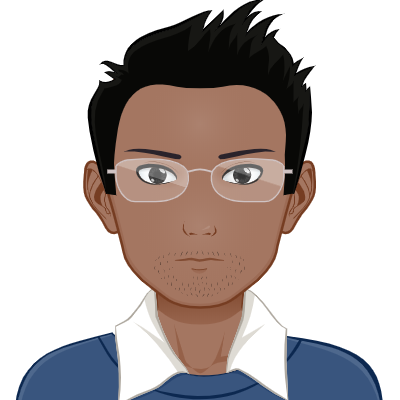
**BASE DE DATOS MYSQL**



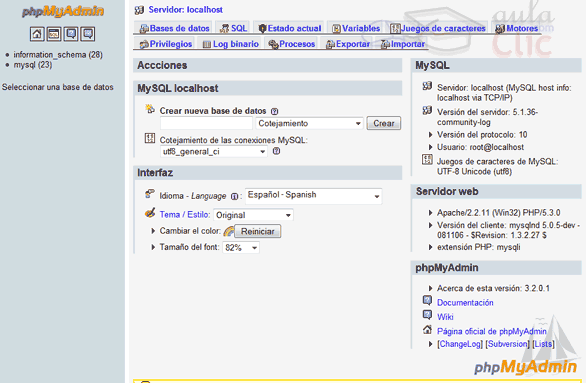
Para descargar XAMPP

<https://www.apachefriends.org/es/index.html>

En este curso no vamos a explicar en profundidad el uso de phpMyAdmin debido a su extensión. Pero sí daremos unas cuantas directrices para que puedas empezar a manejarte con él y a dar tus primeros pasos en el uso de las bases de datos.

Para abrir phpMyAdmin, haz clic en la opción phpMyAdmin del menú de WampServer que se obtienen al hacer clic sobre su icono. O accede a la dirección <http://localhost/phpmyadmin/>.

Se abrirá una ventana del navegador y mostrará la ventana principal de la aplicación:



Desde aquí podremos crear nuevas bases de datos, tablas y usuarios con sus respectivos privilegios.

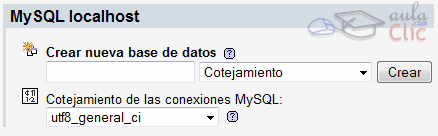
Pero, vayamos paso a paso.

Nota: Algunos iconos y menús pueden variar ligeramente si empleas una versión posterior a la del curso, aunque la funcionalidad debería ser la misma.

## **20.8. Crear una base de datos en phpMyAdmin**

Para crear una nueva base de datos en phpMyAdmin basta con escribir el nombre de la nueva base de datos en el cuadro de texto que aparece bajo Crear nueva base de datos en la página principal y pulsar el botón Crear.

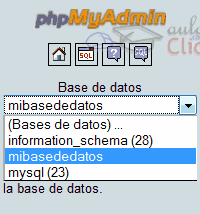
La nueva base de datos se creará automáticamente.



Puedes seleccionar un juego de caracteres en el desplegable Cotejamiento, si lo dejas en blanco se tomará el valor por defecto que variará según la configuración de cada servidor.

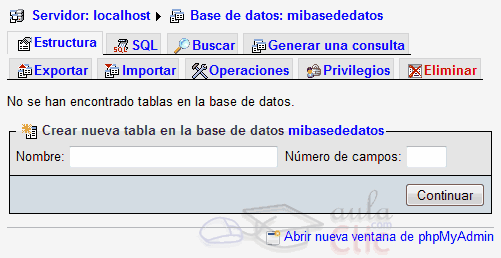
Es recomendable que escojas la opción utf-8\_spanish\_ci para maximizar la compatibilidad con todos los caracteres especiales del alfabeto en castellano y para que la salida en cualquier navegador sea correcta.

