

# Introducción a MariaDB/MySQL



En este documento veremos cómo instalar y poner en funcionamiento un SGBD relacional. Tomaremos como ejemplo el sistema MariaDB/MySQL, que puede ser instalado en diversas plataformas.

## 1. Instalación

Para poder tener un sistema MariaDB/MySQL funcionando y poderlo gestionar cómodamente, vamos a instalar el servidor MySQL como servicio en nuestro sistema.

**Versión requerida:** ninguna en particular, podemos instalar la última que tenga el gestor de paquetes que utilicemos, o la última que descarguemos de la web oficial, según el caso.

### 1.1. Instalación en Linux

Para instalar el servidor en un sistema Debian, Ubuntu o similar podemos seguir estos pasos desde el terminal:

```
sudo apt update
sudo apt install mysql-server
```

Esto instalará el servidor básico en el sistema y lo dejará iniciado como servicio. Con esto tenemos suficiente para desarrollar las actividades del curso. Podemos comprobar que el servidor está en marcha con el comando:

```
sudo systemctl status mysql.service
```

Y podemos parar, iniciar o reiniciar el servicio con estos comandos:

```
sudo systemctl stop mysql.service
sudo systemctl start mysql.service
sudo systemctl restart mysql.service
```

### Instalación segura

El siguiente paso (opcional) es hacer una instalación más segura, ya que por defecto el usuario *root* tiene una contraseña vacía. Primero deberemos acceder a la consola de MySQL para establecer la contraseña de ROOT.

Escribiremos `sudo mysql` para que se inicie la consola y, desde ésta, escribimos el siguiente comando:

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password by 'nuevo_passwor
```

donde *nuevo\_password* será el password que queramos que tenga *root*. En la máquina virtual proporcionada tendréis un usuario *root* con password *root*, como ejemplo. Si queremos añadir nuevos usuarios y permisos, podemos consultar cómo hacerlo en webs como [ésta](#).

Una vez asegurado el usuario *root*, podemos ejecutar este comando para configurar una instalación segura:

```
sudo mysql_secure_installation
```

Con esto se iniciará un asistente que nos guiará a través de una serie de pasos para eliminar tablas de test, permitir/prohibir acceso remoto de usuario *root*, etc. Podemos configurar estas opciones como mejor nos parezca.

## 1.2. Instalación en otros sistemas (Windows y Mac)

La instalación de MySQL en otros sistemas no forma parte de los contenidos importantes del curso, ya que es recomendable que utilicéis la máquina virtual proporcionada. No obstante, si queréis realizar esta instalación, aquí se os facilitan algunos recursos externos que podéis consultar para hacerlo.

- Si queremos instalar MySQL en Windows, podemos seguir [esta guía](#).
- Para instalarlo en MacOSX, podemos seguir [esta otra guía](#).
- También tenéis disponible la información para ambos sistemas en la [documentación oficial](#) del sitio.

## 1.3. XAMPP

Como alternativa a instalar MySQL como servicio en el sistema podemos optar por instalar un sistema XAMPP, que integra, entre otras cosas, un servidor web Apache, el SGBD MySQL y el lenguaje PHP.

Para instalar XAMPP, basta con descargarlo de su [web oficial](#) y seguir los pasos del asistente. Nos basta con tener instalado Apache, MySQL y PHP, así que podemos descartar otras opciones que nos ofrezca, si nos da a elegir.

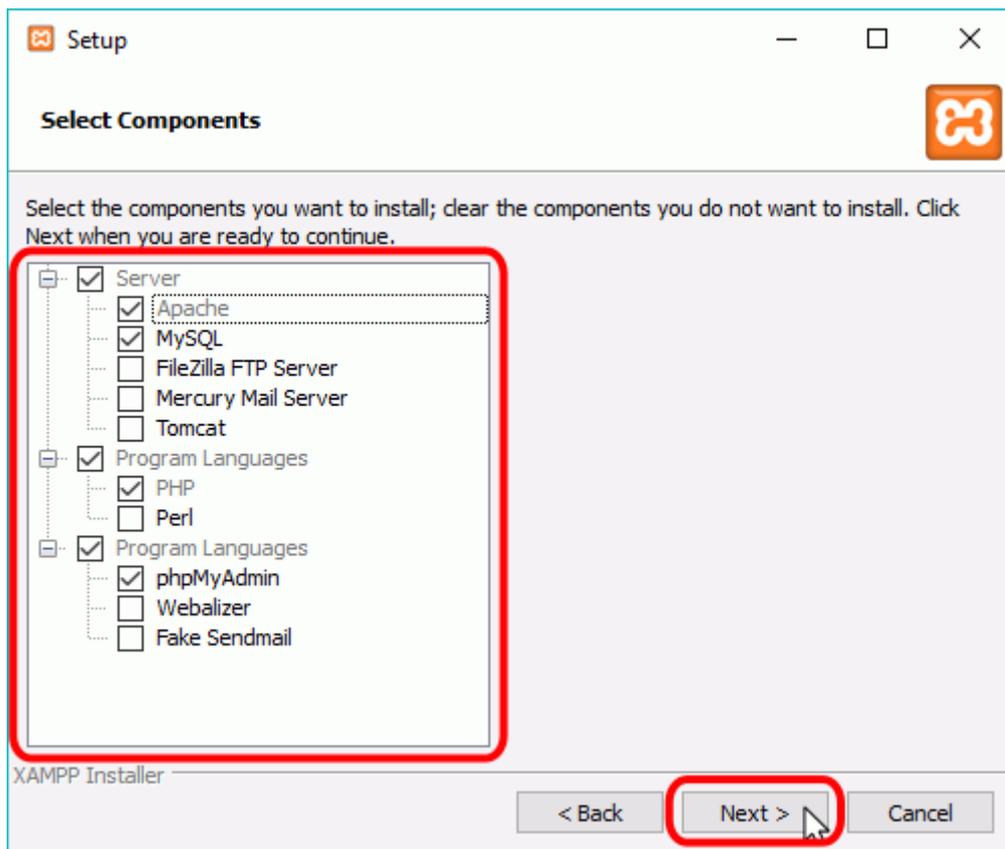
### Linux

En el caso de Linux, debemos dar permisos de ejecución y ejecutar el archivo `.run` que descarguemos desde algún terminal, con permisos de administrador (*sudo*). Suponiendo que el archivo se llame `xampp-linux-x64-7.4.5-installer.run`, por ejemplo, los pasos son los siguientes (desde la carpeta donde lo hemos descargado):

