoft Windows [Versión 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\user>cd\
C:\>cd XAMPP
C:\xampp>cd mysql\bin
C:\xampp\mysql\bin>mysql -h localhost -u root -p
Enter password:
  
```

- Seguidamente, se le solicitará el *password* y, como no se ha configurado, solo pulse la tecla **Enter** y pruebe el servidor con la sentencia: **select version();**

```

C:\Windows\system32\cmd.exe - mysql -h localhost -u root -p
Copyright (c) 2009 Microsoft Corporation. Reservados todos los derechos.

C:\Users\user>cd\
C:\>cd XAMPP
C:\xampp>cd mysql\bin
C:\xampp\mysql\bin>mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.0.51b-community MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select version();
+-----+
| version() |
+-----+
| 5.0.51b-community |
+-----+
1 row in set (0.00 sec)

mysql>
  
```

1.5 Uso de comandos básicos desde MySQL Console

Pruebe algunos comandos MySQL para comprobar que el servidor trabaja de manera correcta:

```
show databases;
```

Permite mostrar las bases de datos almacenadas en el servidor.

```

C:\windows\system32\cmd.exe - mysql -h localhost -u root -p

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cdcol |
| mysql |
| phpmyadmin |
| test |
| webauth |
+-----+
6 rows in set (0.01 sec)

mysql>
  
```

use mysql;

Permite activar una base de datos registrada en el servidor.

```

C:\windows\system32\cmd.exe - mysql -h localhost -u root -p
mysql> use mysql;
Database changed
mysql> _
  
```

show tables;

Permite listar las tablas de una base de datos activa.

```

C:\windows\system32\cmd.exe - mysql -h localhost -u root -p
mysql> show tables;
+-----+
| Tables_in_mysql |
+-----+
| columns_priv    |
| db              |
| event          |
| func           |
| general_log     |
| help_category  |
| help_keyword   |
| help_relation  |
| help_topic     |
| host           |
| ndb_binlog_index |
| plugin         |
| proc          |
| procs_priv     |
| servers        |
| slow_log       |
| tables_priv    |
| time_zone      |
| time_zone_leap_second |
| time_zone_name |
| time_zone_transition |
| time_zone_transition_type |
| user          |
+-----+
23 rows in set (0.00 sec)

mysql>
  
```

show fields from user;

Permite listar los campos de una tabla, para este caso se ha listado los campos de la tabla user.

