

Software necesario



Para trabajar con Node.js, es necesario disponer de una serie de elementos software instalados. En este documento haremos un repaso del software necesario, y daremos las pautas para instalarlo en distintos sistemas.

En concreto, los elementos que vamos a instalar a lo largo de este documento son:

- Un editor de código (IDE) con el que desarrollar las aplicaciones. En nuestro caso, emplearemos **Visual Studio Code**.
- El propio framework **Node.js**
- Un servidor de bases de datos relacional (**MySQL**, por ejemplo) y otro No-SQL (**MongoDB**)

También se facilitará [al final](#) de este documento un enlace a una máquina virtual donde poder instalar todo el software, así como otra máquina virtual con el software ya preinstalado y listo para utilizarse, usando las versiones que se indican a lo largo del documento.

1. Visual Studio Code

Como IDE para desarrollar nuestras aplicaciones emplearemos **Visual Studio Code**, que es uno de los IDEs más versátiles que existen hoy en día para desarrollo web.

Versión requerida: ninguna en particular, sirve con la última versión disponible.

Desde la [web oficial](#) de Visual Studio Code podemos descargarlo para la plataforma deseada.

Linux (Debian)

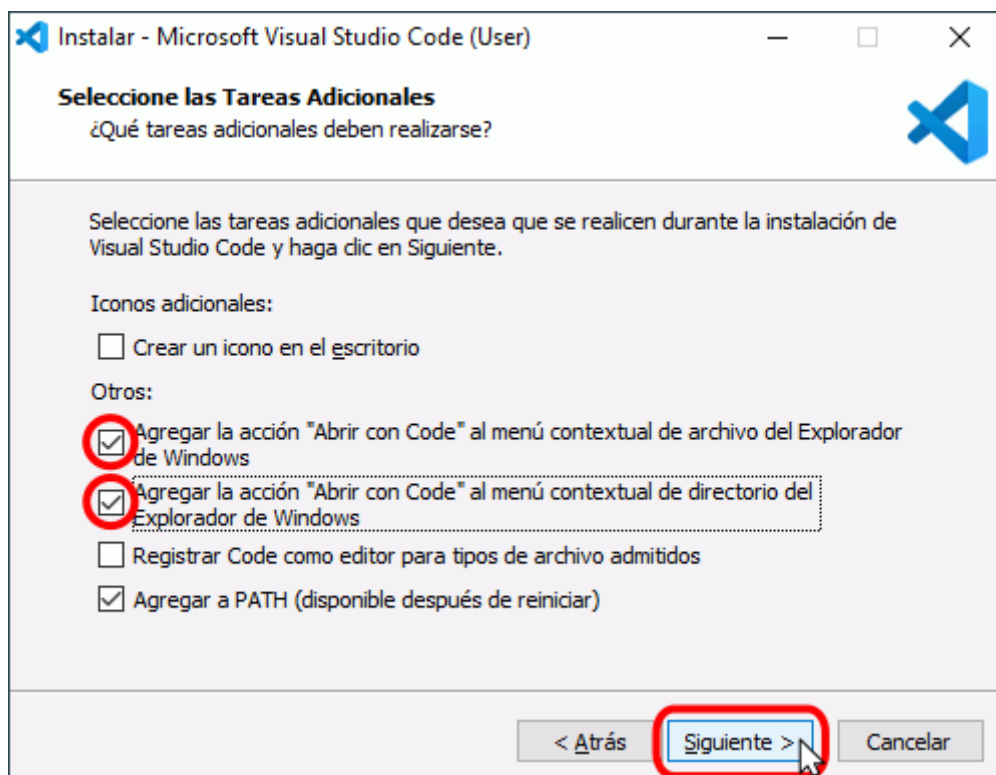
En el caso de nuestra máquina virtual o una distribución similar, descargaremos un archivo `.deb`. Una vez descargado, accedemos por terminal a la carpeta donde esté y ejecutamos este comando para instalarlo:

```
sudo dpkg -i nombre_del_archivo.deb
```

Se creará automáticamente un acceso directo en el menú de inicio, dentro de la sección de *Programación* en el caso de Ubuntu.

