

php MyAdmin

Una de las combinaciones más utilizadas para los sitios webs dinámicos es la del lenguaje PHP y la base de datos MySQL, ambos programas OpenSource, utilizados en miles de sitios web a nivel mundial. Por ello, en esta oportunidad vamos a hablar de un software específico para el manejo de base de datos llamado phpMyAdmin.

Sitios dinámicos	14
Instalación de phpMyAdmin	16
Utilizar phpMyAdmin	21
Estructura	26
Examinar	27
SQL	28
Buscar	29
Exportar	31
Operaciones	31
Vaciar	32
Eliminar	32
Resumen	33
Actividades	34

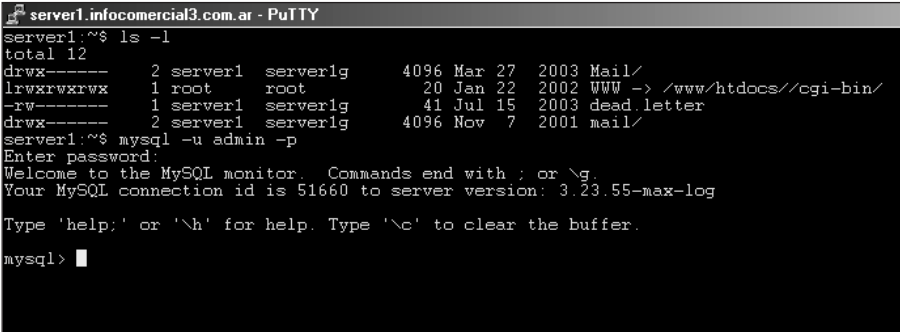
SITIOS DINÁMICOS

Es indudable que los sitios webs dinámicos, aquellos que basan gran parte de su funcionamiento gracias a la tecnología de un lenguaje de programación, por ejemplo PHP, combinado con una base de datos, han llegado para quedarse.

Cada vez vemos más y más sitios de este tipo, algunos manejados con programas OpenSource, como los que comentamos en el libro, como **Mambo**, **osCommerce**, etc. Otros con pequeños cambios dinámicos que le permiten administrar funcionalidades específicas sobre las necesidades del site, pero todos, generalmente coinciden en la utilización de una base de datos para la administración de sus contenidos, tendencia que crece día a día.

Generalmente en una aplicación web, disponemos de un servidor remoto que tiene instalado el motor de base de datos mysql y el lenguaje PHP, entre otras funcionalidades, obviamente dependiendo del plan de hosting que tengamos contratado, estas opciones varían, aunque en términos generales se da esta combinación.

La base de datos MySQL se puede administrar perfectamente conectándose al servidor en forma remota, mediante **SSH**, y desde ahí ingresar utilizando el comando MySQL como vemos en la siguiente imagen (**Figura 1**).



```

server1.infocomercial3.com.ar - PuTTY
server1:~$ ls -l
total 12
drwx----- 2 server1 server1g 4096 Mar 27 2003 Mail/
lrwxrwxrwx 1 root root 20 Jan 22 2002 WWW -> /www/htdocs/cgi-bin/
-rw----- 1 server1 server1g 41 Jul 15 2003 dead.letter
drwx----- 2 server1 server1g 4096 Nov 7 2001 mail/
server1:~$ mysql -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 51660 to server version: 3.23.55-max-log
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql>

```

Figura 1. Accediendo a una base de datos MySQL en forma remota vía SSH.



CONEXIÓN POR SSH

Si tienen habilitado en su servidor la posibilidad de conectarse por SSH y necesitan un software, puede bajarse el programa **Putty**, que es totalmente gratuito, de la siguiente dirección www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Como herramienta administrativa MySQL vía SSH, se torna bastante poco funcional, sobre todo para un usuario no experimentado, ya que como podrán observar, el manejo es similar al de una pantalla del sistema operativo **DOS** (Figura 2). Si además de considerar eso, contemplamos que no muchos proveedores de hosting, nos permiten acceder vía SSH al servidor, tendremos el problema de que no podremos ingresar a configurar nuestra base de datos.

```

server1.infocomercial3.com.ar - PuTTY
prensap
psalud_banners
psalud_foros
raz_foros
registro
roverano
salasa
socias
soldelhuapi
spaghettillearning
survey
test
universidaddelvino
users
vertigo_banners
-----
65 rows in set (0.00 sec)

mysql> use organizacionrm
Database changed
mysql> show tables;
+-----+
| Tables_in_organizacionrm |
+-----+
| z100_grupos                |
| z100_permisos              |
| z100_programas             |
| z100_usuarios              |
| z2_articulos               |
| z2_categorias              |
| z2_monedas                 |
| z2_operaciones             |
| z2_zonas                   |
+-----+
9 rows in set (0.01 sec)

mysql> desc z2_articulos;
+-----+
| Field      | Type          | Null | Key | Default        | Extra      |
+-----+
| id_articulo | int(7)         |      | PRI | NULL           | auto_increment |
| nombre      | varchar(30)    |      |     |                 |
| desc_basica | varchar(255)   |      |     |                 |
| desc_detalada | text          |      |     |                 |
| cantidad    | int(5)         |      |     | 0               |
| precio      | varchar(10)    |      |     |                 |
| fecha_carga | datetime       |      | MUL | 0000-00-00 00:00:00 |
| cod_operacion | int(3)        |      |     | 0               |
| cod_categoria | int(3)        |      | MUL | 0               |
| cod_zona     | int(3)        |      | MUL | 0               |
| cod_moneda   | int(3)        |      | MUL | 0               |
+-----+
11 rows in set (0.01 sec)

```

Figura 2. Como vemos aquí, ver la estructura de una tabla no es complejo, pero poco funcional si debemos realizar muchos trabajos con la base de datos.

Para solucionar este tema, se comenzaron a desarrollar distintas aplicaciones web, que nos permiten manejar desde el navegador web nuestras bases de datos. Entre estas aplicaciones que se desarrollaron, tenemos a phpMyAdmin, que fue creada por la misma persona que creó **phpAds**, Tobias Ratschiller.

phpMyAdmin, no sólo es un programa OpenSource, sino que es “el programa” con mayúsculas para la administración de bases de datos MySQL en forma remota. No solamente por las funcionalidades que nos ofrece y que van mejorando con

cada nueva versión, sino también porque a lo largo de los años ha sabido ganarse su lugar, y estar presente en casi todos los proveedores de hosting a nivel mundial, además de haber ganado numerosos premios.

También debemos considerar que para administrar una base de datos, necesitaremos saber algo de **SQL** (*Structured Query Language*), que es el lenguaje que se utiliza para realizar una consulta a la base de datos. Por ejemplo cuando utilizamos un buscador, basado en el texto que hayamos introducido, se arma la consulta, que es enviada a la base de datos, y ejecutada. Gracias a phpMyAdmin esto es bastante más sencillo en algunos aspectos, porque independientemente de conocer o no mucho sobre SQL, el manejo es bastante intuitivo y nos permite movernos con relativa soltura.