```

C:\windows\system32\cmd.exe - mysql -h localhost -u root -p
mysql> show fields from user;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Defau |
|-----+-----+-----+-----+-----+
| Host  | char(60) | NO | PRI |       |
| User  | char(16) | NO | PRI |       |
| Password | char(41) | NO |     |       |
| Select_priv | enum('N','Y') | NO |     | N |
| Insert_priv | enum('N','Y') | NO |     | N |
| Update_priv | enum('N','Y') | NO |     | N |
| Delete_priv | enum('N','Y') | NO |     | N |
| Create_priv | enum('N','Y') | NO |     | N |
| Drop_priv | enum('N','Y') | NO |     | N |
| Reload_priv | enum('N','Y') | NO |     | N |
| Shutdown_priv | enum('N','Y') | NO |     | N |
| Process_priv | enum('N','Y') | NO |     | N |
+-----+-----+-----+-----+-----+
  
```

1.6 MySQL Workbench

En vista de que sería un poco engorroso trabajar en un entorno tipo D.O.S, se usa una herramienta visual como MySQL Workbench, que integra la administración de bases de datos, el diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL, etc. Todo lo que un sistema de gestión de base de datos necesita.

1.6.1 Descargar MySQL Workbench

1. Entre en la siguiente URL: <http://dev.mysql.com/downloads/>
2. Seleccione **MySQL Workbench**.

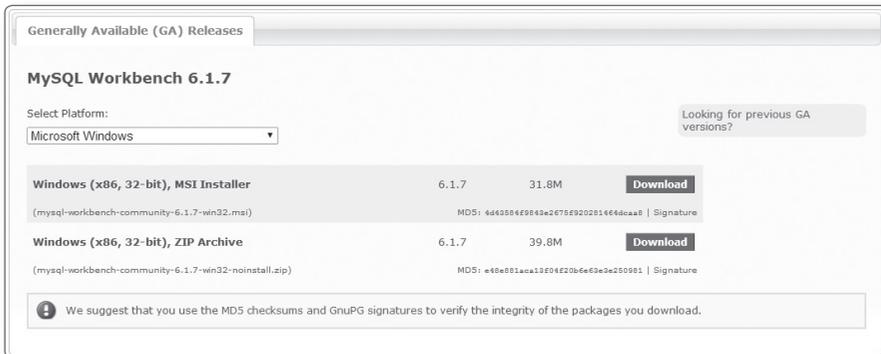
MySQL Workbench (GPL)

(Current Generally Available Release: 6.1.7)

MySQL Workbench is a next-generation visual database design application that can be used to efficiently design, manage and document database schemata. It is available as both, open source and commercial editions.

DOWNLOAD

3. Seleccione **Download**.



Generally Available (GA) Releases

MySQL Workbench 6.1.7

Select Platform:
 Looking for previous GA versions?

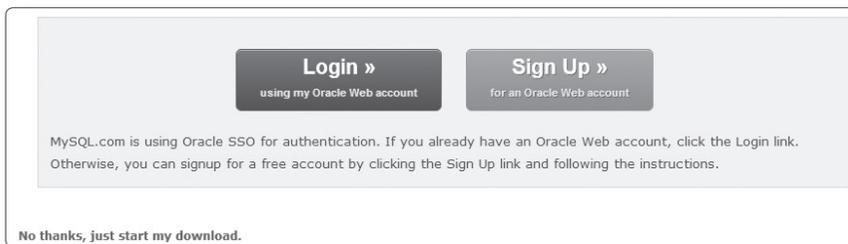
Windows (x86, 32-bit), MSI Installer <small>(mysql-workbench-community-6.1.7-win32.msi)</small>	6.1.7	31.8M	Download