Una vez creada la base de datos podrás acceder a ella desde el menú de la izquierda. Despliega el listado Base de datos y selecciona la base de datos que quieras modificar:



## **Crear una tabla en phpMyAdmin**

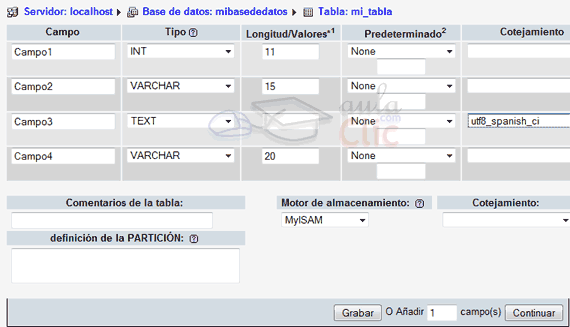
Cuando selecciones una Base de Datos en phpMyAdmin verás una nueva pantalla desde donde podrás ver todas las opciones para poder trabajar en ella:



En este caso en la pantalla principal encontraremos muy poca cosa porque no contiene ninguna tabla por ahora.

Para crear una tabla lo único que debes hacer es escribir su Nombre y especificar su Número de campos (luego puedes seguir añadiendo campos).

Cuando hayas terminado pulsa el botón Continuar y verás una pantalla como ésta:



Aquí deberás establecer la configuración de cada uno de los campos de la tabla.

Rápidamente definiremos las propiedades más frecuentemente utilizadas:

* Campo: El nombre del campo.
* Tipo: El tipo del valor del campo: texto (varchar o text), número entero (int), decimal (decimal), etc...
* Longitud/Valores Tamaño máximo del campo. Un texto de tamaño 15 sólo podrá tener 15 caracteres.
* Predeterminado: Un valor predeterminado para el campo, si al crear el registro no se le asigna otro valor.
* Cotejamiento: Juego de caracteres que utilizará el campo.
* Atributos: Dependientes del tipo de dato. Por ejemplo, que un numérico solo sea positivo.
* Nulo: Si el campo permite valores nulos, o debe rellenarse obligatoriamente.
* Índice: Indica si el campo está indexado. Puede ser clave principal, valores únicos, etc...
* A\_I: Si el campo es un auto-incremento, que se establecerá automáticamente al crear nuevos registros.
* Comentarios: Nos permite agregar comentarios descriptivos del campo o su contenido.

Pulsa el botón Grabar cuando hayas acabado y la tabla se creará.

Cada vez que realicemos una modificación se mostrará un mensaje similar a éste si todo ha ido bien:

http://www.aulaclic.es/dreamweaver-cs5/graficos/phpmyadmin_ok.gif

O parecido a esto si no se ha podido realizar la acción:

http://www.aulaclic.es/dreamweaver-cs5/graficos/phpmyadmin_wrong.gif

Ahora el aspecto de la ventana de la base de datos será diferente pues mostrará el listado de tablas contenidas con sus propiedades.

Haciendo clic en el botón propiedades al lado del nombre de la tabla o haciendo clic sobre ella en el menú de la izquierda podrás acceder a la vista en detalle de sus campos y propiedades. Para ello, asegúrate de que estás en la pestaña Estructura.

En esta última vista puedes hacer clic en la pestaña Examinar para ver el listado de todos los registros. Utiliza la pestaña Buscar para filtrar el listado por los valores de algún campo.

# **CONSULTAS sql**

# **Intro a SQL y MySQL - Tema 1 - Consultas básicas con una tabla**

## **1.1. Entrando al sistema**

Hemos visto cómo instalar MySQL, y como acceder a su consola. Ahora, como primer ejemplo, crearemos una **base de datos** sencilla, que llamaremos "ejemplo1". Esta base de datos contendrá una única **tabla**, llamada "agenda", que contendrá algunos datos de cada uno de nuestros amigos. Como es nuestra primera base de datos, no pretendemos que sea perfecta, sino sencilla, así que apenas guardaremos tres datos de cada amigo: el nombre, la dirección y la edad.

Para crear la base de datos, la estructura que contiene todo, usaremos "create database", seguido del nombre que tendrá la base de datos, como ya has visto:

**CREATE** **DATABASE** ejemplo1;

Si, por un despiste, vuelves a teclear la orden, no será ningún gran problema. Simplemente obtendrás un mensaje de error que te avisará de no se ha podido crear la base de datos porque ya existe.

