

Bases de Datos



Índice

Introducción

Relaciones entre tablas

MySQL workbench

Comandos de MySQL



Introducción

Una base de datos es cualquier cosa que agrupe información y que tenga un sentido.

Para poder acceder a los datos de una base de datos formada en una computadora, es necesario utilizar una herramienta llamada “*Relational DataBase Management System*” como, por ejemplo, MySQL, Postgres, MariaDB, Oracle, etc.

Estos softwares se van a encargar de:

- Respaldo
- Importar
- Exportar
- Concurrencia (gestiona el acceso múltiple a una misma base de datos).
- Siempre existe prioridad para leer el dato**
- Operaciones CRUD (desde cualquier lenguaje de programación) mediante *query*
 - Create crear un registro
 - Read leer un registro
 - Update actualizar un registro
 - Delete borrar un registro

SQL

Bases de datos relacionales. Se organiza la información en tablas con columnas y filas (*registros*). Cada celda almacena un *dato*.

Una columna almacenará los identificadores (ID) de cada fila (*registro*). Dichos identificadores se denominan **primary key**.

No-SQL

Almacena la información en formato JSON, BSON, BLOB, key-value, entre otros.



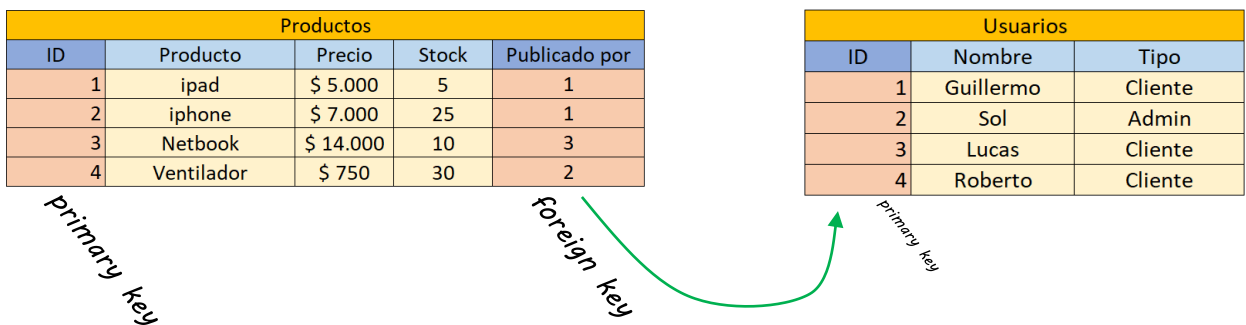
Relaciones entre tablas

Se dice que una base de datos es relacional porque se relacionan unas tablas con otras.

1 → n

Un registro de una tabla se asocia con otra tabla.

Por ejemplo, una tabla de Usuarios está asociada con una tabla de Productos mediante una **foreign key** (representa el ID de otra tabla).