Windows (x86, 32-bit), ZIP Archive <small>(mysql-workbench-community-6.1.7-win32-noinstall.zip)</small>	6.1.7	39.8M	Download

MD5: 4642854f9843a2675f9202816646caa8 | Signature

MD5: e48e881a112f94e20b6e6e3e2c0981 | Signature

! We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

4. Seleccione «No thanks, just start my download».



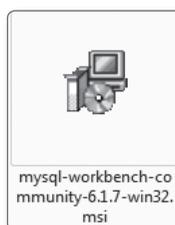
Login »
using my Oracle Web account

Sign Up »
for an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

No thanks, just start my download.

5. Finalmente, revise la sección de descarga de su ordenador.



1.6.2 Instalar MySQL Workbench

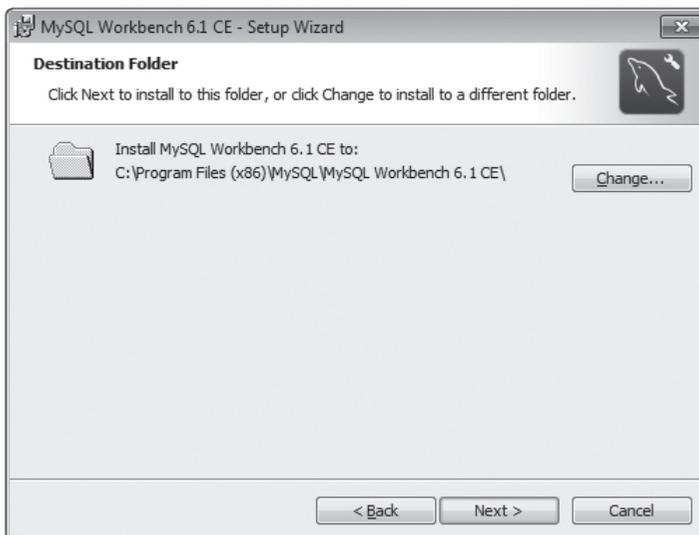
Recuerde que antes de iniciar la instalación de MySQL Workbench debe tener MySQL instalado en su ordenador personal, ya sea usando el propio *software* MySQL o por medio de WAMP Server o XAMPP.

Realice los siguientes pasos:

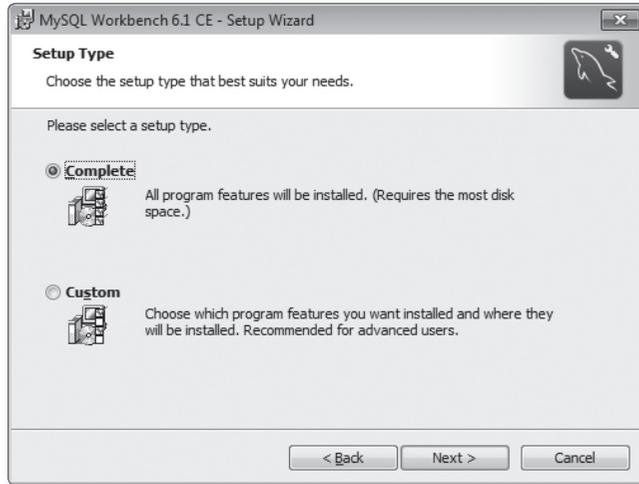
1. Inicie el proceso de instalación de **MySQL Workbench**.



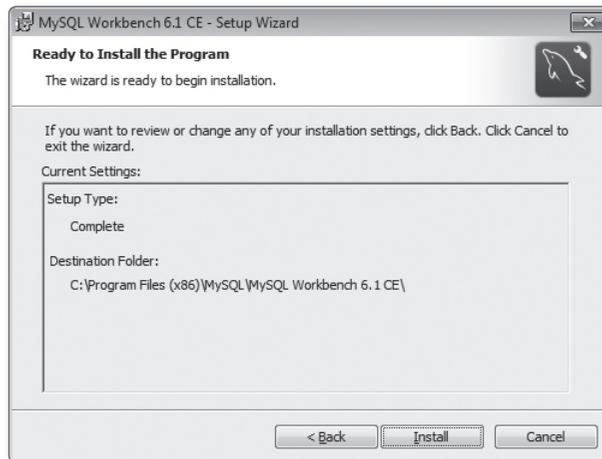
2. No será necesario cambiar la ubicación del lugar de la instalación, así que en el proceso de instalación se procederá a seleccionar **Next** en todas las ventanas.



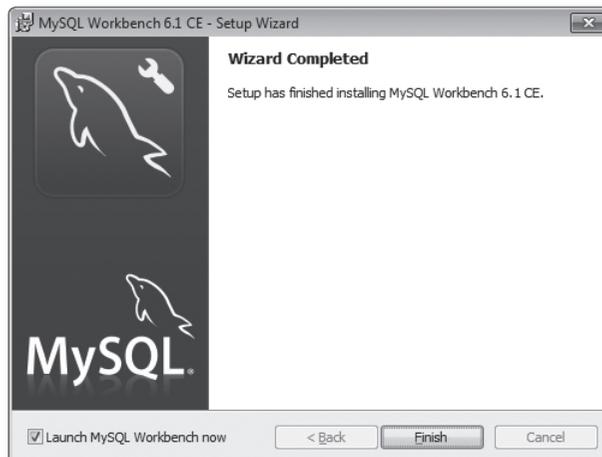
3. En el proceso de instalación, seleccione el tipo **Completo**, ya que el *software* viene con todas las especificaciones necesarias para la instalación.



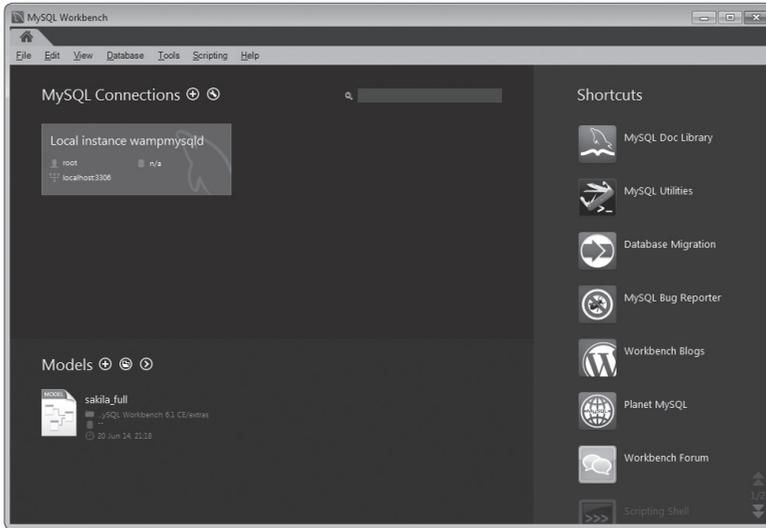
4. Finalmente, inicie la instalación seleccionando la opción **Install**.



5. La siguiente ventana muestra el final de la instalación.



1.6.3 Pantalla inicial de MySQL Workbench y conexión a MySQL



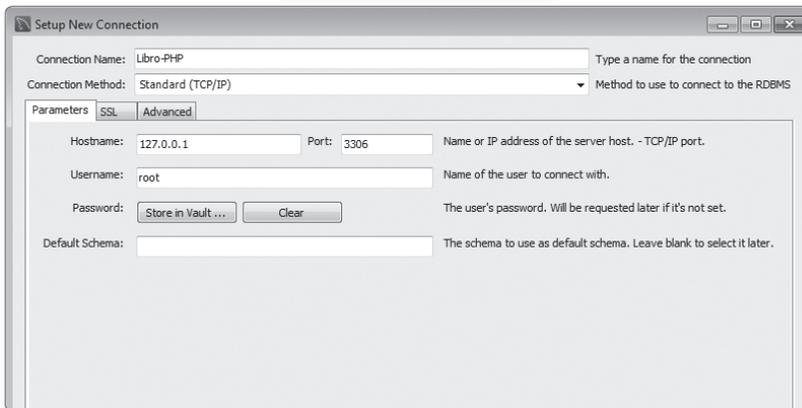
- **Creando una nueva conexión a MySQL**

Para crear y administrar información de MySQL, se debe crear una nueva conexión que permitirá tener acceso a todos los comandos MySQL.