Otro aspecto a destacar es que la versión de phpMyAdmin está disponible en 47 lenguajes diferentes, y se puede cambiar entre los distintos lenguajes disponibles, simplemente seleccionando el lenguaje de nuestro interés desde una lista desplegable, lo cual la hace sumamente adaptable a muchos usuarios y países.

En las próximas páginas veremos la instalación de phpMyAdmin, que servirá en el caso de que no la tengan instalada en su proveedor de hosting, o si se están creando un entorno de desarrollo local en su PC, por ejemplo con un servidor web como apache, con PHP y con MySQL.

INSTALACIÓN DE PHPMYADMIN

Los requerimientos para poder instalar phpMyAdmin son, en primer caso, tener instalado el motor de base de datos MySQL (lo pueden obtener de www.mysql.com), y les sugiero la versión 3.23 en adelante.

Otro requerimiento es que dispongan del lenguaje php, en la versión 4.1 o superior, y es recomendable que dispongan de la librería **GD2** para la previsualización de imágenes **JPEG**. Si no la tienen disponible, la pueden obtener en www.boutell.com/gd o pedirle a su proveedor de hosting que la instale.



ASIGNAR PERMISOS

phpMyAdmin no nos provee de ningún método de seguridad adicional para el servidor MySQL, por lo que, como administradores, deberemos asegurarnos de que los permisos estén asignados correctamente para evitar posibles problemas.

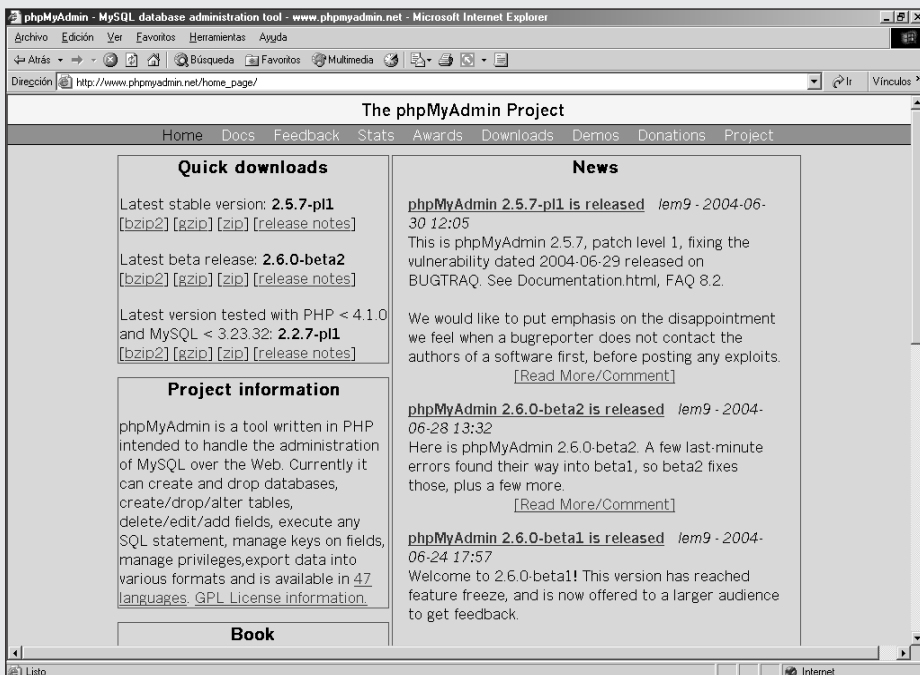
Importante: pueden utilizar phpMyAdmin con versiones menores de PHP y de MySQL. De todos modos, verifique cuáles son las últimas versiones testeadas disponibles en el sitio de phpMyAdmin.

El sitio web oficial del proyecto phpMyAdmin es **www.phpmyadmin.net** y lo primero que tiene que hacer luego de haber cumplido con todos los requerimientos necesarios, es obtener la última versión estable del proyecto, que al momento de escribir este libro era la 2.5.7-pl1.

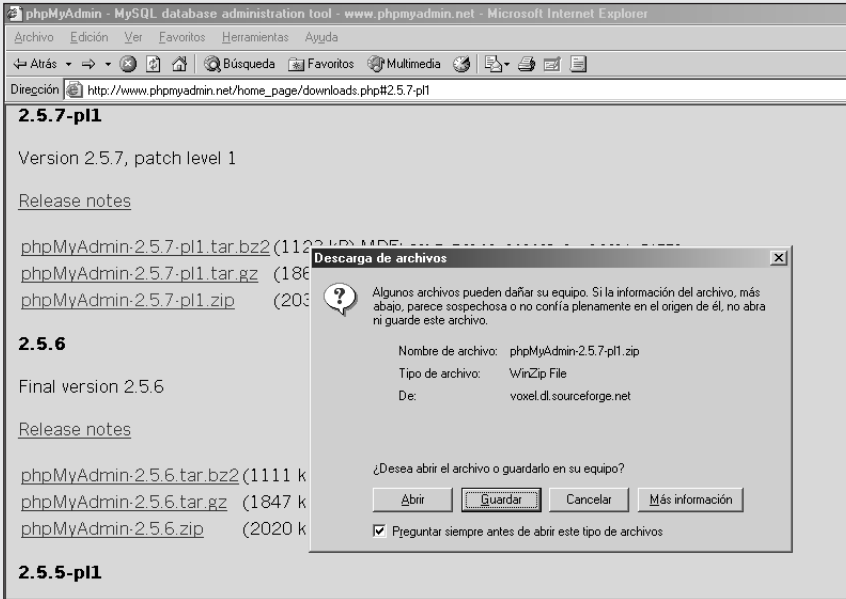
■ Obtener phpMyAdmin

PASO A PASO

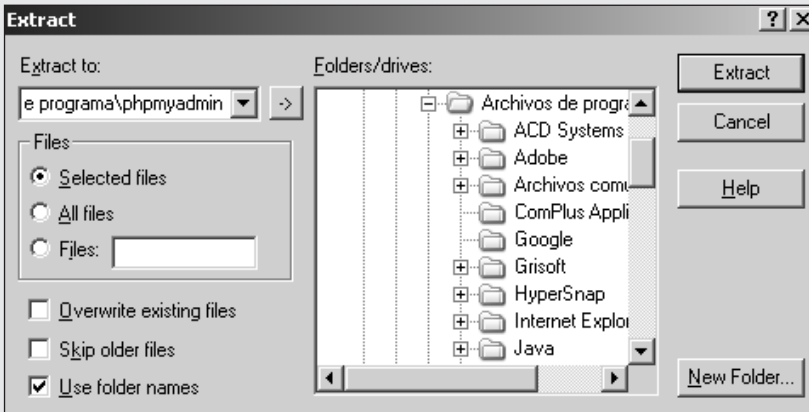
- 1 Ingrese al sitio oficial de phpMyAdmin y seleccione la opción de **downloads** en **www.phpmyadmin.net/home_page/downloads.php** y luego de abrir esta pantalla seleccione la última versión estable del programa, que seguramente será el primer link que encuentre. Note que también encontrará en **Quick downloads** la última versión beta del programa y la última versión testada con versiones anteriores de PHP y MySQL.