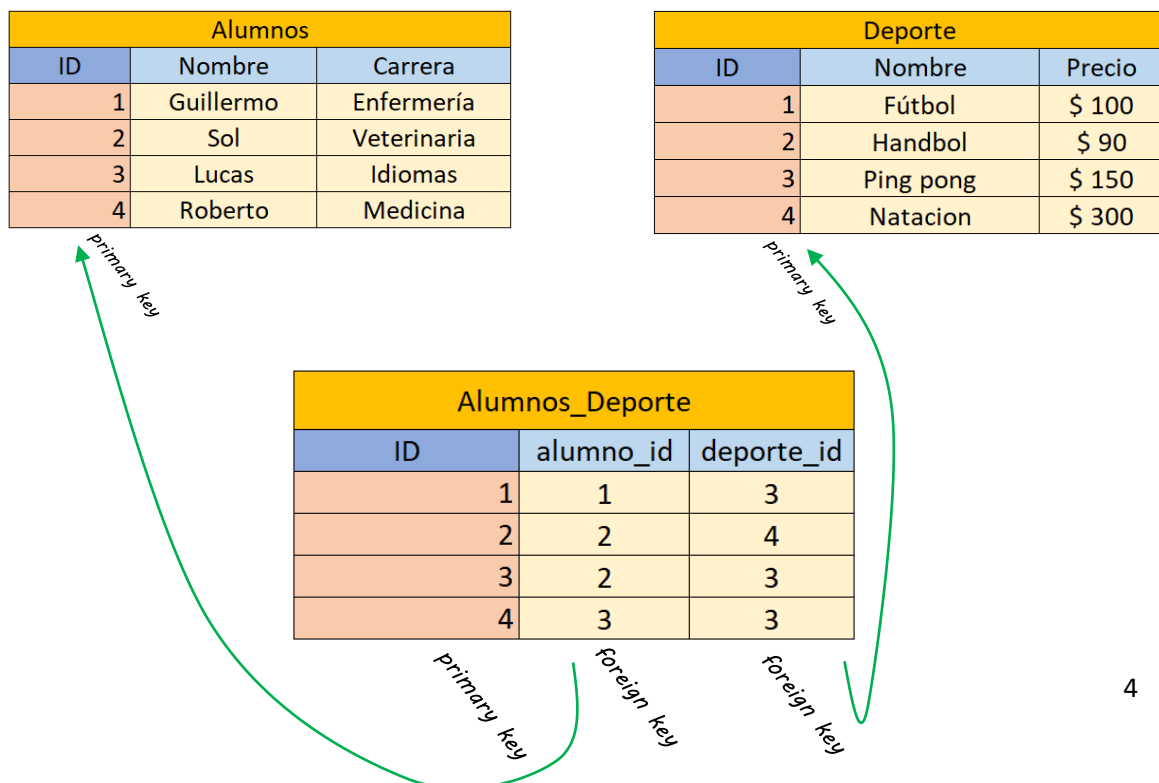


n ↔ n

Más de un registro de una tabla se puede relacionar con más de un registro de otra tabla.

Por ejemplo, una tabla de Alumnos asociado a una tabla de Deportes.

Para ello, se utiliza una tercera tabla que asocie a las dos anteriores.



MySQL workbench



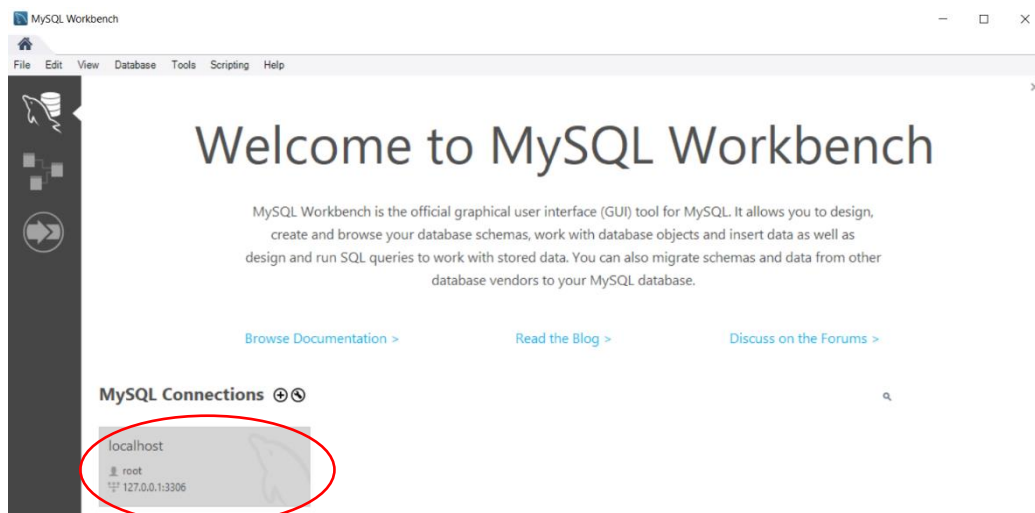
Para poder empezar a utilizar MySQL, es necesario descargarlo desde su página oficial.

Atención: Debe instalar **MySQL Installer**, no ~~MySQL Community Server~~.

La página para descargar es: <https://dev.mysql.com/downloads/installer/>

Seleccionar la última versión, y luego descargarlo. Puede hacerlo sin necesidad de registrarse desde el botón que dice: **No thanks, just start my download**

Una vez instalado, se mostrará una pantalla con una primera conexión local, nuestro localhost. Podrás ingresar allí para comenzar con la primer base de datos.