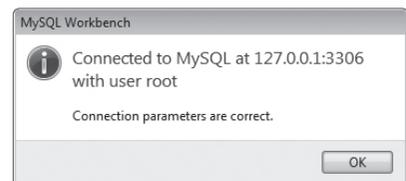
Realice los siguientes pasos:

1. Desde la ventana principal seleccione la opción +.

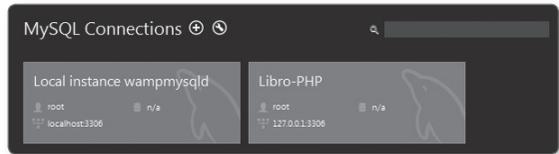
MySQL Connections + ↻



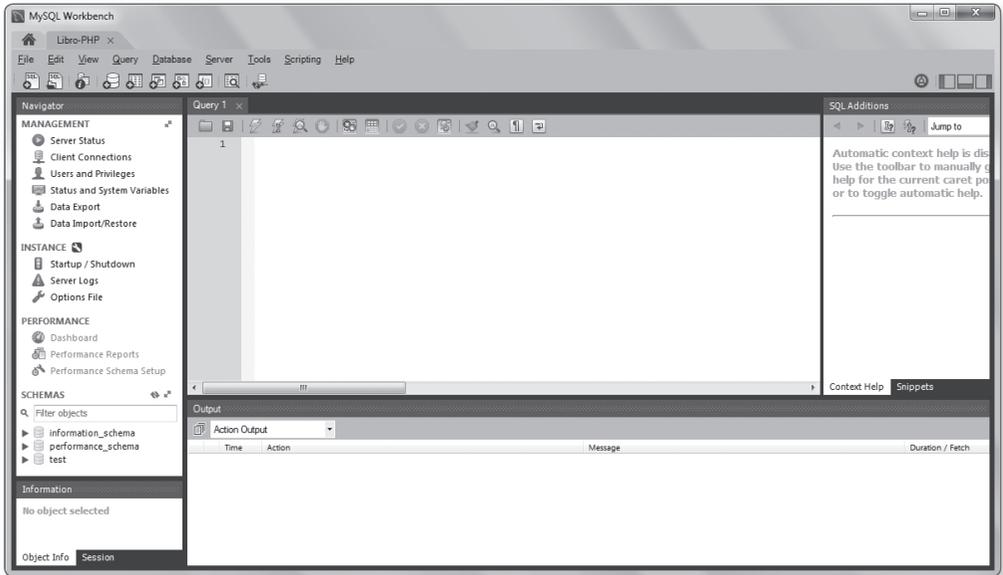
2. Desde la ventana **Setup New Connection** deberá registrar un nombre a la conexión, verifique que el **Username** sea «root» y, finalmente, seleccione la opción **Test Connection**. Si todo es correcto debe mostrarse la siguiente ventana:



3. Seguidamente, se mostrará en la pantalla principal la conexión realizada tal como se muestra en la siguiente imagen:



4. Para comenzar a trabajar con MySQL, debe seleccionar la conexión realizada y se mostrará la pantalla principal del entorno. Desde aquí podrá administrar toda la información que concierne a un proyecto cliente-servidor con MySQL.

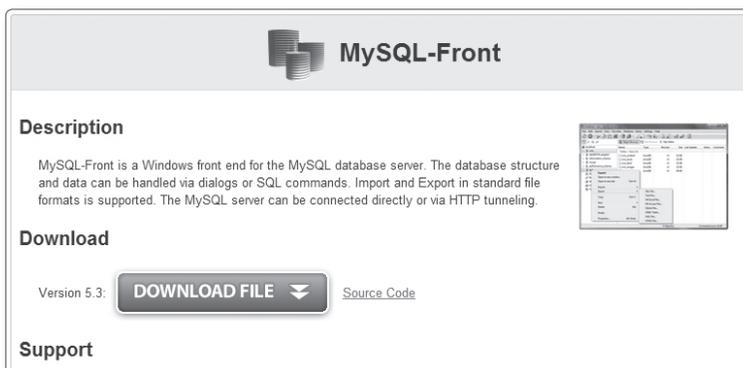


1.7 MySQL-Front

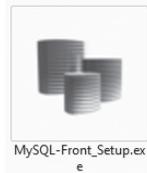
Es una aplicación que permite administrar la información de manera sencilla y es de fácil uso para el usuario de MySQL. Presenta un diseño bastante intuitivo que permite concentrarse en las sentencias de MySQL como una secuencia de comandos así como de manera visual.

1.7.1 Descargar MySQL-Front

1. Entre en la siguiente URL: <http://www.mysqlfront.de/>
2. Seleccione la opción **DOWNLOAD FILE**.



3. Finalmente, revise la sección de descarga de su ordenador.



1.7.2 Instalar MySQL-Front

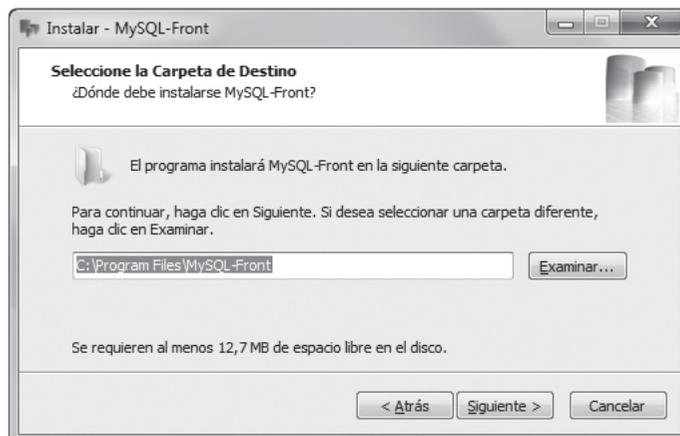
Recuerde que antes de iniciar la instalación de MySQL-Front debe contar con MySQL instalado en su ordenador personal, ya sea usando el propio *software* MySQL o por medio de WAMP Server o XAMPP.

Realice los siguientes pasos:

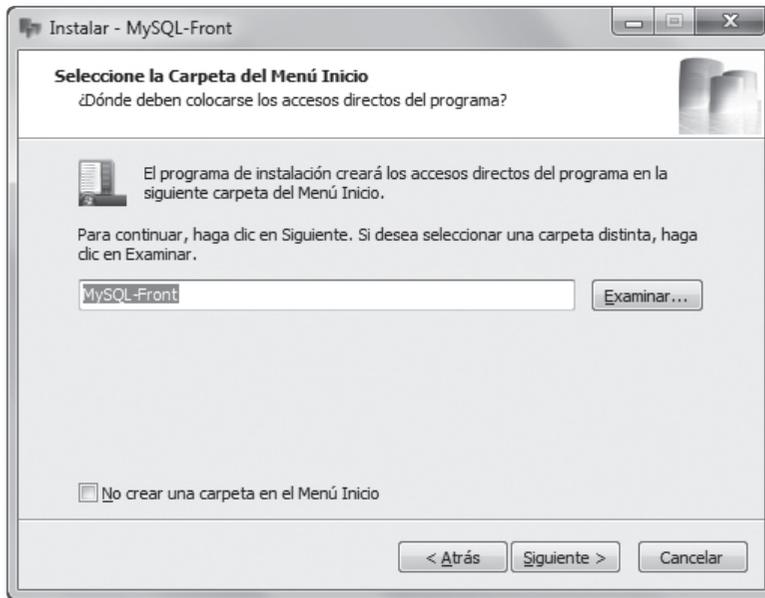
1. Inicie el proceso de instalación de MySQL-Front.



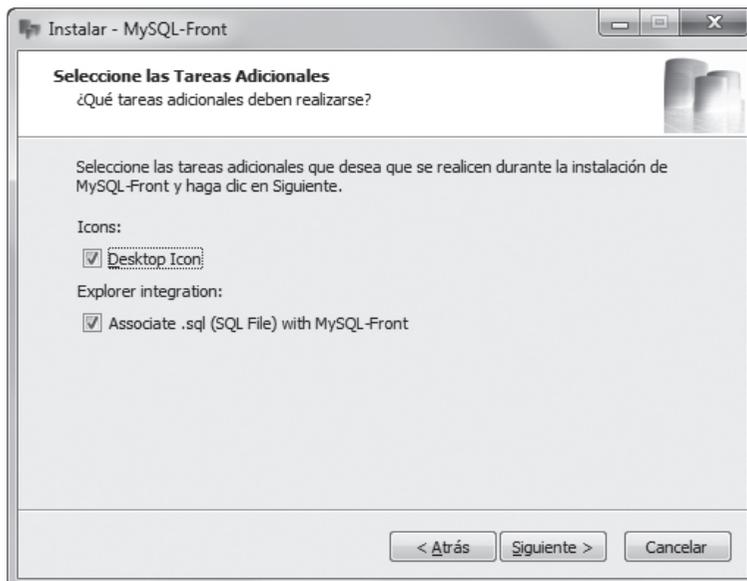