```
sudo chmod +x xampp-linux-x64-7.4.5-installer.run
sudo ./xampp-linux-x64-7.4.5-installer.run
```

## Windows y MacOSX

En el caso de **Windows** o **Mac OSX** simplemente hay que lanzar el instalador y seguir los pasos, eligiendo las opciones que nos interese instalar (al menos, Apache, MySQL y PHP), si nos dan a elegir. Así es como podemos dejarlo en el caso de Windows, por ejemplo:



### 1.3.1. El manager de XAMPP

XAMPP proporciona una herramienta *manager* o *panel de control* que nos permite gestionar en todo momento los servicios activos.

En el caso de **Linux** se encuentra en **/opt/lampp/manager-linux-x64.run**. Podemos acceder a la carpeta desde el terminal para ejecutarlo (con permisos de superusuario), o bien crear algún acceso directo en otra ubicación que nos resulte más cómoda. Por ejemplo, podemos crear un acceso directo en el escritorio con el editor *nano* o con el propio editor Visual Studio Code que hemos instalado previamente. Suponiendo la carpeta `/home/alumno/Escritorio/` o `/home/alumno/Desktop/`, podemos primero crear el archivo vacío:

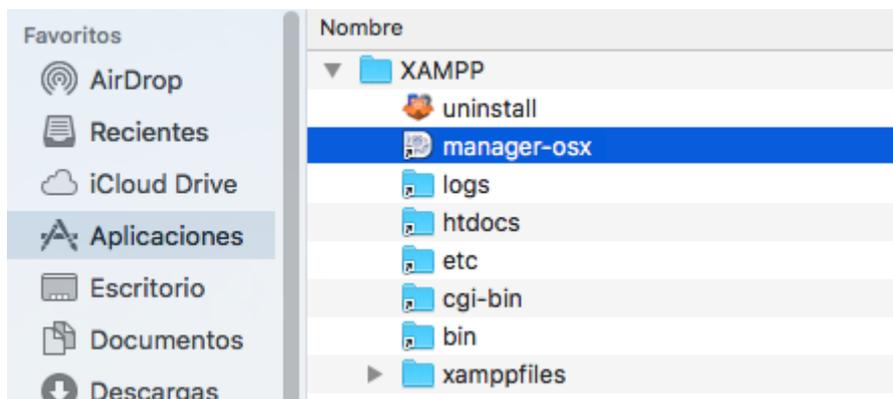
```
touch /home/alumno/Desktop/XAMPP.desktop
```

Editamos el contenido del archivo y añadimos las líneas de configuración para el acceso directo:

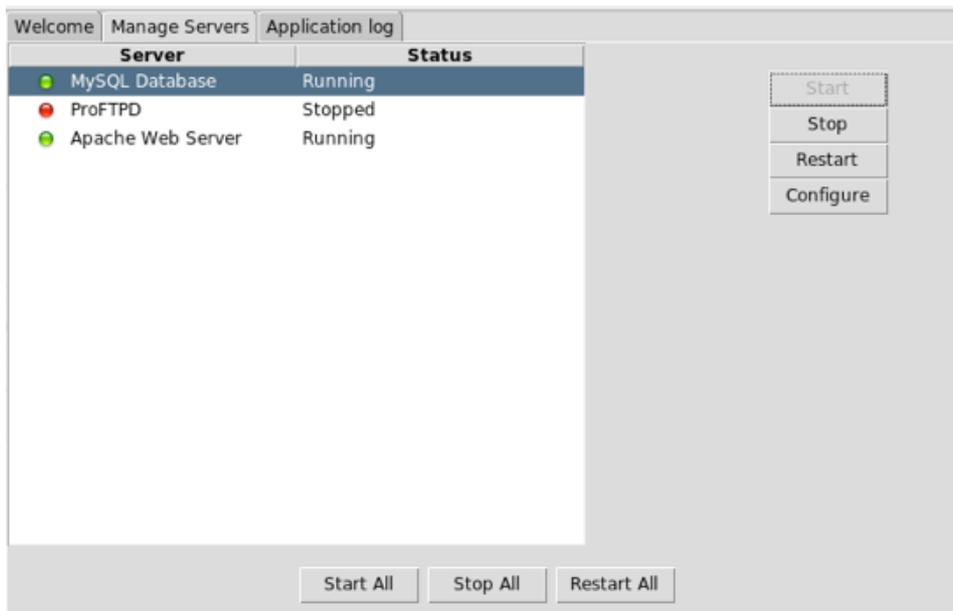
```
[Desktop Entry]
Encoding=UTF-8
Name=Manager XAMPP
Comment=Manager XAMPP
Exec=sudo /opt/lampp/manager-linux-x64.run
Icon=/opt/lampp/htdocs/favicon.ico
Categories=Aplicaciones;Programación;Web
Version=8.1.6
Type=Application
Terminal=1
```

**NOTA:** la versión del programa (atributo *Version*) dependerá de la versión que hayamos instalado de XAMPP en el momento concreto. El atributo *Terminal* lo ponemos a 1 para poder especificar el password de superusuario al ejecutar, de lo contrario no funcionará.

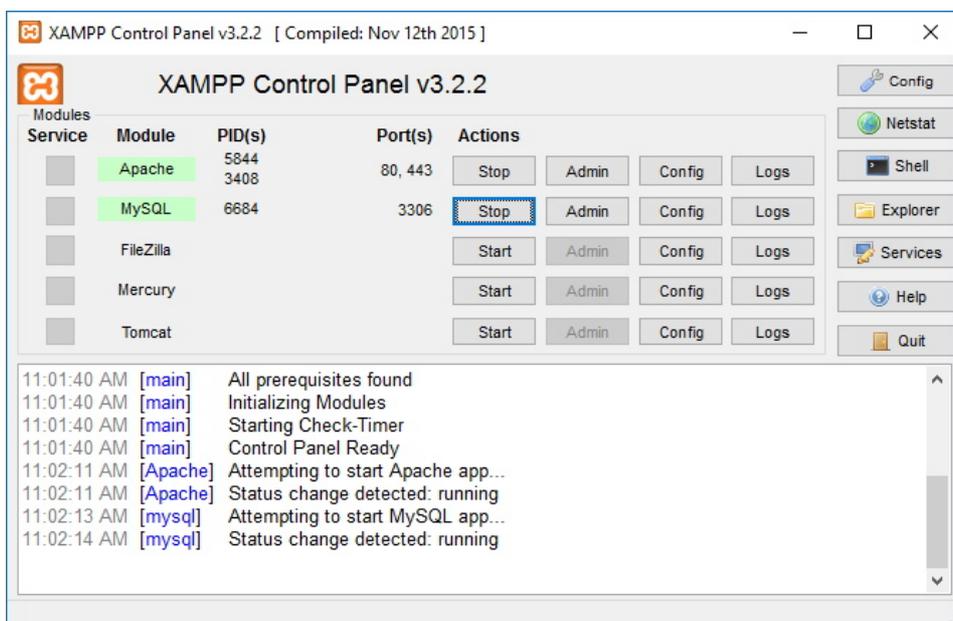
En el caso de **Windows**, dicho manager está en la carpeta de instalación (típicamente *C:\xampp*), en un archivo llamado *xampp-control.exe*, que podemos ejecutar. En el caso de **Mac OSX**, se habrá creado un acceso en la sección de *Aplicaciones* para poder poner en marcha este manager.



El manager nos permitirá lanzar o detener cada servidor. Para las pruebas que haremos deberemos tener iniciados tanto Apache como MySQL. En Linux y Mac OS X tendrá una apariencia como ésta aproximadamente:



En el caso de Windows la apariencia es algo diferente, aunque igualmente funcional:



Por defecto, Apache estará escuchando en el puerto 80 (o 443 para conexiones SSL), y MySQL en el 3306. Podemos modificar estos puertos en los respectivos archivos de configuración ("*httpd.conf*" y "*my.cnf*"), dentro de las carpetas de la instalación de XAMPP (la ubicación concreta de estos archivos varía entre versiones y entre sistemas operativos).

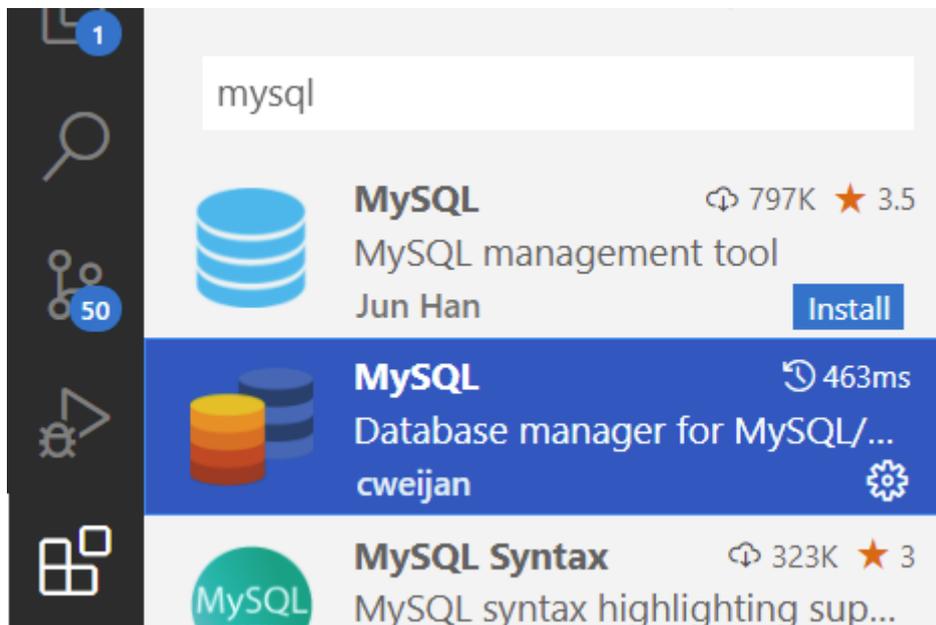
## 2. Gestión de bases de datos

Para poder gestionar cómodamente las bases de datos MySQL tenemos algunas alternativas:

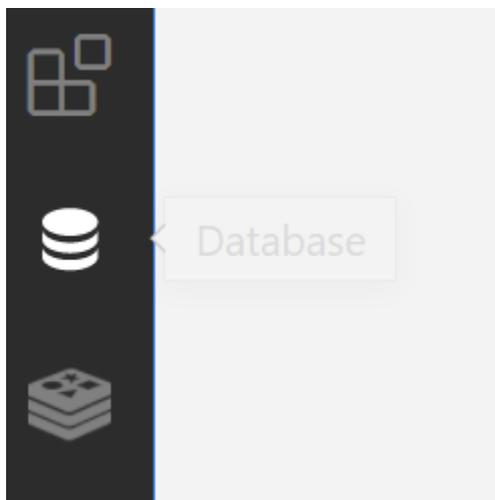
- Herramienta *phpMyAdmin*, si hemos instalado XAMPP. No veremos aquí cómo se trabaja con esta herramienta, pero podéis consultar algunos pasos [aquí](#).
- Extensión para Visual Studio Code, que nos sirve en todos los supuestos

## 2.1. El plugin *MySQL* para VS Code

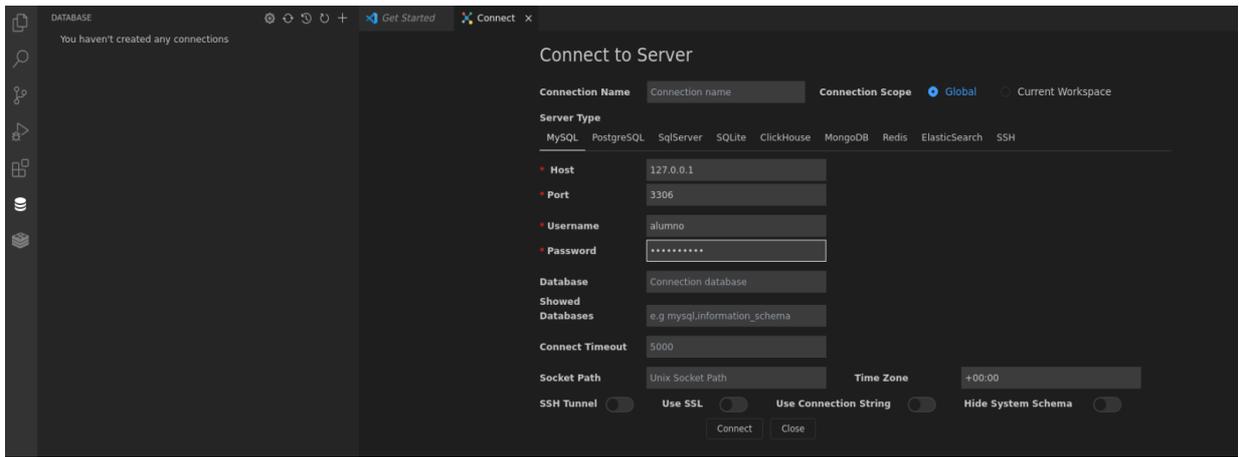
La extensión *MySQL* (desarrollada por *cweijan*) para Visual Studio Code nos va a permitir conectar y gestionar bases de datos MySQL (y también MongoDB) desde el propio IDE.



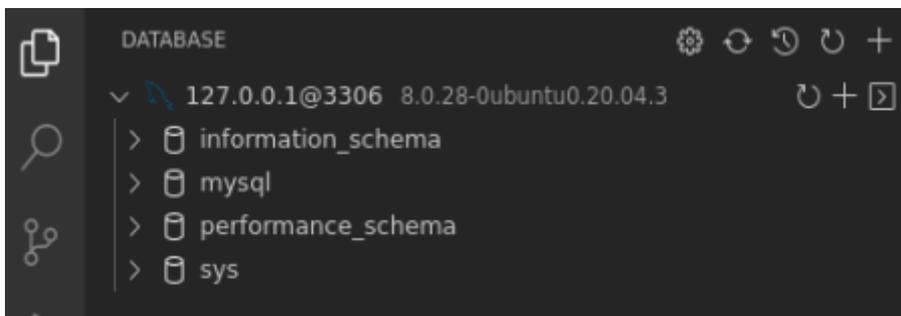
Una vez la tengamos instalada, veremos que en la barra izquierda aparecen dos iconos nuevos, para gestionar bases de datos relacionales y No-SQL, respectivamente:



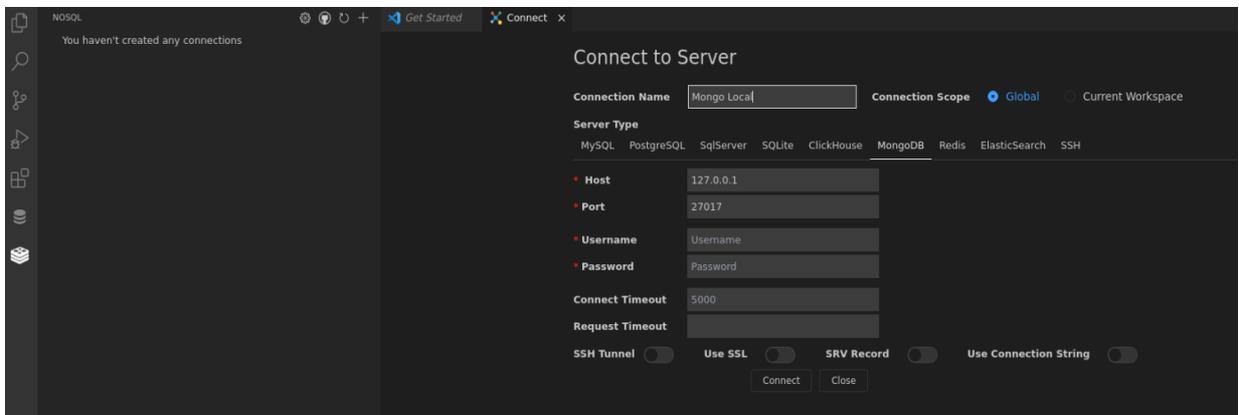
Tenemos que hacer clic en cada una de ellas y crear la conexión a cada servidor (icono de + en la parte superior). Nos rellenará ya los campos con los parámetros por defecto, y sólo tenemos que poner un nombre a cada conexión. Así podría quedar la de MySQL (cambiando el usuario y contraseña por el que tengamos en nuestro sistema):



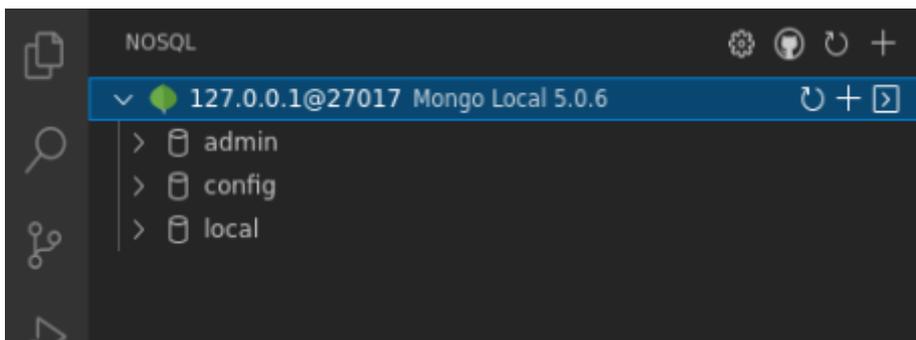
Tras conectar, podremos ver en el panel izquierdo las bases de datos existentes en dicho servidor:



Así configuraremos la conexión de MongoDB (sin usuario ni contraseña):



Y nuevamente, podremos ver a la izquierda las bases de datos existentes:



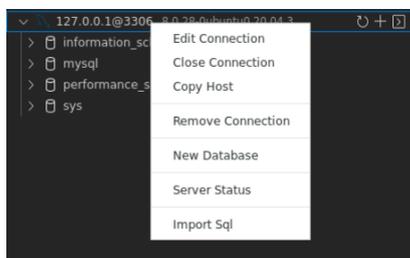
**NOTA:** en la máquina virtual con todo el software instalado ya se dejan configuradas ambas conexiones (a MySQL y a MongoDB) desde esta extensión de Visual Studio Code.

## 2.2. Operaciones contra la base de datos

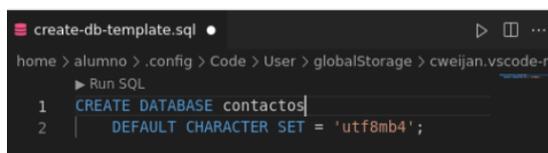
Desde la extensión *MySQL* podemos hacer las tareas habituales de gestión de una base de datos, tales como:

- Crear bases de datos
- Importar/Exportar bases de datos desde/a scripts de *backup*
- Crear tablas con sus campos
- ... etc.

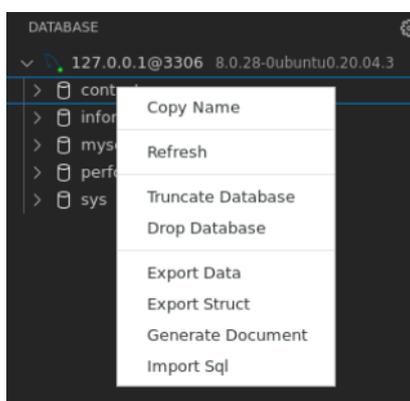