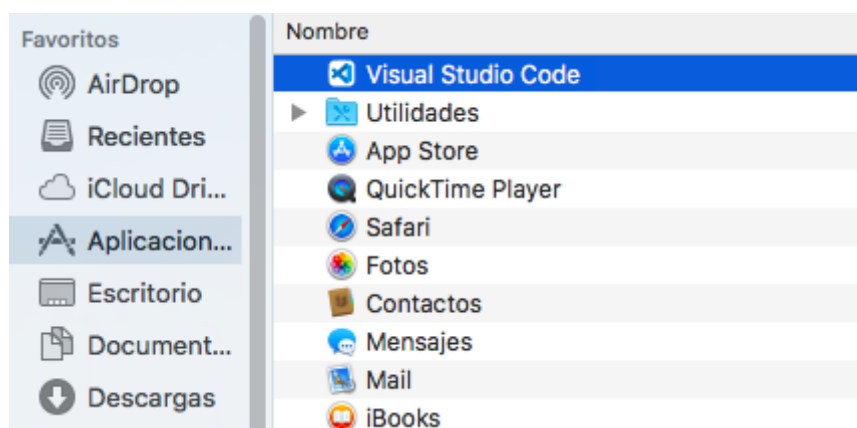
Windows

Para Windows descargamos el instalador y seguimos los pasos. No hay mucho que configurar; en todo caso, podemos dejar marcada la casilla para añadir el menú contextual "Abrir con Code" para poder abrir archivos y carpetas con VS Code desde el explorador de archivos directamente, con un clic derecho.



Mac OSX

Para **Mac OSX**, descargamos la aplicación y la podemos ejecutar directamente. También podemos moverla a la carpeta de *Aplicaciones* para tenerla localizada.



2. Node.js

Vamos ahora a instalar la piedra angular de este curso: Node.js. Esta instalación dependerá de nuestro sistema operativo, así que vamos a ver las opciones disponibles.

Versión requerida: recomendable tener instalada la actual versión LTS (*Long Term Support*), que es la que más soporte a largo plazo va a tener. [Esta](#) es la web oficial de Node.js, donde podemos encontrar información sobre la versión actual, y el calendario de implantación de las sucesivas versiones LTS, que serán las que principalmente nos interesen.

2.1. Instalación en Linux y Mac: el gestor *nvm*

A la hora de instalar Node.js en Linux o Mac, podemos emplear el gestor de versiones **NVM** (*Node Version Manager*), una herramienta que nos permite automatizar la instalación de Node, e incluso tener distintas versiones instaladas y elegir cuál activar en cualquier momento. Podemos consultar información en su [web oficial en GitHub](#).

Para instalar NVM, debemos descargarlo con el comando `curl` o `wget`, según se explica en la propia web de GitHub. Si optamos por `wget`, el comando es como sigue (en una sola línea):

```
wget -qO- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```

En el caso de no disponer del comando `wget` instalado, podemos o bien instalarlo, o bien emplear este otro comando equivalente, con la orden `curl` (también en una sola línea):

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.1/install.sh | bash
```

NOTA: el número de versión `v0.39.1` puede variar. Es preferible consultar la web de GitHub para obtener el comando actualizado.

NOTA: después de ejecutar el comando anterior, será necesario cerrar el terminal y volverlo a abrir para poder utilizar el comando `nvm`.

Ya tenemos `nvm` instalado en el sistema. Aquí mostramos algunas de las opciones más interesantes que podemos utilizar:

- `nvm install node` : instala la última versión disponible de Node
- `nvm install --lts` : instala la última versión LTS disponible (opción recomendada para este curso)
- `nvm install 12.16.0` : instala la versión especificada de Node
- `nvm uninstall 12.16.0` : desinstala la versión especificada de Node
- `nvm ls-remote` : muestra todas las versiones disponibles para instalar
- `nvm list` : muestra todas las versiones instaladas localmente
- `nvm current` : muestra la versión actualmente activa
- `nvm use 12.16.0` : marca la versión indicada como actualmente activa
- `nvm use --lts` : marca como activa la última versión LTS instalada

En nuestro caso, vamos a instalar la última versión LTS disponible, por lo que ejecutaremos el comando:

```
nvm install --lts
```

Podemos ejecutar ahora `node -v` en el terminal y comprobar que nos muestra el número de versión adecuado.