2. No será necesario cambiar la ubicación del lugar de la instalación, así que en el proceso de instalación deberá seleccionar **Siguiente** en todas las ventanas.



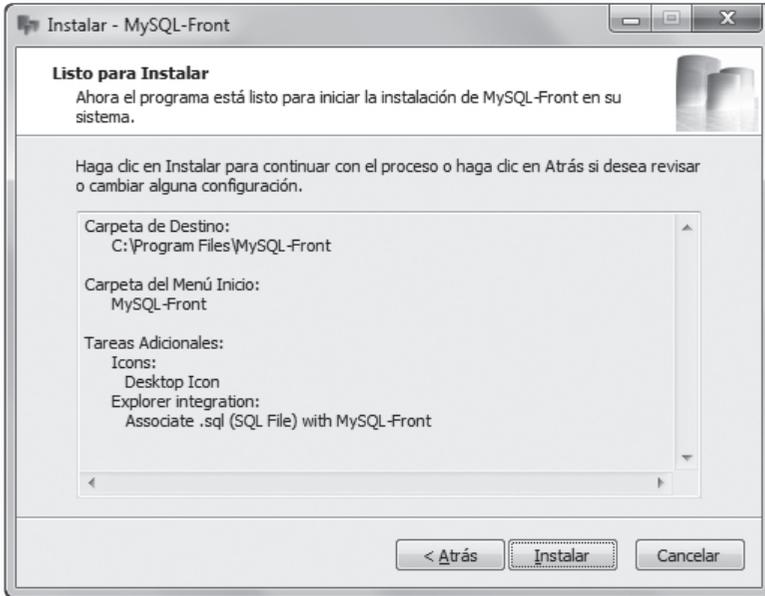
3. La instalación de MySQL-Front crea un acceso directo en el menú Inicio, no será necesario modificar, solo seleccione **Siguiente**.



4. La instalación solicita la creación de un icono en el escritorio y la asociación con archivos de extensión SQL, los cuales son activados en la siguiente ventana. Luego seleccione **Siguiente**.



5. Antes de comenzar la instalación se nos muestra una estadística de opciones seleccionadas en pasos anteriores, solo seleccione **Instalar**.



6. Una vez instalado seleccione **Finalizar**.

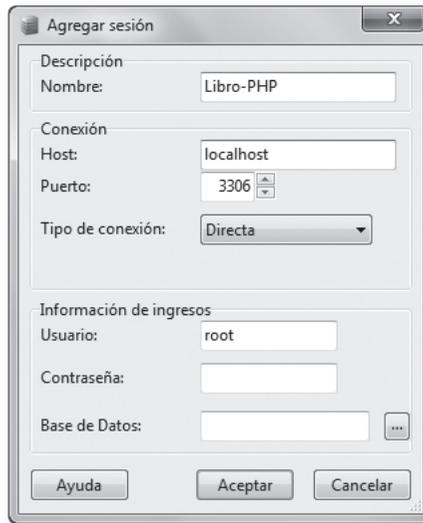


1.7.3 Crear una nueva conexión a MySQL

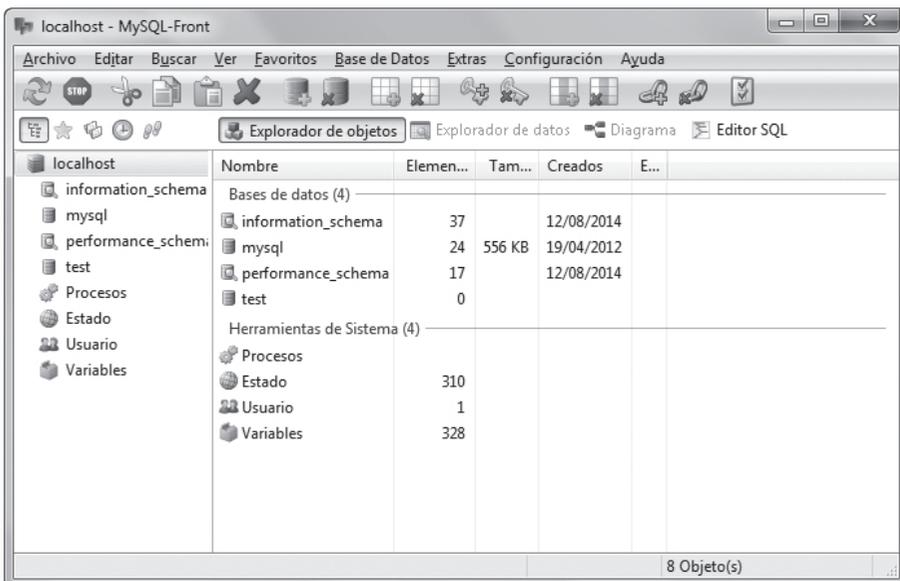
Para crear y administrar información de MySQL se debe crear una nueva conexión que permitirá tener acceso a todos los comandos MySQL.

Realice los siguientes pasos:

- Al iniciar la aplicación MySQL-Front se muestra una ventana de **Agregar sesión**. En esta ventana, en **Nombre** asigne «Libro-PHP», en la opción **Host** asigne «localhost» y asegúrese de que en **Usuario** sea «root». No será necesaria la contraseña, por lo tanto, solo seleccione **Aceptar** para iniciar la sesión.



El entorno inicial muestra las bases de datos implementadas en MySQL, así como el Explorador de objetos, Explorador de datos, Diagrama y Editor SQL. Este último permitirá ejecutar instrucciones MySQL.



1.8 Sentencias MySQL para el objeto de base de datos

Estas sentencias permiten administrar la información de modo que se puede crear y eliminar una base de datos.

1.8.1 Crear una base de datos

A. Comentario

La creación de una base de datos permite implementar una base de datos vacía, próximamente debe activarla y, luego, crear tablas y objetos necesarios.

B. Sintaxis