Para crear una base de datos, hacemos clic derecho sobre la conexión (panel izquierdo) y elegimos la opción correspondiente:



En la instrucción SQL que aparece en el panel derecho, completamos el nombre de la base de datos y pulsamos en el icono de *play* en la parte superior derecha para ejecutar la instrucción y crear la base de datos.

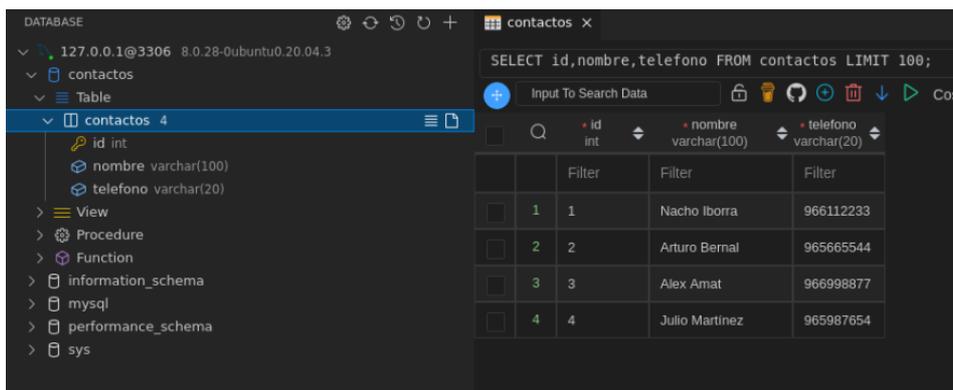


Para rellenar el contenido de la base de datos creada, podemos importar las instrucciones de algún *script* SQL. Para ello hacemos clic derecho sobre la base de datos y elegimos la opción de importar del menú contextual:

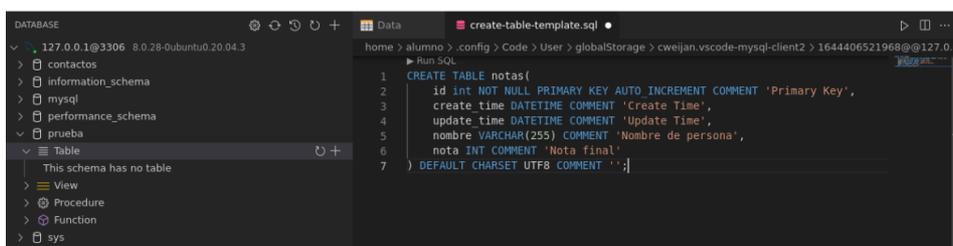


**NOTA:** en ocasiones el propio *script* SQL ya tiene la instrucción *CREATE DATABASE*, con lo que no es necesario crearla antes de importar. Pero puede ser una buena práctica para evitar que dé error. A fin de cuentas, si la base de datos ya existe, esta instrucción no la creará de nuevo.

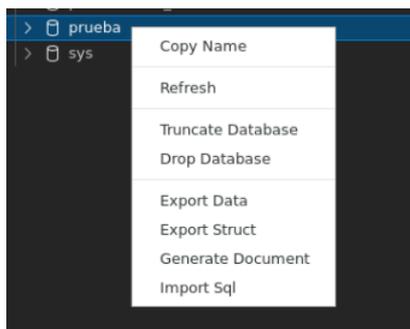
Una vez tengamos la base de datos importada, haciendo doble clic en cada tabla podemos ver su contenido en el panel derecho:



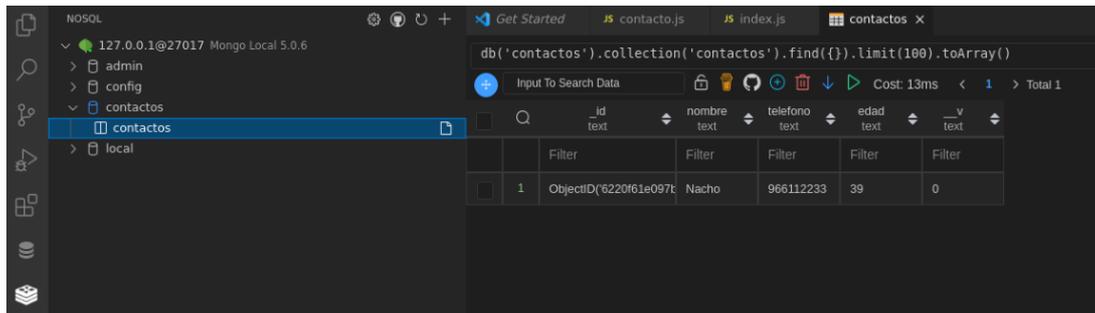