- 2 Ahí verá un listado de archivos, que en definitiva contienen lo mismo, pero se diferencian por cómo están comprimidos. Debe seleccionar uno para bajar a su equipo, por ejemplo el **.ZIP** como se ve en la siguiente imagen:



- 3 Luego de haber obtenido la copia del programa (generalmente la habrá bajado en su PC), deberá proceder a descomprimirlo en un directorio específico.



Después de ello, sólo le resta conectarse vía FTP (**Figura 3**) a su servidor para copiar todos los archivos del programa al directorio que más le guste, por ejemplo phpMyAdmin, o si la instalación es en un equipo local, copiarlos dentro de un directorio accesible desde el servidor web, por ejemplo si tiene Apache en **C:\Archivos de programa\Apache Group\Apache2\htdocs** aunque depende de cómo tenga configurado Apache esta dirección puede cambiar.

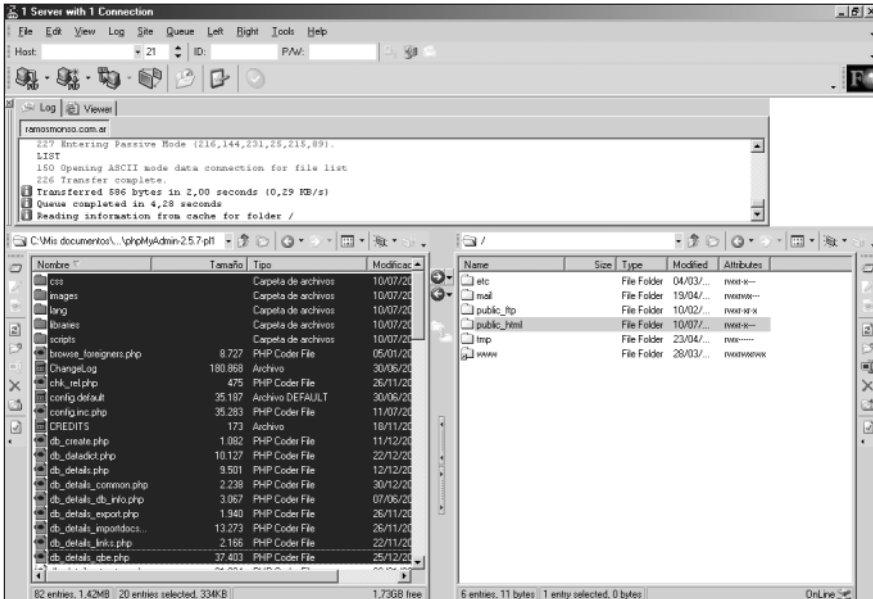


Figura 3. Subiendo el contenido de phpMyAdmin a su servidor utilizando un programa de FTP gratuito.

Luego de haber copiado todos los archivos al directorio de destino, debe abrir el archivo **config.inc.php** que es el que controla todas las variables de configuración del programa para poder comenzar a personalizarlo. Para abrir ese archivo puede hacerlo con el **Notepad** o cualquier editor de texto (**Figura 4**).

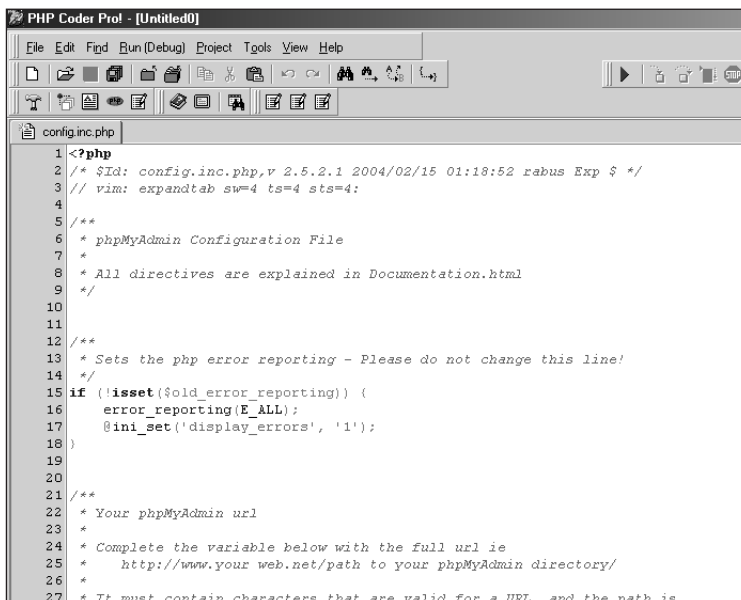


Figura 4. Editando el archivo de configuración de phpMyAdmin con el editor phpCoder.

■ Editar el archivo de configuración

PASO A PASO

- 1 Abra el archivo **config.inc.php** y busque la línea que contenga la siguiente variable: **\$cfg ['PmaAbsoluteUri'] = " ;**
- 2 Luego deberá indicar la ruta absoluta donde reside phpMyAdmin, que podría ser algo como **http://localhost/phpMyAdmin** o **http://www.midominio.com/phpMyAdmin**, y le debería quedar algo así:
\$cfg ['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin'
- 3 Luego debe buscar la variable: **\$cfg['Servers'][\$i]['auth_type']**. Esta variable viene configurada por defecto con el valor **config**, y la misma define el tipo de autenticación que desea que utilice el programa para validar a los usuarios. Aquí está bien que diga **config**, aunque no se lo recomiendo porque constituye una vulnerabilidad en la seguridad, por ello le sugiero utilizar **http** para que le solicite el usuario y contraseña de acceso vía **HTTP** o **cookie** para autenticar con cookies.
- 4 Después busque **\$cfg['Servers'][\$i]['user']** y **\$cfg['Servers'][\$i]['password']**. Estas variables le permiten especificar el usuario y contraseña para conectarse a la base de datos MySQL, por ejemplo si sólo los administradores de la base de datos, su usuario podría ser **'root'**. Si elige como método de autenticación **config**, debe colocar la contraseña si por el contrario elige otro como **http** o **cookie**, no será necesario.