Comandos de MySQL

- Create

XAMPP

Paquete de software libre que consiste en el sistema de gestión de bases de datos **MySQL**, servidor web Apache e intérpretes PHP y Perl.

Permite montar un servidor web Apache para poder utilizar bases de datos de **MySQL** y programar en PHP y Perl. Además de utilizar los lenguajes de HTML, CSS y JavaScript.

X: para cualquier sistema operativo

A: Apache

M: MariaDB (o **MySQL**)

P: PHP

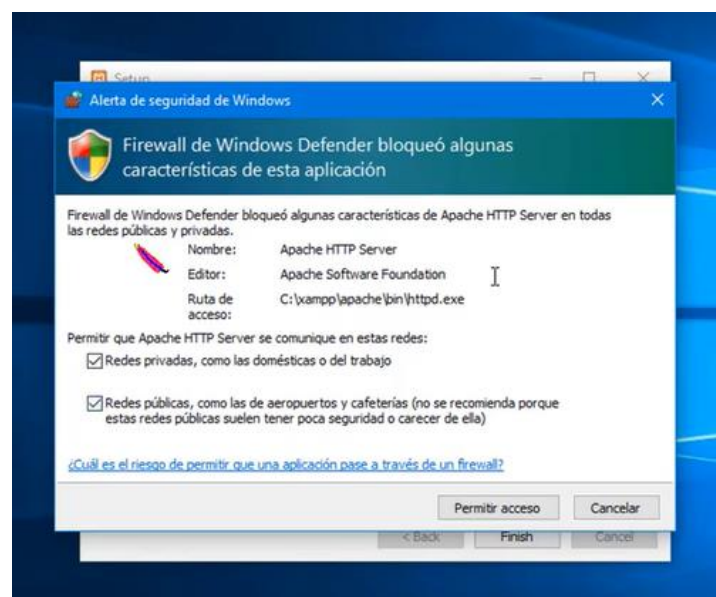
P: Perl



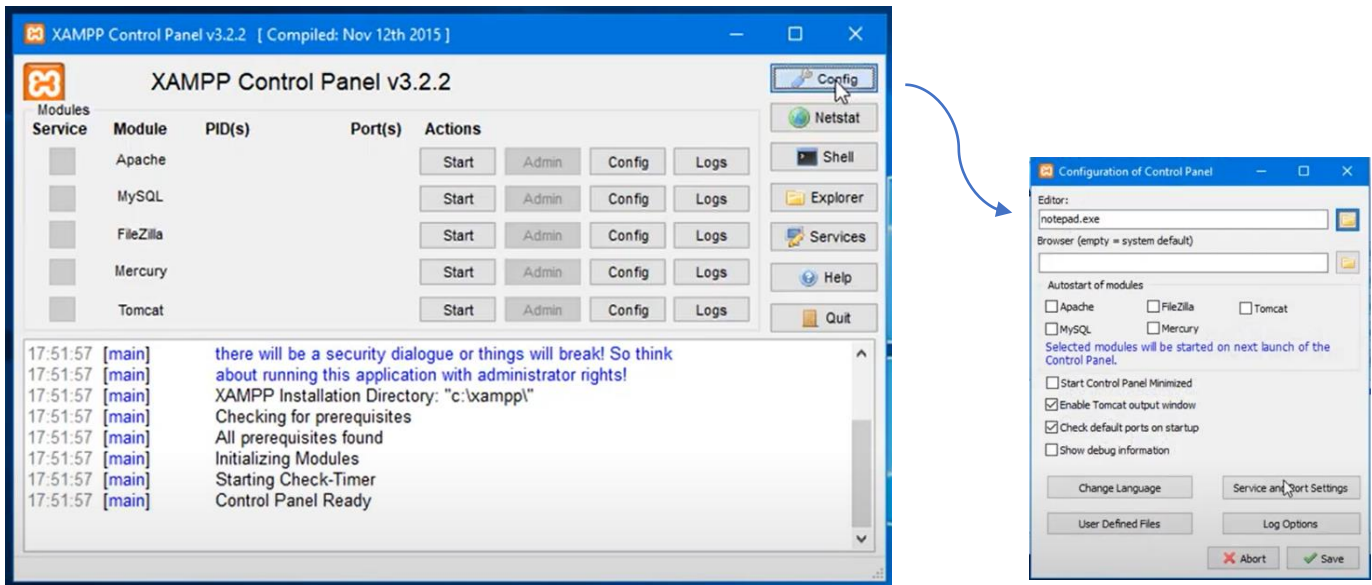