2.2. Instalación en Windows

Para instalar Node.js en Windows, no disponemos del mismo gestor *nvm* que en Linux o Mac. Como alternativa, existe alguna implementación paralela de *nvm* que podemos hacer servir, como [esta](#). Podemos descargar un instalador (*nvm-setup.zip*) y ejecutarlo para instalar este gestor. Después, desde línea de comandos tendremos disponibles, entre otras, estas opciones:

- `nvm install 16.13.2` : instala la versión especificada de Node
- `nvm uninstall 16.13.2` : desinstala la versión especificada de Node
- `nvm list` : muestra todas las versiones instaladas localmente
- `nvm list available` : muestra todas las versiones disponibles para instalar con esta adaptación de NVM.
- `nvm use 16.13.2` : marca como activa la versión de Node especificada (previamente instalada).

También desde la propia [web de Node](#) tenemos disponible un instalador para Windows, que permite instalar el framework, aunque sin la posibilidad de tener distintas versiones coexistiendo, como ocurre con NVM.

3. MariaDB/MySQL

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.

3.1. Instalación en Linux

Para instalar el servidor en nuestra máquina virtual Ubuntu o un sistema 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 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 establecer el nivel de seguridad de la contraseña. Si elegimos que sí queremos establecerlo, deberemos elegir el nivel de seguridad de 0 (*LOW*) a 2 (*STRONG*), y dependiendo de ello, tendremos que elegir una contraseña con más o menos requisitos.

```
Securing the MySQL server deployment.
```

```
Connecting to MySQL using a blank password.
```

```
VALIDATE PASSWORD COMPONENT can be used to test passwords  
and improve security. It checks the strength of password  
and allows the users to set only those passwords which are  
secure enough. Would you like to setup VALIDATE PASSWORD component?
```

```
Press y|Y for Yes, any other key for No: Y
```

```
There are three levels of password validation policy:
```

```
LOW      Length >= 8
```

```
MEDIUM Length >= 8, numeric, mixed case, and special characters
```

```
STRONG Length >= 8, numeric, mixed case, special characters and dictionary file
```

```
Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG:
```

```
0
```

NOTA: en la máquina virtual que se os proporciona hemos configurado un usuario llamado **alumno** con contraseña **alumno1234**, que será el que podréis utilizar para conectar a las bases de datos.

3.2. 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.

4. MongoDB

MongoDB es, a día de hoy, el servidor de bases de datos No-SQL más popular que existe. Veremos más adelante algunas nociones sobre bases de datos No-SQL, pero de momento vamos a dejar instalado el servidor, de forma similar a como hemos hecho antes con MySQL.

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.

4.1. Instalación en Linux

Para dejar instalado MongoDB en nuestra máquina virtual o distribuciones similares, ejecutamos estos comandos desde terminal. Podéis consultar en la [web oficial](#) de MongoDB los comandos actualizados a otra versión si lo preferís. En este caso se muestran los pasos para instalar la versión 5.0 en Ubuntu 20.04 (cada línea separada se debe escribir de forma continua aunque aquí se muestre dividida en varias partes):

```
wget -qO - https://www.mongodb.org/static/pgp/server-5.0.asc | sudo apt-key add -  
  
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu  
focal/mongodb-org/5.0 multiverse" |  
sudo tee /etc/apt/sources.list.d/mongodb-org-5.0.list  
  
sudo apt update  
  
sudo apt install mongodb-org
```

Igual que ocurre con el servidor MySQL, podemos utilizar estos comandos para conocer el estado actual del servidor, pararlo, iniciarlo o reiniciarlo, respectivamente:

```
sudo systemctl status mongod  
sudo systemctl stop mongod  
sudo systemctl start mongod  
sudo systemctl restart mongod
```

Para permitir que MongoDB se inicie automáticamente en cada reinicio del sistema, ejecutamos este comando (ya lo hemos dejado ejecutado en la máquina virtual con todo instalado):

```
sudo systemctl enable mongod
```

4.2. Instalación en otros sistemas

Nuevamente, si preferís instalar MongoDB en otros sistemas, aquí os dejamos algunas guías para hacerlo, ya que estos contenidos no forman parte del curso en sí:

- [Windows](#)
- [MacOSX](#)