Hay otra variable que puede resultarle de utilidad, en caso de que tenga instalado MySQL en un puerto diferente al normal. Si este es su caso, debe buscar la variable **\$cfg['Servers'][\$i]['port']** y especificar el puerto correspondiente de su instalación. También, mediante la variable **\$cfg['Servers'][\$i]['only_db']** puede especificar si quiere que phpMyAdmin le muestre sólo los datos de una determinada base de datos. Esta opción es útil si quiere ofrecer acceso a distintos usuarios para que cada uno vea solamente sus bases.

III COMANDO TAR

Si dispone de acceso SSH en su servidor, podrá subir el archivo comprimido (usualmente archivo **.TAR.GZ**) a la dirección definitiva y descomprimirlo utilizando el comando **tar**.

★ CUIDADO CON LA RUTA

Al editar el archivo de configuración de phpMyAdmin recuerde **no** terminar la especificación de la ruta con una barra al final (/).

Además podemos buscar la variable `$cfg['blowfish_secret'] = ''`; en el caso de que hayamos elegido como método de autentificación **cookie**, y deseamos que la contraseña esté encriptada, para lo cual le colocamos como valor a esta variable una palabra que se nos ocurra, la que será utilizada por **Blowfish** para encriptar la contraseña. Hasta aquí es todo lo necesario que debemos hacer generalmente para poder configurar phpMyAdmin, al menos para empezar, ya que al continuar recorriendo el archivo de configuración podrán encontrar muchísimas otras opciones que les permitirán personalizar phpMyAdmin a medida de sus necesidades.

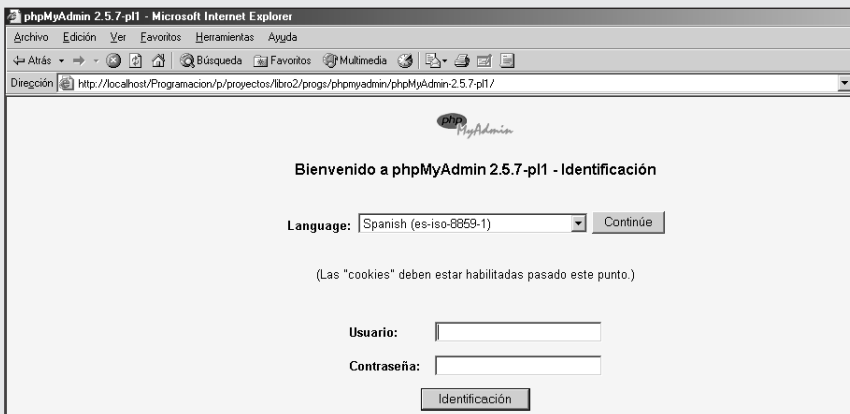
UTILIZAR PHPMYADMIN

Después de haber configurado el archivo **config.inc.php** sólo debemos ingresar al directorio donde hemos instalado phpMyAdmin para empezar a utilizar el programa. Por ejemplo: **www.misitioweb.com/phpMyAdmin/**.

■ Conectarse a la base de datos

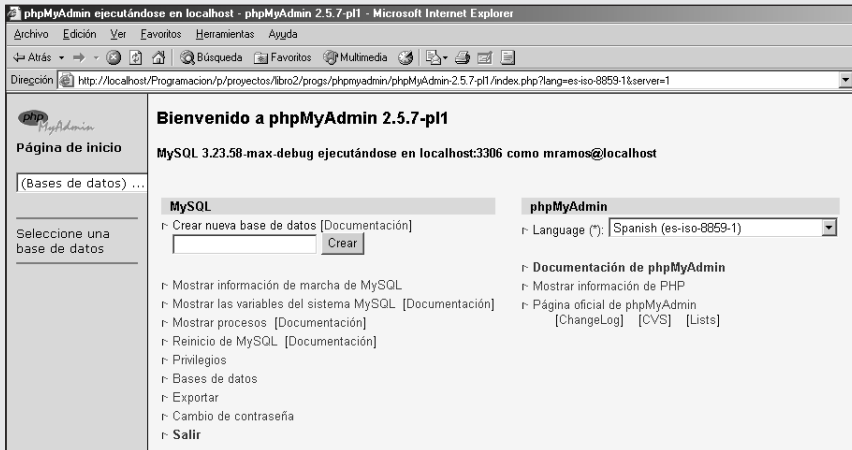
PASO A PASO

- 1 Una vez que ingrese a phpMyAdmin, el sistema le pedirá el usuario y contraseña, luego de lo cual podrá ingresar a la pantalla principal de administración.

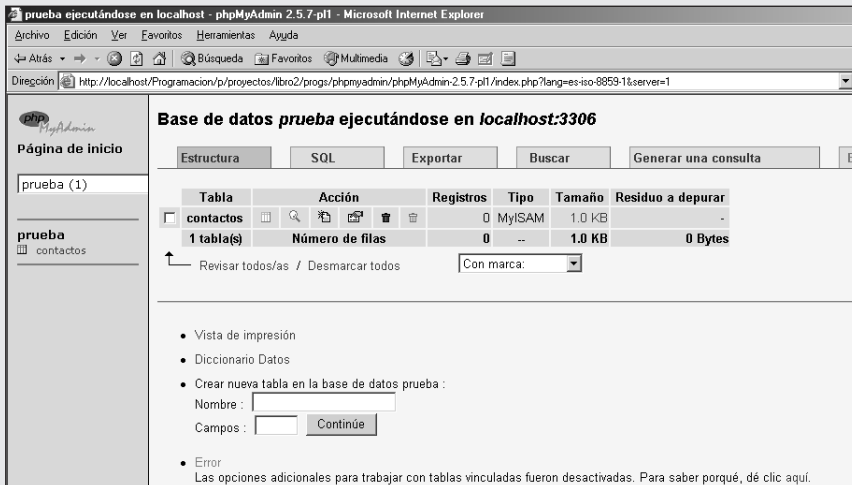


- 2 Si cometió algún error al definir las variables en el archivo **config.inc.php**, le aparecerá el mensaje respectivo. Recuerde no dejar como método de autentificación el valor **config**, ya que constituye un potencial problema de seguridad. De acuerdo con los permisos que tenga sobre las bases de datos, podrá acceder a distintas opciones, ya que por ejemplo si ingresa como usuario **root**

porque es el administrador del servidor de base de datos, podrá realizar todas las operaciones posibles, como dar de alta usuarios, reiniciar la base de datos, etc, y si sólo tiene permisos para algunas bases, con restricciones las opciones serán menores.



3 La pantalla principal de phpMyAdmin, está compuesta por una página de marcos, dividida en dos partes: por un lado el lateral izquierdo donde podrá observar todas las bases de datos a las cuales tiene acceso y por el otro, el lateral derecho donde le mostrará en principio todas las opciones a las cuales puede acceder según sus privilegios de usuario, y posteriormente, luego de haber seleccionado una base de datos desde el marco izquierdo, podrá ver las propiedades de la misma, como cantidad de tablas, registros, etc.