Instalación

Descargar desde la página oficial <https://www.apachefriends.org/es/index.html>

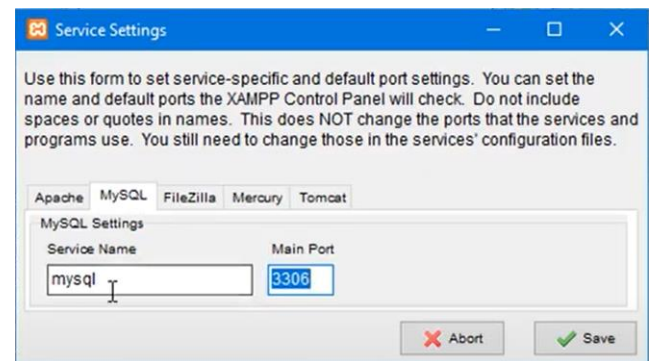
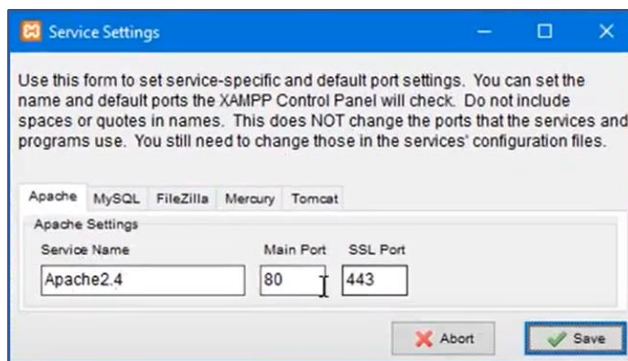
Al momento en que el Firewall pida acceso, permitir tanto para redes públicas como para redes privadas (luego puede configurarse mejor).



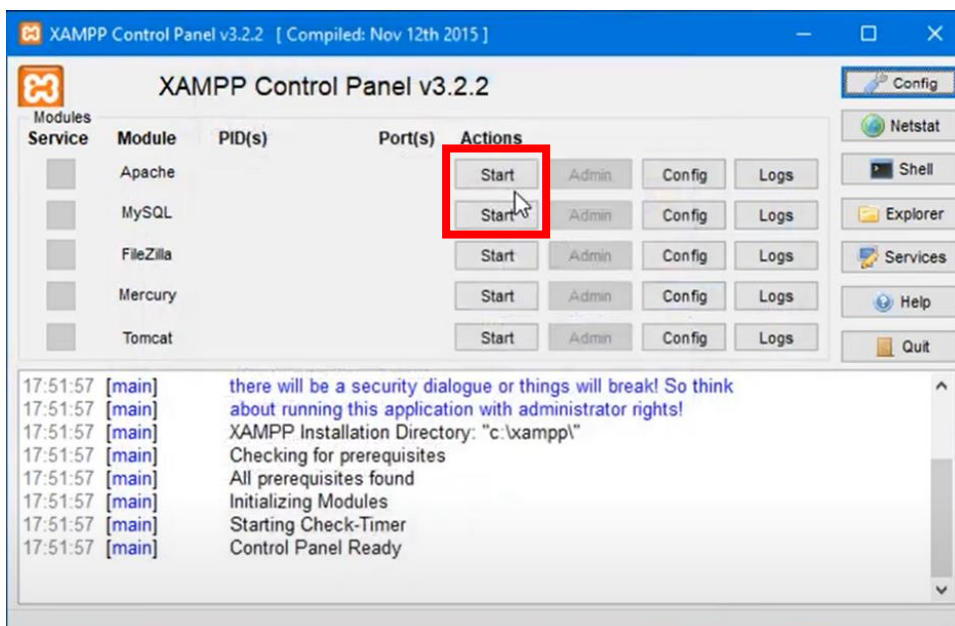
Finalmente, cuando se haya instalado, se visualizará la siguiente pantalla:



Es necesario comprobar que tanto Apache está en 80 – 443 y que MySQL está en el puerto 3306.