En ocasiones también nos puede interesar crear una base de datos vacía y definir sus tablas desde cero. En ese caso, desplegamos la base de datos en el panel izquierdo, vamos al apartado de *Tables* y hacemos clic en el botón **+**. En la parte derecha completamos el *script* de creación de la tabla con los campos, tipos y restricciones que queramos. Finalmente, pulsamos el botón *play* de la esquina superior derecha.



Finalmente, si queremos guardar los datos en un archivo *script* de *backup*, hacemos clic derecho sobre la base de datos, y elegimos la opción *Export Data* (si queremos guardar las tablas con sus datos) o *Export Struct* (si sólo nos interesa la estructura de las tablas, sin la información que contienen).



Como anexo, desde la extensión *No-SQL* también podemos gestionar MongoDB, conectar a bases de datos, crear colecciones, ver el contenido, etc. Sin embargo, la extensión *MongoDB for Visual Studio Code* que hemos visto en [otros apartados](#) ofrece una gestión algo más completa.



### Ejercicio 1:

Crea desde la extensión para VS Code una base de datos llamada *contactos* en MySQL. Descarga [este archivo comprimido](#) que contiene un *backup* de esa base de datos, e importa el *script* SQL desde la propia base de datos. Prueba a examinar el contenido desde VS Code.