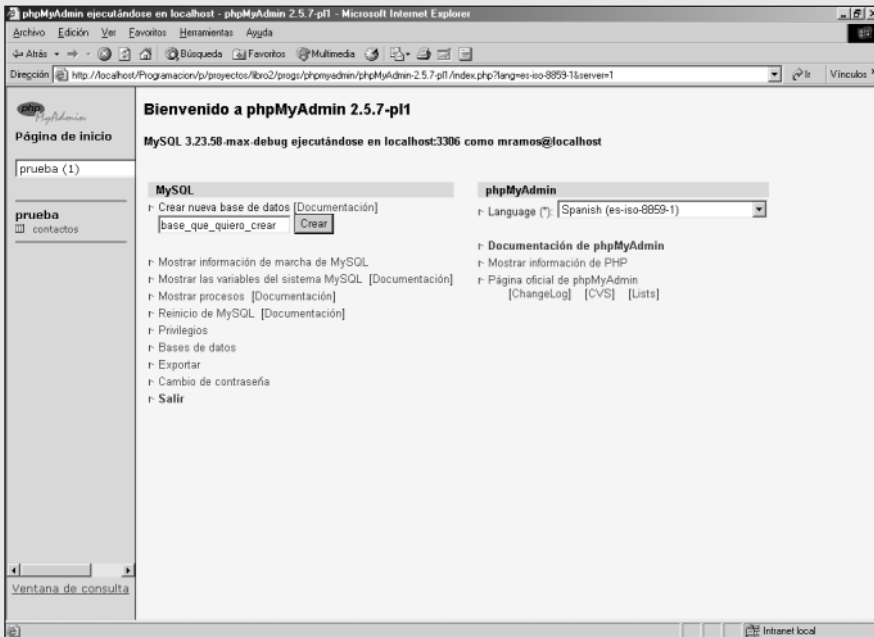


A continuación veremos las principales operaciones que podemos realizar con phpMyAdmin, ya que muchas de ellas son bastante intuitivas. Antes de ello, vale la pena aclarar, aunque tal vez muchos de ustedes ya lo sepan, que las bases de datos contienen dentro de ellas, tablas, y las tablas contienen registros, que es en definitiva el lugar donde queda contenido el dato.

■ Crear una base de datos

PASO A PASO

- 1 Para crear una base de datos, simplemente vamos a la página principal del sistema y completamos el formulario donde podremos indicar el nombre de la base de datos que queremos crear, en el ítem **Crear nueva base de datos**. Luego de colocar el nombre de la base de datos que deseamos crear, debemos hacer clic sobre el botón **Crear**.



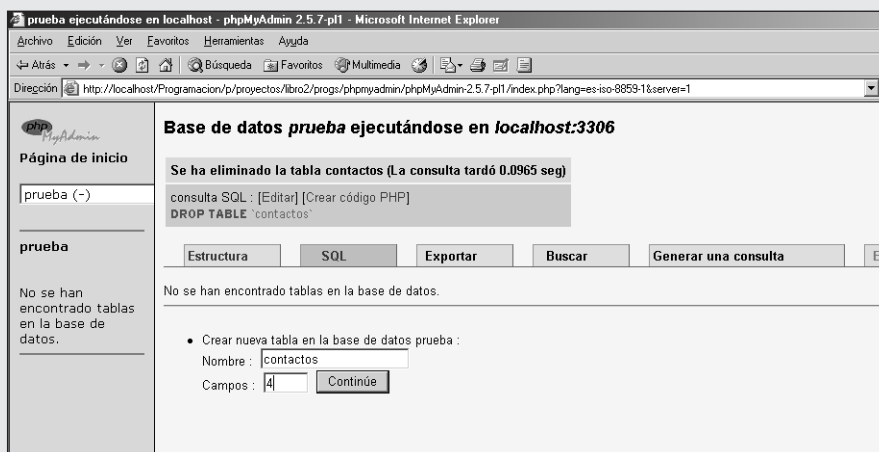
- 2 Después de haber creado la nueva base de datos, podremos observar que el sistema nos muestra un mensaje en caso de haber creado la base de datos satisfactoriamente, y además actualiza el marco izquierdo incluyendo en el listado de bases de datos disponibles la que acabamos de crear.

Una vez creada la base de datos debemos proceder a crear las tablas. Para eso, completamos los pasos descritos a continuación:

■ Crear tablas

PASO A PASO

- 1 Seleccionar desde el marco izquierdo la base de datos sobre la cual desea crear la tabla, haciendo clic sobre el nombre, luego de lo cual le aparecerá sobre el marco derecho un formulario para completar con el nombre de la tabla a crear y la cantidad de campos. Tenga en cuenta que si no conoce la cantidad de campos exactos que llevará la tabla, podrá luego ingresar campos adicionales sin problemas.
- 2 Si quisiera crear una tabla para almacenar los datos de contacto de una agenda, podría crear la tabla **contactos**, con 4 campos que serían el **id** del contacto, el **nombre** del contacto, el **teléfono** y la **dirección de email**, por lo que en el formulario debería colocar en **Nombre: contactos** y en **Campos: 4**. Después haga clic sobre el boton **Continúe**.



- 3 Luego de haber creado la tabla con la cantidad de campos, le aparecerá un formulario para poder elegir los valores correspondientes a cada uno de los campos, para saber el tipo de datos que guardará, si es la clave primaria, un campo

★ COMPLETAR LOS CAMPOS

{ } ¿SOFTWARE LIBRE?

Tengan en cuenta que hasta no haber completado todos los datos de los campos, la tabla no será creada definitivamente.

Si tienen alguna duda sobre la venta del software libre y si debemos cobrar por él, visite www.gnu.org/philosophy/selling.es.html.

autonumérico, entre otras muchas opciones. Cada fila corresponderá a un campo en la tabla. Luego de haber completado todos los datos simplemente haga clic sobre el botón **Grabar**.

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Campo	Tipo [Documentación]	Longitud/Valores*	Atributos	Nulo	Predeterminado**	Extra	Primaria
id_contacto	INT	6		not null			<input checked="" type="radio"/>
nombre	VARCHAR	50		not null			<input type="radio"/>
telefono	VARCHAR	50		not null			<input type="radio"/>
e_mail	VARCHAR	50		null			<input type="radio"/>

Comentarios de la tabla :

Tipo de tabla :

* Si el tipo de campo es "enum" o "set", por favor ingrese los valores usando este formato: 'a','b','c'...
Si alguna vez necesita poner una barra invertida("\") o una comilla simple (") entre esos valores, siempre ponga una barra invertida. (Por ejemplo "\xyz" o 'a\b').

** Para valores predeterminados, por favor ingrese solamente un valor, sin "escaping backslash" ni comillas, usando este formato: a

Con la tabla del ejemplo, podemos ver, luego de hacer clic sobre el botón **Grabar**, que el programa nos muestra el mensaje de que la tabla ha sido generada y también podremos visualizar el código SQL generado para crearla (**Figura 5**).