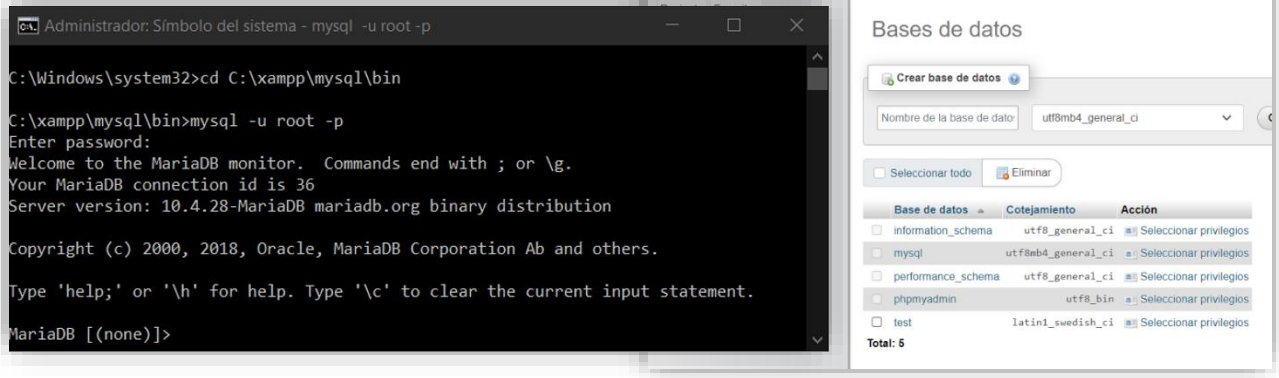


Comprobado lo anterior, se puede iniciar tanto el servidor Apache como la base de datos.



De esta forma, quedan activa la Base de Datos con el servidor que puede ser usado por nuestra página web, la cual se puede visualizar desde phpMyAdmin.

O bien, trabajar desde la consola del sistema con privilegios de administrador.



➤ Ver Bases de Datos existentes

SHOW DATABASES;

➤ Crear una Base de Datos

CREATE DATABASE nombre_database;

➤ Acceder a una Base de Datos

USE nombre_database;

- Ver Tablas
- Ver detalles de una Tabla
- Crear una Tabla

SHOW TABLES;

DESCRIBE nombre_tabla;

CREATE TABLE nombre_tabla (parámetros);

donde los **parámetros** se envía el nombre de la columna, el tipo de dato, se especifica si puede ser NULL y cuál es la columna primaria

CREATE TABLE table_name (id INT AUTO_INCREMENT, name varchar(90) NOT NULL, number INT, PRIMARY KEY(id));

• Seleccionar una Columna

SELECT * FROM nombre_tabla;

• Ingresar una Fila

INSERT INTO nombre_tabla (id, name, number) VALUES (null, 'Guille', 28);

• Borrar una Fila

DELETE FROM nombre_tabla WHERE id = 2;

• Editar una Fila

UPDATE nombre_tabla SET name = 'Guille' WHERE name = 'Nico';

Además, se puede usar **fecha DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP** para crear una columna que guarde la fecha en la que se realice un registro.

Existen varias formas de usar WHERE:

```
***** SELECCION POR ALGO ESPECIFICO
select * from usuarios WHERE nombre like 'y%';
select * from usuarios WHERE nombre like 'm%a';
select * from usuarios WHERE email like '%com';
select * from usuarios ORDER BY nombre DESC;
select * from usuarios ORDER BY nombre ASC;
select * from usuarios LIMIT 1;
select * from usuarios LIMIT 2;
select * from usuarios LIMIT 2,3;
DELETE FROM usuarios WHERE id=2;
```