```
CREATE DATABASE [IF NOT EXISTS] NOMBRE_BASE_DATOS
```

- CREATE: Sentencia de definición de datos que permite crear un objeto en el servidor de MySQL.
- DATABASE: Palabra reservada que asociada a la sentencia CREATE permite crear un objeto de base de datos.
- [IF NOT EXISTS]: Cláusula que permite omitir el error que se ocasionaría si la base de datos existiera en el servidor de MySQL.
- NOMBRE_BASE_DATOS: Se especifica el nombre de la base de datos que desea implementar. Tenga en cuenta que dicho nombre no debe tener espacios en blanco ni símbolos.

Caso 1: Crear la base de datos Ventas de forma básica.

```
create database ventas;
```

Se debe tener en cuenta que si la base de datos Ventas no existe en el servidor, la sentencia se ejecuta correctamente. En caso contrario, emitirá un mensaje de error y no se continuará con la ejecución de las demás sentencias.

Caso 2: Crear la base de datos Ventas validando su existencia, de forma que valide la existencia de la misma.

```
create database IF NOT EXISTS ventas;
```

La base de datos Ventas se creará solo si no existe en el servidor.

Caso 3: Crear la base de datos Ventas validando la existencia de la misma, de forma que si la base de datos existe se deberá eliminar.