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Tabla contactos se creó.

consulta SQL : [Editar] [Crear código PHP]

```
CREATE TABLE `contactos` (
  `id_contacto` INT(6) NOT NULL ,
  `nombre` VARCHAR(50) NOT NULL ,
  `telefono` VARCHAR(50) NOT NULL ,
  `e_mail` VARCHAR(50) ,
  PRIMARY KEY ( `id_contacto` )
);
```

Estructura Examinar SQL Buscar Insertar Exportar Operaciones Vaciar Eliminar

Campo	Tipo	Atributos	Nulo	Predeterminado	Extra	Acción
<input checked="" type="checkbox"/> id_contacto	int(6)		No	0		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> nombre	varchar(50)		No			<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> telefono	varchar(50)		No			<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
<input type="checkbox"/> e_mail	varchar(50)		Sí	NULL		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

↶ Revisar todos/as / Desmarcar todos Con marca: ☒ ☐

Índices : [Documentación]

Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	0	<input checked="" type="checkbox"/> <input type="checkbox"/>	id_contacto

Crear un índice en columnas

Espacio utilizado :

Tipo	Uso
Datos	0 Bytes
Índice	1,024 Bytes
Total	1,024 Bytes

Estadísticas de la fila :

Enunciado	Valor
Formato	dinámico/a
Filas	0
Creación	12-07-2004 a las 15:59:20

Figura 5. La tabla contactos ya ha sido creada exitosamente.

Si no dispusiéramos de esta herramienta, y tuviéramos que crear la base de datos desde MySQL, directamente deberíamos crear todo el SQL a mano para crear la tabla, como vemos a continuación:

```
CREATE TABLE `contactos` (
  `id_contacto` INT AUTO_INCREMENT ,
  `nombre` VARCHAR( 50 ) NOT NULL ,
  `telefono` VARCHAR( 50 ) NOT NULL ,
  `e_mail` VARCHAR( 50 ) NOT NULL ,
  PRIMARY KEY ( `id_contacto` )
);
```

En este ejemplo, esto no sería nada complejo, porque es algo bastante corto, pero si disponemos de una tabla más grande, las cosas se empiezan a complicar a veces por un simple error de sintaxis, pero gracias a estos administradores podemos hacerlo mucho más fácilmente.

Luego de haber creado la tabla **contactos**, podremos observar que sobre el marco derecho, nos aparece bajo el nombre de la base de datos que estamos utilizando, el listado de tablas que contiene. Desde ahora podremos hacer clic sobre la tabla, para poder ver las propiedades de la misma.








Al hacer clic sobre la tabla nos aparece sobre el marco derecho una nueva pantalla con información general sobre la tabla como cantidad de registros, índices, espacio utilizado y estadísticas de la misma, además de la estructura de la tabla, con los campos que la conforman. También nos muestra un menú con diferentes opciones, las cuales veremos a continuación.


Estructura

Al hacer clic sobre esta opción (**Figura 6**), podremos acceder a la misma pantalla que la anterior, la cual principalmente nos muestra el listado de campos que componen la tabla, y las propiedades de los mismos, además de 6 iconos sobre la columna **Acción**, los cuales nos permitirán realizar diversas acciones sobre el campo seleccionado, como ser editar el campo para cambiar sus propiedades, eliminarlo de la tabla (tengan en cuenta que perderán todos los datos del mismo), definirlo como clave primaria, crear un índice, definirlo como único, y por último sobre los campos de texto, poder definirlo para utilizar la opción *Full Text Search* que es una propiedad de las bases de datos MySQL a partir de la versión 3.23, que realmente les recomiendo investiguen si les gusta desarrollar.


Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar SQL Buscar Insertar Exportar Operaciones Vaciar Eliminar

Campo	Tipo	Atributos	Nulo	Predeterminado	Extra	Acción
<input type="checkbox"/> id_contacto	int(5)		No	0		     
<input type="checkbox"/> nombre	varchar(50)		No			     
<input type="checkbox"/> telefono	varchar(50)		No			     
<input type="checkbox"/> e_mail	varchar(50)		Sí	NULL		     

Revisar todos/as / Desmarcar todos Con marca:  

Índices : [Documentación]

Nombre de la clave	Tipo	Cardinalidad	Acción	Campo
PRIMARY	PRIMARY	0	 	id_contacto

Crear un índice en 1 columnas

Espacio utilizado :

Tipo	Uso
Datos	0 Bytes
Índice	1,024 Bytes
Total	1,024 Bytes

Estadísticas de la fila :

Enunciado	Valor
Formato	dinámico/a
Filas	0
Creación	12-07-2004 a las 15:59:20
Última actualización	12-07-2004 a las 15:59:22

- Vista de impresión
- Insertar nuevo campo : 1 Al final de la tabla
- Planteamiento de una estructura de tabla [Documentación]

Ejecute la/s consulta/s SQL en la base de datos prueba [Documentación]

Campos:



☐ id_contacto
☐ nombre
☐ telefono

Figura 6. Visualizando la estructura de la tabla contactos.

Desde aquí también podremos incorporar nuevos campos a la tabla, completando el formulario que aparece en la sección **Insertar nuevo campo**.

Examinar

Si hacemos clic sobre el botón **Examinar** (Figura 7), podremos ver el listado de registros que están contenidos en nuestra base de datos (en nuestro proyecto, **contactos**), y además nos muestra el SQL utilizado para generar la consulta. Esta opción también nos permite ordenar los registros de diversas formas (horizontal, vertical, etc), definir la cantidad de registros que deseamos ver, organizarlos según una determinada clave, entre otras actividades. Además por cada registro tendremos de dos iconos para poder editarlo o eliminarlo.



MATAR EL PROCESO

Puede suceder que tengan algún usuario que esté ejecutando una consulta bastante pesada, o mal generada, que tome más recursos de los necesarios o ralentice el sistema demasiado. Si ese es el caso, podemos matar el proceso desde la opción **Mostrar procesos/Matar proceso** con lo cual podremos normalizar el sistema.

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar **SQL** Buscar Insertar Exportar Operaciones Vaciar Eliminar

Mostrando registros 0 - 1 (2 total, La consulta tardó 0.1029 seg)

consulta SQL : [Editar] [Explicar el SQL] [Crear código PHP]
`SELECT *
FROM `contactos` LIMIT 0 , 30`

Mostrar : 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

Organizar según la clave: Ninguna Continúe

	id_contacto	nombre	telefono	e_mail
<input type="checkbox"/>	0	Martin Ramos	1111-1111	mramos@infocomercial.com
<input type="checkbox"/>	1	Carlos Perez	2222-2222	cperez@misitio.com

Con marca: ☒

Mostrar : 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

Figura 7. Visualizando todos los registros contenidos en la tabla contactos.