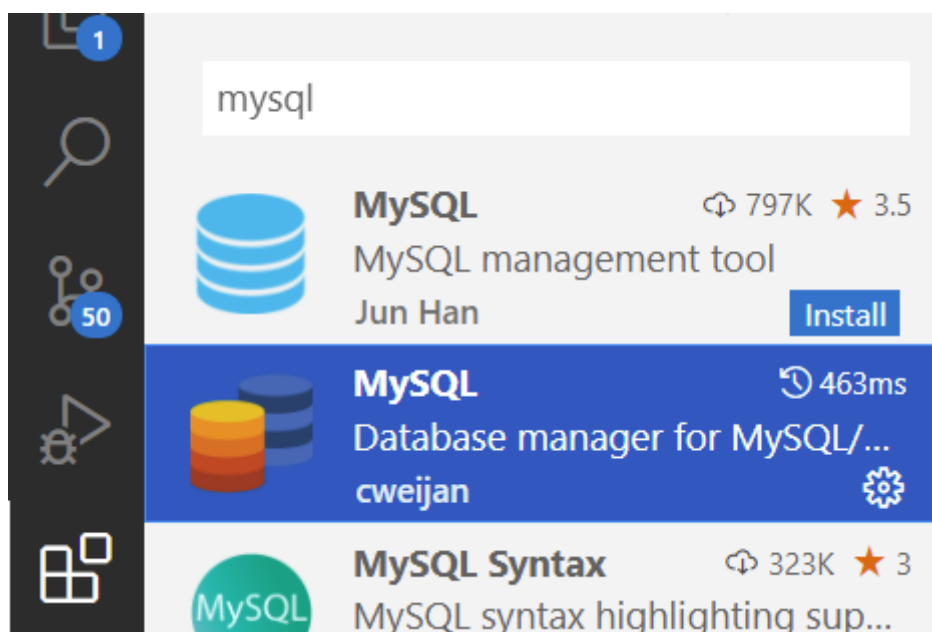
5. Extensiones de Visual Studio Code

En pasos anteriores hemos instalado el IDE Visual Studio Code, que será el que utilizaremos en el curso para desarrollar nuestros proyectos. Entre sus muchas ventajas, incluye el hecho de poder instalar ciertos *plugins* adicionales que facilitan el trabajo con determinadas herramientas o lenguajes. Estos plugins o **extensiones** son muy variados.

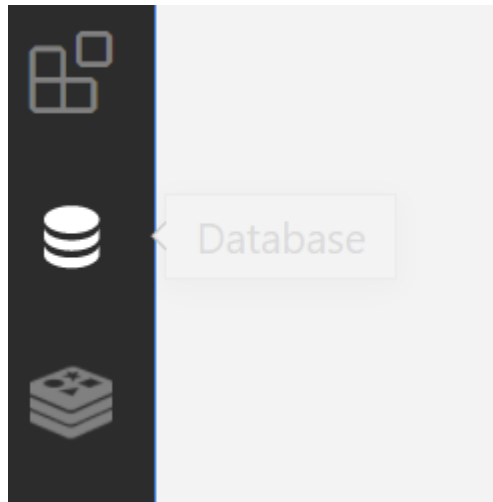
En nuestro caso, vamos a instalar dos extensiones (vamos al panel *Extensiones* en la barra izquierda de Visual Studio Code y las buscamos).

MySQL (desarrollado por cweijan)

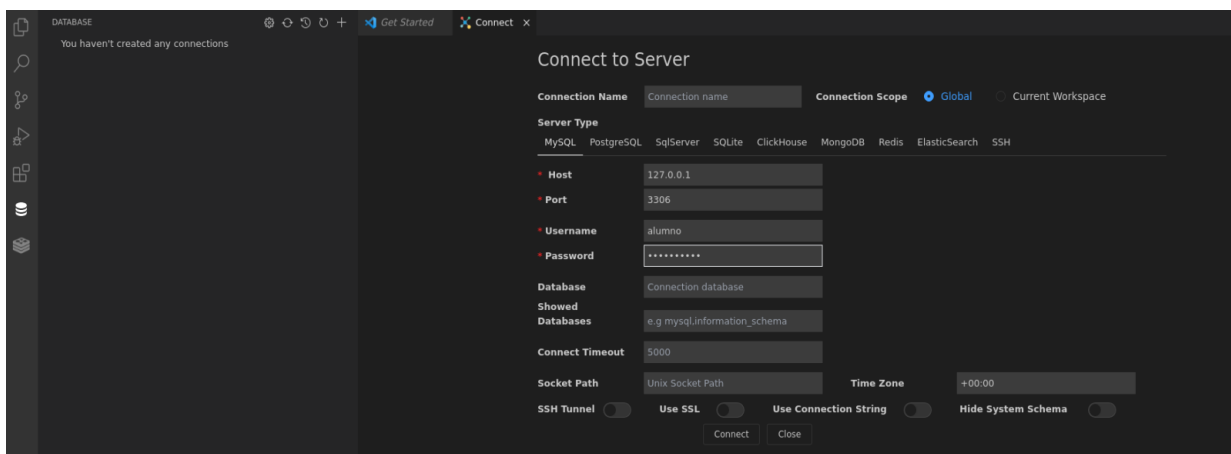
Esta extensión nos va a permitir conectar y gestionar bases de datos MySQL y MongoDB desde el propio IDE.