Podemos tener varias bases de datos en nuestro SGBD (Sistema Gestor de Bases de Datos), así que cada vez que accedamos a MySQL deberemos comenzar por indicar cual de ellas queremos usar, con la orden "use":

**USE** ejemplo1;

## **1.2. Creación de una tabla**

Una base de datos, en general, estará formada por varios bloques de información llamados "**tablas**". En nuestro caso, nuestra única tabla almacenará los datos de nuestros amigos. Por tanto, el siguiente paso será decidir qué datos concretos (lo llamaremos "**campos**") guardaremos de cada amigo. Deberemos pensar también qué tamaño necesitaremos para cada uno de esos datos, porque al gestor de bases de datos habrá que dárselo bastante cuadriculado. Por ejemplo, podríamos decidir lo siguiente:

nombre - texto, hasta 20 letras  
dirección - texto, hasta 40 letras  
edad - números, de hasta 3 cifras

Cada gestor de bases de datos tendrá una forma de llamar a esos tipos de datos. Por ejemplo, en MySQL podemos usar "VARCHAR" para referirnos a texto hasta una cierta longitud variable, y "NUMERIC" para números de una determinada cantidad de cifras. Además, en la mayoría de gestores de base de datos, será recomendable (a veces incluso obligatorio) que los nombres de los campos no contengan acentos ni caracteres internacionales, y los nombres de tablas y campos se suelen indicar en minúsculas, de modo que la orden necesaria para crear esta tabla sería:

**CREATE** **TABLE** personas (

nombre varchar(20),

direccion varchar(40),

edad numeric(3)

);

## **1.3. Introducción de datos**

Para introducir datos usaremos la orden "**INSERT INTO**", e indicaremos tras la palabra "**VALUES**" los valores de los campos, en el mismo orden en el que se habían definido los campos. Los valores de los campos de texto se deberán indicar entre comillas, y los valores para campos numéricos ni necesitarán comillas, así:

**INSERT** **INTO** personas **VALUES** ('juan', 'su casa', 25);

Si no queremos indicar todos los datos, deberemos detallar qué campos vamos a introducir y en qué orden, así:

**INSERT** **INTO** personas (nombre, direccion) **VALUES** ('Héctor', 'calle 1');

(Como se ve en este ejemplo, los datos sí podrán contener acentos y respetar las mayúsculas y minúsculas donde sea necesario).

De igual modo, también será necesario indicar los campos si se desea introducir los datos en un orden distinto al de definición:

**INSERT** **INTO** personas (direccion, edad, nombre) **VALUES** ("Calle 2", 30, "Daniel");

(Y como puedes apreciar en este ejemplo, los campos de texto se pueden indicar también entre comillas dobles, si te resulta más cómodo).

## **1.4. Mostrar datos**

Para ver los datos almacenados en una tabla usaremos el formato "**SELECT** campos **FROM** tabla". Si queremos ver todos los campos, lo indicaremos con un asterisco:

**SELECT** \* **FROM** personas;

que, en nuestro caso, daría como resultado

+---------+-----------+------+

| nombre | direccion | edad |

+---------+-----------+------+

| juan | su casa | 25 |

| Héctor | calle 1 | NULL |

| Daniel | Calle 2 | 30 |

+---------+-----------+------+

Como puedes observar, no habíamos introducido los detalles de la edad de Héctor. No se muestra un cero ni un espacio en blanco, sino que se anota que hay un "**valor nulo**" (NULL) para ese campo. Usaremos esa peculiaridad más adelante.