SQL

En la sección **SQL** (**Figura 8**), podremos ver en la pantalla un formulario donde podremos especificar nuestra consulta SQL. Por ejemplo si ya hemos ingresado algunos registros en la tabla contacto, podremos consultar los mismos con la siguiente consulta que deberemos colocar en el formulario: **SELECT * FROM "contactos"** Esta consulta ya viene predeterminada cada vez que ingresamos al formulario, aunque podremos ejecutar cualquier tipo de consulta SQL, lo único que deberemos aprender un poco, en caso de no saber, es el lenguaje SQL.

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar **SQL** Buscar Insertar Exportar Operaciones Vaciar Eliminar

Ejecute la/s consulta/s SQL en la base de datos prueba [Documentación]

SELECT * FROM `contactos` WHERE 1

Campos:
id_contacto
nombre
telefono
e_mail

Insertar

☒ Mostrar esta consulta otra vez
O Localización del archivo de texto :
Examinar... (Tamaño máximo: 2,048KB)

Compresión: ☒ Autodetecte ☐ Ninguna ☐ "Comprimido con gzip"

Continúe

Figura 8. Ingresando una sentencia sql directamente.

Además podremos observar que a la derecha del formulario disponemos de una lista con los campos de la tabla, desde donde podremos insertar los campos si no queremos escribirlos uno por uno.

Por otro lado, también disponemos en el formulario de consulta de la opción de poder enviar un archivo que contenga un SQL, por ejemplo, podemos tener un archivo de texto que contenga la consulta **SELECT * FROM contactos**; poder incluirlo en el formulario haciendo clic sobre el botón **Examinar**, y seleccionando el archivo desde nuestro equipo local. También podremos aquí especificar si el archivo está comprimido o no, aunque dispone de una opción de autodetección que funciona muy bien. La posibilidad de incluir el contenido de un archivo es sumamente útil, cuando por ejemplo instalamos algún programa, donde debemos crear la estructura de la base de datos por separado. También es usado para recuperar un backup, etc.

Buscar

Desde la opción **Buscar** (Figura 9) podremos acceder a un formulario que nos permitirá seleccionar las distintas preferencias para nuestra búsqueda, como ser la elección de los campos que deseamos mostrar como resultados al ejecutar la búsqueda (al menos debemos seleccionar un campo), y luego poder definir el criterio de búsqueda en cada campo disponible en la tabla, desde donde podremos elegir, de una lista desplegable las opciones: igual, representada por el signo **=**, la opción distinto, representada por el símbolo **!=** y la opción **LIKE** que nos permite buscar sobre el contenido de un campo en forma parcial para ver si contiene determinada palabra (Figura 10).

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar SQL **Buscar** Insertar Exportar Operaciones Vaciar Eliminar

Seleccionar campos (al menos uno):

id_contacto
nombre
telefono
e_mail

• registros por página 30

• Insertar las condiciones de búsqueda (cuerpo de la cláusula "where"):

[] [Documentación]

☐ Hacer una consulta (comodin: "%")

Campo	Tipo	Función	Valor
id_contacto	int(6)	=	
nombre	varchar(50)	LIKE	%martin%
telefono	varchar(50)	LIKE	
e_mail	varchar(50)	LIKE	

• Mostrar en este orden:

[] ☒ Ascendente ☐ Descendente

Continúe

Figura 9. Buscando un registro que contenga la palabra *martín* en el campo *nombre*.

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar SQL Buscar Insertar Exportar Operaciones Vaciar Eliminar

Mostrando registros 0 - 0 (1 total, La consulta tardó 0.0497 seg)

consulta SQL : [Editar] [Explicar el SQL] [Crear código PHP]
SELECT "
FROM 'contactos'
WHERE 1 AND 'nombre'
LIKE '%martin%' LIMIT 0 , 30

Mostrar : 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

←→	id_contacto	nombre	telefono	e_mail
<input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	0	Martin Ramos	1111-1111	mramos@infocomercial.com

Con marca: ☒ ☐

Mostrar : 30 filas empezando de 0
en modo horizontal y repite encabezados cada 100 celdas

Insertar nueva fila
Vista de impresión
Previsualización para imprimir (documento completo)

Figura 10. Obteniendo los resultados de nuestra búsqueda de **martín**

Insertar

En la opción **Insertar** (Figura 11), podremos ver un formulario con todos los campos de la tabla, desde donde podremos insertar nuevos registros a la tabla. Tengan en cuenta que lo ideal en una aplicación web no es insertar los registros desde aquí, sino hacerlo desde un formulario específico en cada aplicación, ya que por ejemplo, si definimos en nuestra tabla un campo autonumérico, normalmente no le damos al usuario la opción de que el mismo pueda completar el valor.

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar SQL Buscar Insertar Exportar Operaciones Vaciar Eliminar

Campo	Tipo	Función	Nulo	Valor
id_contacto	int(6)		<input type="checkbox"/>	2
nombre	varchar(50)		<input type="checkbox"/>	Marcela Lopez
telefono	varchar(50)		<input type="checkbox"/>	4444-3332
e_mail	varchar(50)		<input type="checkbox"/>	mlopez@otrositio.com

Insertar como una nueva fila -- y luego --

☒ Volver
☐ Insertar un nuevo registro

Continúe Reset

Figura 11. Ingresando un nuevo contacto en la tabla contactos.

Exportar

Desde aquí podremos volcar el contenido de una tabla, es decir que el sistema nos muestre el contenido en formato de consulta SQL (u otros, depende de nuestra selección en el formulario) por ejemplo para usarlo como copia de seguridad de la misma, o para ser usado en otra base de datos. Las opciones disponibles son varias, entre ellas la de exportar sólo la **Estructura** (no los datos), la estructura y los datos, **Añadir "drop table"**, es decir que en la consulta lo primero que hará es borrar la tabla existente, entre muchas otras (**Figura 12**).

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar SQL Buscar Insertar **Exportar** Operaciones Vaciar Eliminar

Mostrar el volcado esquema de la tabla

Exportar

- ☒ SQL
- ☐ LaTeX
- ☐ CSV para datos de MS Excel
- ☐ Datos CSV
- ☐ XML

Opciones SQL (Documentación)

☒ Estructura

- ☐ Añadir 'drop table'
- ☒ Añada el valor AUTO_INCREMENT
- ☒ Usar "backquotes" con tablas y nombres de campo

Añada en los comentarios

☐ Fechas de creación/actualización/revisión

☒ Datos

- ☐ Completar los "Inserts"
- ☐ "Inserts" extendidos
- ☐ Use "inserts" con retraso

Tipo de exportación:

Volcar filas empezando por la fila .

☐ Enviar (genera un archivo descargable)

Plantilla del nombre del archivo: (☒ recuerde la plantilla)*

Figura 12. Exportando la tabla contactos.