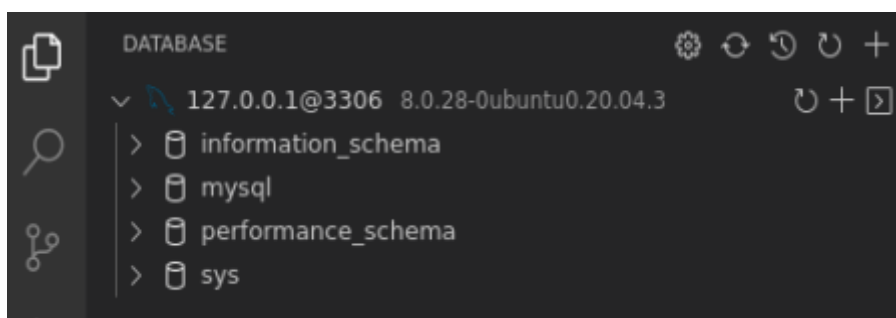
Una vez la tengamos instalada, veremos que en la barra izquierda aparecen dos iconos nuevos, para gestionar bases de datos relacionales y No-SQL, respectivamente:



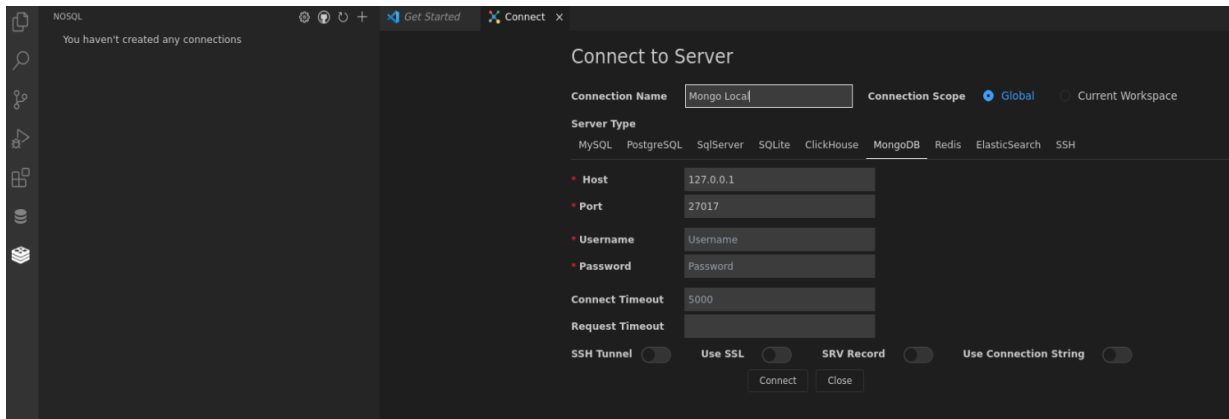
Tenemos que hacer clic en cada una de ellas y crear la conexión a cada base de datos (icono de + en la parte superior). Nos rellenará ya los campos con los parámetros por defecto, y sólo tenemos que poner un nombre a cada conexión. Así podría quedar la de MySQL (usamos el usuario *alumno* con la contraseña *alumno1234* que hemos dicho antes, en el caso de nuestra máquina virtual con todo el software instalado):