```
/* Eliminar la base de datos ventas solo si existe */  
drop database if exists ventas;  
  
/* Creando la base de datos ventas */  
create database ventas;
```

Al eliminar la base de datos en la primera sentencia, no será necesario validarla en la segunda.

1.8.2 Listar las bases de datos registradas en el servidor

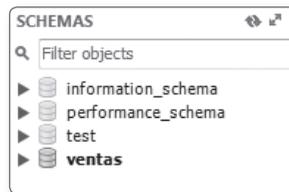
A. Comentario

Una vez creada la base de datos se puede comprobar su existencia haciendo una lista de las bases que contiene el servidor.

B. Sintaxis

```
SHOW DATABASES();
```

Listado de base de datos desde la ventana de Esquemas de MySQL Workbench.



1.8.3 Activar la base de datos

A. Comentario

Cuando se crea la base de datos en el servidor de MySQL no se activa de forma automática. Esta activación es necesaria para crear, modificar o eliminar objetos dentro de la base de datos.

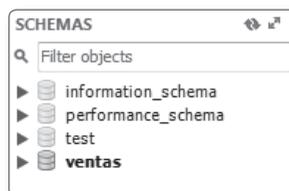
B. Sintaxis

```
USE NOMBRE_BASE_DATOS;
```

Caso 1: Activar la base de datos Ventas.

```
use ventas;
```

Mostrando en MySQL Workbench la base de datos activa, se muestra resaltada en negrita desde la lista de base de datos de la ventana SCHEMAS.



1.8.4 Mostrar la base de datos activa

A. Comentario

Cuando se crean objetos dentro de la base de datos, el servidor ejecuta las sentencias enviadas por el usuario sobre la base de datos activa, por tanto, se debe asegurar que dicha base sea la correcta.

B. Sintaxis

```
SELECT DATABASE();
```

Caso: Mostrar el nombre de la base de datos activa desde el servidor MySQL.

```
USE ventas;  
SELECT database();
```

Se debe tener en cuenta que inicialmente se debe activar una base de datos registrada en el servidor, luego se puede comprobar si se encuentra activa o no.

1.8.5 Eliminar la base de datos

A. Comentario

La eliminación de una base de datos del servidor es irreversible, es decir, no podrá recuperar la información almacenada en dicha base de datos.

B. Sintaxis

```
DROP DATABASE NOMBRE_BASE_DATOS;
```

Caso 1: Eliminar la base de datos Ventas de forma estándar.

```
DROP DATABASE ventas;
```

Caso 2: Eliminar la base de datos Ventas condicionando su existencia, es decir, se eliminará si la base de datos Ventas existe. En caso contrario, emitirá un mensaje de advertencia.

```
DROP DATABASE IF EXISTS ventas;
```

Al incluir la cláusula IF EXISTS, se verifica que la base de datos exista, y si es así lo eliminará del servidor MySQL. En caso contrario, emitirá un mensaje de advertencia. Ahora supongamos que la base de datos Ventas no existe y se desea eliminarla usando la siguiente sentencia:

```
DROP DATABASE ventas;
```

Se emitirá un error que anulará la ejecución de las sentencias, finalizando así su ejecución. Por lo tanto, es recomendable colocar la cláusula IF EXISTS en la sentencia DROP DATABASE.

1.9 Sentencias MySQL para el objeto tabla

Estas sentencias permiten administrar el objeto de tipo tabla dentro de una base de datos activa. Se debe tener en cuenta que para activar una base de datos se debe usar la sentencia USE.

1.9.1 Motor de almacenamiento

A. Comentario

El motor de almacenamiento tiene la misión de almacenar, administrar y recuperar información contenida en una tabla. Actualmente existen muchos motores que realizan las mismas tareas, la diferencia entre ellos es la velocidad y el escenario donde se ejecutará la aplicación. Entre los motores más usados tenemos MyISAM e InnoDB.

• Motor MyISAM

Es considerado como el motor por defecto y es comúnmente usado para administrar las consultas de la sentencia SELECT de MySQL.

Para poder asignar el motor MyISAM, se debe seguir el siguiente formato:

```
CREATE TABLE Nombre_de_la_tabla(  
    Columna1        tipo,  
    Columna2        tipo,  
    ...  
    ColumnaN        tipo  
) ENGINE=MyISAM;
```

También se debe considerar que cuando no se coloca la cláusula ENGINE se toma por defecto el motor MyISAM. Veamos la implementación de la tabla Cliente que cuenta con las columnas identificador de cliente, nombres, apellidos y su correo electrónico.

```
CREATE TABLE if not exists cliente (  
    idCliente        INT,  
    nombres          VARCHAR (30),  
    apellidos        VARCHAR (30),  
    email            VARCHAR (50)  
) ENGINE=MyISAM;
```

• Motor InnoDB

Se recomienda su aplicación cuando se realicen transacciones, ya que estas podrán asegurar su registro con commit y cancelar con rollback. Estas características aumentan el rendimiento para gestionar múltiples usuarios.