Si queremos ver **sólo ciertos campos**, deberemos detallar sus nombres, en el orden deseado, separados por comas:

**SELECT** nombre, direccion **FROM** personas;

y obtendríamos

+---------+-----------+

| nombre | direccion |

+---------+-----------+

| juan | su casa |

| Héctor | calle 1 |

| Daniel | Calle 2 |

+---------+-----------+

## **1.5. Buscar por contenido**

Normalmente no querremos ver todos los datos que hemos introducido, sino sólo aquellos que cumplan cierta condición. Esta condición se indica añadiendo un apartado **WHERE** a la orden "SELECT". Por ejemplo, se podrían ver los datos de una persona a partir de su nombre de la siguiente forma:

**SELECT** nombre, direccion **FROM** personas **WHERE** nombre = 'juan';

+--------+-----------+

| nombre | direccion |

+--------+-----------+

| juan | su casa |

+--------+-----------+

En ocasiones no querremos comparar todo el campo con un texto exacto, sino ver si contiene un cierto texto (por ejemplo, porque sólo sepamos un apellido o parte de la calle). En ese caso, no compararíamos con el símbolo "igual" (=), sino que usaríamos la palabra "**LIKE**", y para las partes que no conozcamos usaremos el comodín "%", como en este ejemplo:

**SELECT** nombre, direccion **FROM** personas **WHERE** direccion **LIKE** '%calle%';

que nos diría el nombre y la dirección de nuestros amigos que viven en lugares que contengan la palabra "calle", aunque ésta pueda estar precedida por cualquier texto (%) y quizá también con cualquier texto (%) a continuación:

+---------+-----------+

| nombre | direccion |

+---------+-----------+

| Héctor | calle 1 |

| Daniel | Calle 2 |

+---------+-----------+

## **1.6. Mostrar datos ordenados**

Cuando una consulta devuelva muchos resultados, resultará poco legible si éstos se encuentran desordenados. Por eso, será frecuente añadir "ORDER BY" seguido del nombre del campo que se quiere usar para ordenar (por ejemplo "ORDER BY nombre").

Es posible indicar varios campos de ordenación, de modo que se mire más de un criterio en caso de que un campo coincida (por ejemplo "ORDER BY apellidos, nombre").

Si se quiere que alguno de los criterios de ordenación se aplique en orden contrario (por ejemplo, textos de la Z a la A o números del valor más grande al más pequeño), se puede añadir la palabra "DESC" (de "descending", "descendiendo") al criterio correspondiente, como en "ORDER BY precio DESC". Si no se indica esa palabra, se sobreentiende que se quiere obtener los resultados en orden ascendente (lo que indicaría la palabra "ASC", que es opcional).

**SELECT** nombre, direccion

**FROM** personas

**WHERE** direccion **LIKE** '%calle%'

**ORDER** **BY** nombre, direccion **DESC**;

+---------+-----------+

| nombre | direccion |

+---------+-----------+

| Daniel | Calle 2 |

| Héctor | calle 1 |

+---------+-----------+

## **1.7. Salir de MySQL**

Es suficiente por hoy. Para terminar nuestra sesión de MySQL, tecleamos la orden **quit** o **exit** y volveremos al sistema operativo.

## **1.8. Ejercicios propuestos**

Si no sólo quieres leer, sino que tu intención es aprender y asegurarte de que consolidas los conocimientos, deberías practicar un poco. Aquí tienes una primera tanda de ejercicios propuestos. Si te surgen dudas al resolverlos o quieres asegurarte de que tus respuestas son correctas, puedes encontrarme en el [foro de SQL de AprendeAProgramar.com](http://www.aprendeaprogramar.com/mod/forum/view.php?id=333).

* 1.1. Crea una base de datos llamada "ejercicio1".
* 1.2. Crea en ella una tabla llamada "ciudades". De cada ciudad se deseará guardar su nombre (de 40 letras o menos) y su población (de 9 cifras o menos).
* 1.3. Usando el formato abreviado de INSERT, añade la ciudad "Alicante", con 328.648 habitantes.
* 1.4. Indicando primero la población y luego el nombre, añade una ciudad de 3.141.991 habitantes llamada "Madrid".
* 1.5. Muestra todos los datos de todas las ciudades existentes en la base de datos.
* 1.6. Muestra la población de la ciudad llamada "Madrid".
* 1.7. Muestra el nombre y la población de todas las ciudades que contienen una letra "e" en su nombre.
* 1.8. Muestra el nombre y la población de todas las ciudades, ordenadas de la más poblada a la menos poblada. Si dos o más ciudades tienen mismca cantidad de habitantes, se mostrarán ordenadas de la A a la Z.