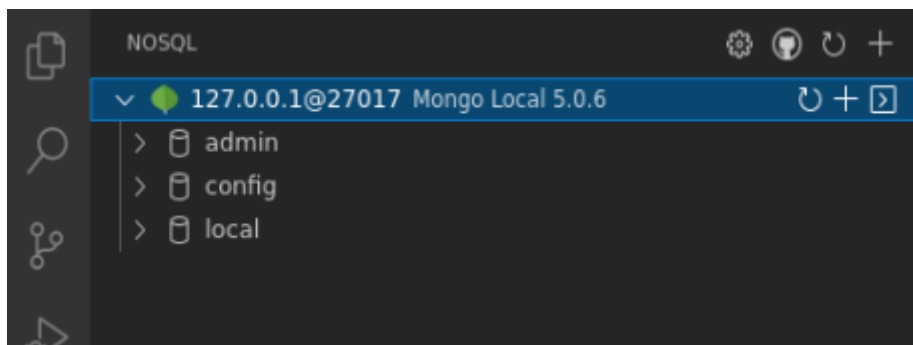
Tras conectar, podremos ver en el panel izquierdo las bases de datos existentes:



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



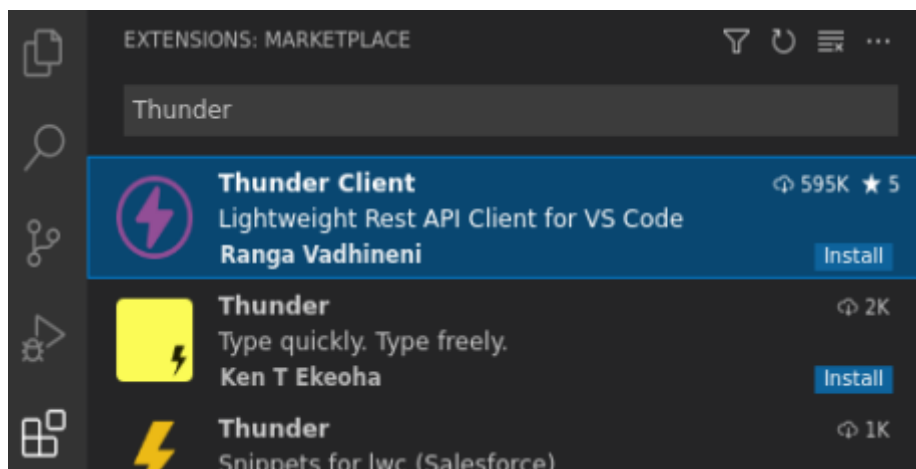
Y nuevamente, podremos ver a la izquierda las bases de datos existentes:



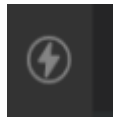
NOTA: en la máquina virtual con todo el software instalado ya se os dejan configuradas ambas conexiones (a MySQL y a MongoDB) desde esta extensión de Visual Studio Code.

ThunderClient

Esta herramienta nos servirá para simular peticiones a servidores web, y recoger y analizar la respuesta. La emplearemos para probar los servicios REST que desarrollaremos en algunas sesiones. La buscamos en el panel de extensiones y la instalamos:



Nos aparecerá un icono en el panel izquierdo desde el que gestionaremos las conexiones y peticiones (aprenderemos a utilizarlo más adelante):



Alternativas

Como alternativas a estas extensiones también se pueden usar, respectivamente, estos tres programas (aunque no se proporcionan instalados en la máquina virtual):

- **MySQL Workbench:** para gestionar bases de datos MySQL ([web oficial](#))
- **Robo3T:** para gestionar bases de datos MongoDB ([web oficial](#))
- **Postman:** para simular peticiones a servidores ([web oficial](#))

6. Recursos adicionales

Para facilitar la labor de instalar y/o trabajar con el software propuesto, se proporcionan dos máquinas virtuales basadas en sistemas Ubuntu (versión 20). En ambos casos, se cuenta con un usuario *alumno* con password *alumno*.

- [Máquina virtual base](#)
- [Máquina virtual con software instalado](#)

Es necesario utilizar el programa [VirtualBox](#) para ejecutar estas máquinas virtuales. También se puede cambiar la configuración por defecto de dichas máquinas virtuales, para darles más memoria RAM o núcleos de procesador, si se cree conveniente.