```
CREATE TABLE if not exists cliente (  
    idCliente        INT,  
    nombres          VARCHAR (30),  
    apellidos        VARCHAR (30),  
    email            VARCHAR (50)  
) ENGINE = InnoDB ;
```

1.9.2 Tipos de datos

Entre las partes que componen una tabla de base de datos de MySQL se presenta la especificación del tipo de datos, es decir, qué valor será soportado por dicha columna. Para tomar una decisión del tipo de datos adecuado para cada columna, se recomienda revisar la siguiente tabla:

Numéricos enteros

Bit	<p>Sus valores se encuentran entre 0 y 1.</p> <p>Objetivo: Declarar una variable de bandera que solo acepte los valores 1 y 0. Declaración: bandera bit Valor registrado: 1 o 0</p>
TinyInt	<p>Sus valores se pueden dar en rango de -128 a +127 cuando tiene signo y de 0 a 255 cuando no tiene signo.</p> <p>Objetivo: Declarar una variable para el registro de una nota de un alumno. Declaración: nota tinyint Valor registrado: 20</p>
SmallInt	<p>Sus valores se pueden dar en rango de -32768 a +32767 cuando tiene signo y de 0 a 65535 cuando no tiene signo.</p> <p>Objetivo: Declarar una variable para registrar la cantidad comprada de un determinado producto. Declaración: cantidad smallint Valor registrado: 1500</p>
MediumInt	<p>Sus valores se pueden dar en rango de -8.388.608 a +8.388.607 cuando tiene signo y de 0 a 429.4967.295 cuando no tiene signo.</p> <p>Objetivo: Declarar una variable para registrar la distancia en metros recorridos por un deportista en una carrera de 100 km. Declaración: distancia mediumint Valor registrado: 100000</p>
BigInt	<p>Sus valores se pueden dar en rango de -9.223.372.036.854.775.808 a +9.223.372.036.854.775.807 cuando tiene signo y de 0 a 18.446.744.073.709.551.615 cuando no tiene signo.</p> <p>Objetivo: Declarar una variable para registrar el recorrido de un automóvil entre la ciudad de Lima y Cusco en metros, si se sabe que tiene una distancia de 1165 km. Declaración: recorrido bigint Valor registrado: 1165000</p>
Integer	<p>Sus valores se pueden dar en rango de -2147483648 a 2147483647 cuando tiene signo y de 0 a 429.4967.295 cuando no tiene signo.</p> <p>Objetivo: Declarar una variable para registrar el total de ventas realizadas por un determinado vendedor en el transcurso de 10 años. Declaración: totalVentas integer Valor registrado: 15206265</p>

Numéricos fraccionarios

Float	<p>Es llamado número pequeño en coma flotante de precisión simple. Sus valores tienen el rango de $-3.402823466E+38$ a $-1.175494351E-38$, 0 y desde $1.175494351E-38$ a $3.402823466E+38$.</p> <p>Objetivo: Declarar una variable para registrar la talla de los trabajadores de una empresa de seguridad. Declaración: talla float Valor registrado: 1.85</p>
Double	<p>Es llamado número en coma flotante de precisión doble. Sus valores tienen un rango de $-1.7976931348623157E+308$ a $-2.2250738585072014E-308$, 0 y desde $2.2250738585072014E-308$ a $1.7976931348623157E+308$.</p> <p>Objetivo: Declarar una variable para registrar el peso en libras autorizado para camiones de carga de dos ejes si se sabe que dicho valor es 18 toneladas. Declaración: peso double Valor registrado: 41 336.6742</p>
Decimal	<p>Es llamado también número de punto flotante sin empaquetar. Esto se debe a que un valor numérico será registrado como una cadena de caracteres, es decir, un número representará un dígito. El máximo valor almacenado es el mismo que del tipo double. También debemos considerar que tiene analogía con los tipos <code>double(n,d)</code> y <code>numeric(n,d)</code>, donde n representa el total de dígitos numéricos, incluido el punto decimal y del número de decimales.</p> <p>Objetivo: Declarar una variable para registrar el sueldo mensual que recibe un trabajador si se sabe que como máximo se puede cobrar el total de \$ 17 000.50. Declaración: sueldoMensual decimal(8,2) Valor registrado: 17 000.50</p>

Cadena de caracteres

Char	<p>Permite registrar caracteres hasta un máximo de 255, asignados entre paréntesis. La longitud mínima es conocida como longitud fija. Si la cantidad asignada es inferior a su máximo límite, este se rellenará de caracteres en blanco hasta completar la especificación del campo.</p> <p>Objetivo: Declarar una variable para registrar el estado civil de un empleado. Declaración: estado_civil char(1) Posibles valores: 'S', 'C', 'V', 'D'</p>
Varchar	<p>Permite registrar una cadena de caracteres con trabajo similar al tipo char, con la diferencia que al definir un tamaño inferior al tamaño máximo no se autorrellena sino que se autoajusta al contenido.</p> <p>Objetivo: Declarar una variable para registrar el cargo de un empleado en una empresa. Declaración: cargo varchar(80) Valor registrado: 'Operario de planta'</p>
TinyText	Permite registrar una cadena de caracteres con un máximo de 255 caracteres.
Text	Permite registrar una cadena de caracteres con un máximo de 65.535.
MediumText	Permite registrar una cadena de caracteres con un máximo de 16.777.215 caracteres.
LongText	Permite registrar una cadena de caracteres con un máximo de 4.294.967.295 caracteres.