Operaciones

Bajo la sección **Operaciones** (**Figura 13**), podremos realizar distintas tareas sobre la tabla, como ser cambiar el orden, renombrarla, moverla o copiarla a otra base de da-

III BUSCAR EN LA BASE

Para hacer una búsqueda con el operador **LIKE**, pueden usar como comodines el carácter % (por ciento), el cual podrá ser incluido al principio, al final o en ambas posiciones, por ejemplo al buscar **%martín%** podrá encontrar esa cadena de texto en cualquier posición del campo.

tos, ejecutar tareas de mantenimiento y optimización sobre la tabla, por ejemplo luego de haber borrado muchos registros de la misma, entre otras.

Base de datos prueba - Tabla contactos ejecutándose en localhost:3306

Estructura Examinar SQL Buscar Insertar Exportar Operaciones Vaciar Eliminar

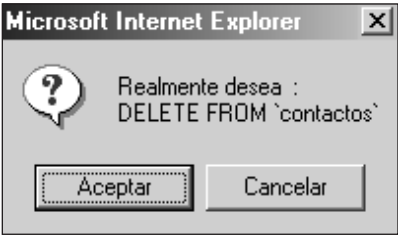
- Modificar el "Order By" de la tabla : Continúe (solo)
- Cambiar el nombre de la tabla a : Continúe
- Mover tabla a (Base de datos.tabla):
 . Continúe
- Copiar la tabla a (base de datos.tabla):
 .
☒ Únicamente la estructura
☐ Estructura y datos
☐ Solamente datos
☐ Añadir 'drop table'
☐ Añada el valor AUTO_INCREMENT
☐ Cambie (switch) a la tabla copiada
- Mantenimiento de la tabla
 - Revisar tabla [Documentación]
 - Analizar la tabla [Documentación]
 - Reparar la tabla [Documentación]
 - Optimizar la tabla [Documentación]
 - Vaciar el caché de la tabla ("FLUSH") [Documentación]
- Comentarios de la tabla : Continúe

Figura 13. Mantenimiento sobre la tabla.

Vaciar

Al hacer clic sobre la opción **Vaciar**, podremos eliminar los registros de la tabla, no eliminar la estructura, sólo los registros (**Figura 14**).

Figura 14. Al vaciar una tabla el sistema nos pedirá una confirmación.



Eliminar

Desde aquí eliminamos todos los registros de la tabla y la estructura de la misma (**Figura 15**).

Figura 15. Eliminando la tabla de nuestro servidor.



También a medida que vayamos creando nuevas tablas, podremos ver la información resumida de toda la base de datos, como ser la cantidad de registros que tiene cada tabla, los registros totales en la base de datos, el tamaño de cada tabla, etc. A su vez, en la parte superior de la pantalla podremos encontrar un menú con distintas opciones como mencionamos anteriormente, las cuales nos permitirán realizar diversas operaciones sobre la base de datos. Por ejemplo, la opción **Exportar** nos permite volcar el contenido ya no de una tabla en particular, sino de varias o todas las contenidas en la base. De esta manera, podremos hacer un backup completo sobre la base de datos (**Figura 16**).

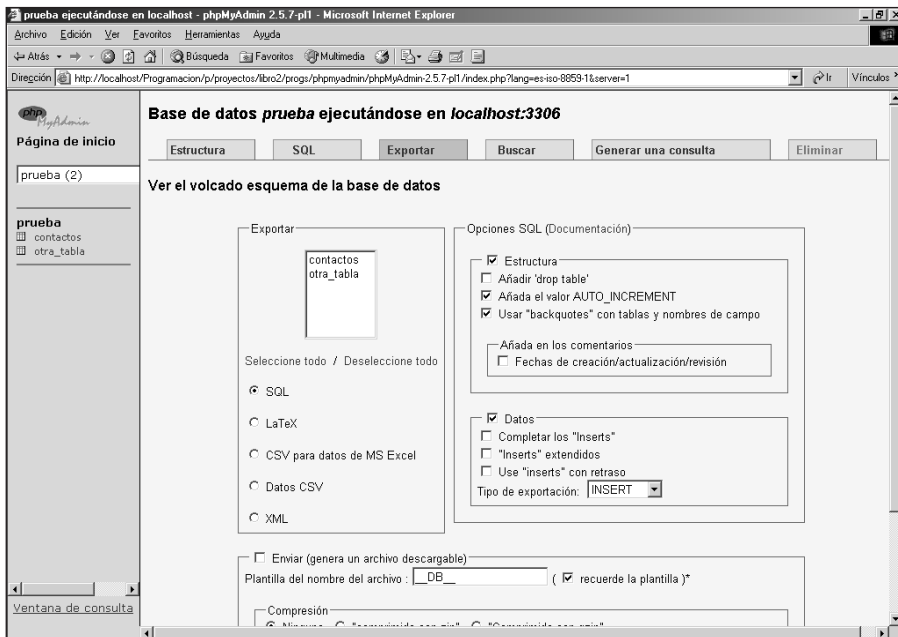


Figura 16. Exportando la información contenida en varias tablas, las cuales pueden ser seleccionadas desde la casilla de texto correspondiente.

RESUMEN

En este capítulo hemos visto cómo obtener e instalar phpMyAdmin, que nos permite administrar en forma remota una base de datos MySQL, con todas las operaciones requeridas para el buen funcionamiento de la misma, desde la creación de nuevas bases de datos y tablas hasta la exportación de todos los datos. Además, disponemos de opciones para poder agregar nuevos usuarios, reiniciar el servidor MySQL, etc.



ACTIVIDADES

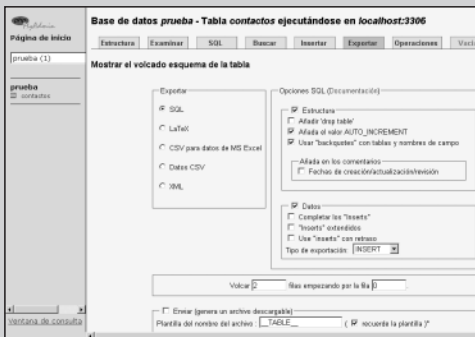
TEST DE AUTOEVALUACIÓN

1 ¿Qué es phpMyAdmin?



2 ¿Cómo se crea una base de datos?

3 ¿Se puede exportar la base de datos?



EJERCICIOS PRÁCTICOS

- ✓ Cree una base de datos para almacenar una agenda.

La base de datos deberá contener un tabla llamada contactos con los siguientes campos: id_contacto, nombre, apellido, teléfono, e_mail. Luego cargue algunos datos de ejemplos.

- ✓ Haga un backup de la base de datos contactos en su PC, y elimínela de su servidor.

- ✓ Restaure la base de datos de contactos con la copia de seguridad almacenada en su PC.