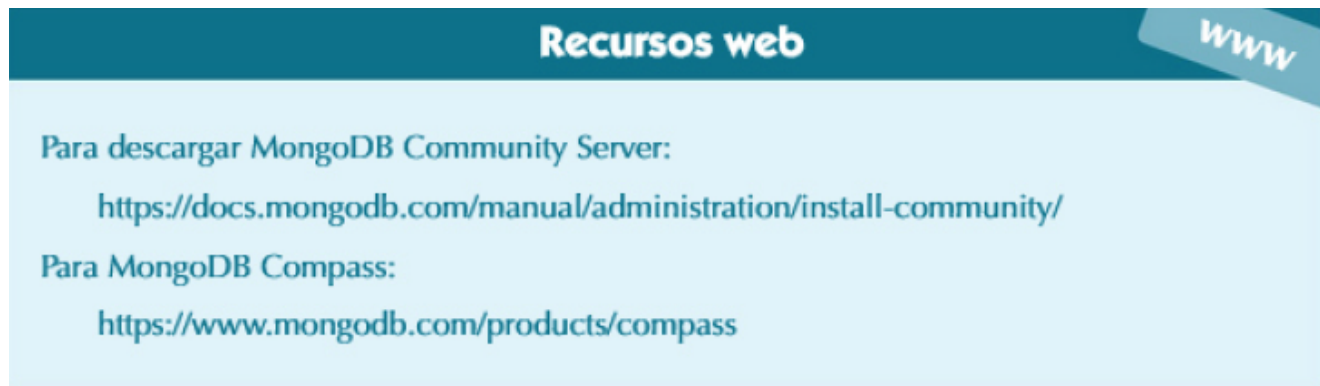


**DWES DAW**  
**TEMA 3\_1**  
**BASES DE DATOS NO RELACIONALES. MONGODB**

## INSTALACIÓN Y PUESTA MARCHA DE MONGODB EN WINDOWS

Hay que **instalar**:

- *MongoDB Community Server*, la versión libre de MongoDB.  
<https://www.mongodb.com/try/download/community>
- *MongoDB Compass*, una aplicación para manejar la base de datos con **interfaz gráfica**.  
<https://www.mongodb.com/try/download/compass>
- El *driver* de PHP para MongoDB.

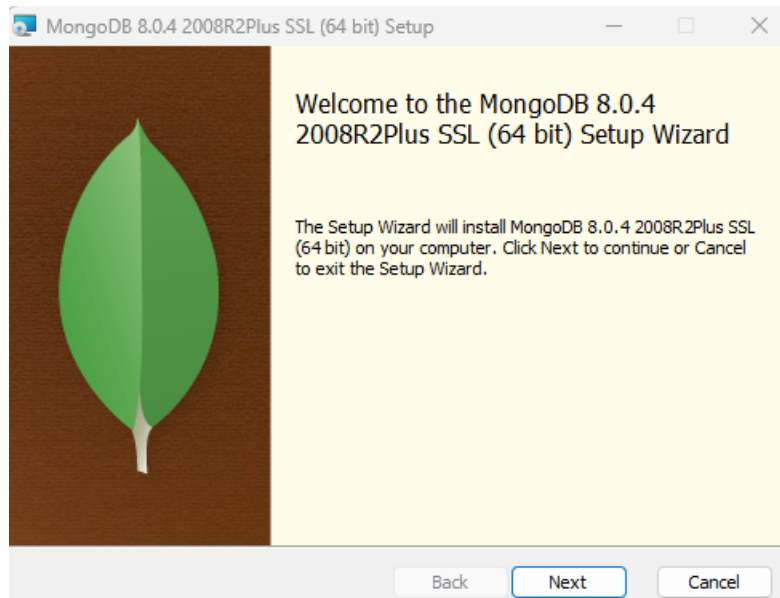


Para empezar, hay que instalar el **servidor MongoDB**.

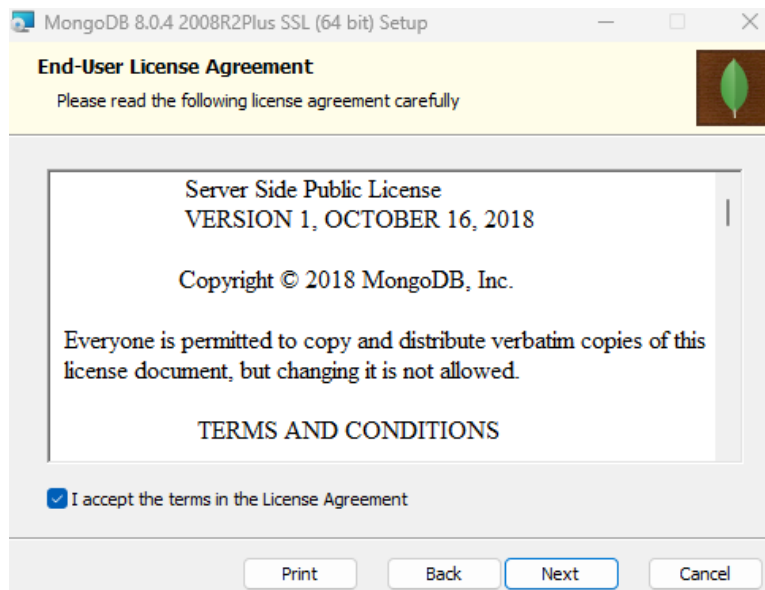
En la configuración de la instalación, por defecto:

- El servidor escucha en el puerto 27017
- La ruta de instalación en Windows es C:\Program Files\MongoDB
- No es necesario utilizar usuario y contraseña para conectarse

## INSTALACIÓN DE MONGODB



## NEXT



NEXT

Elegimos instalación COMPLETE

MongoDB 8.0.4 2008R2Plus SSL (64 bit) Service Customization

**Service Configuration**  
Specify optional settings to configure MongoDB as a service.

☒ Install MongoDB as a Service

☒ Run service as Network Service user

☐ Run service as a local or domain user:

Account Domain:

Account Name:

Account Password:

Service Name:

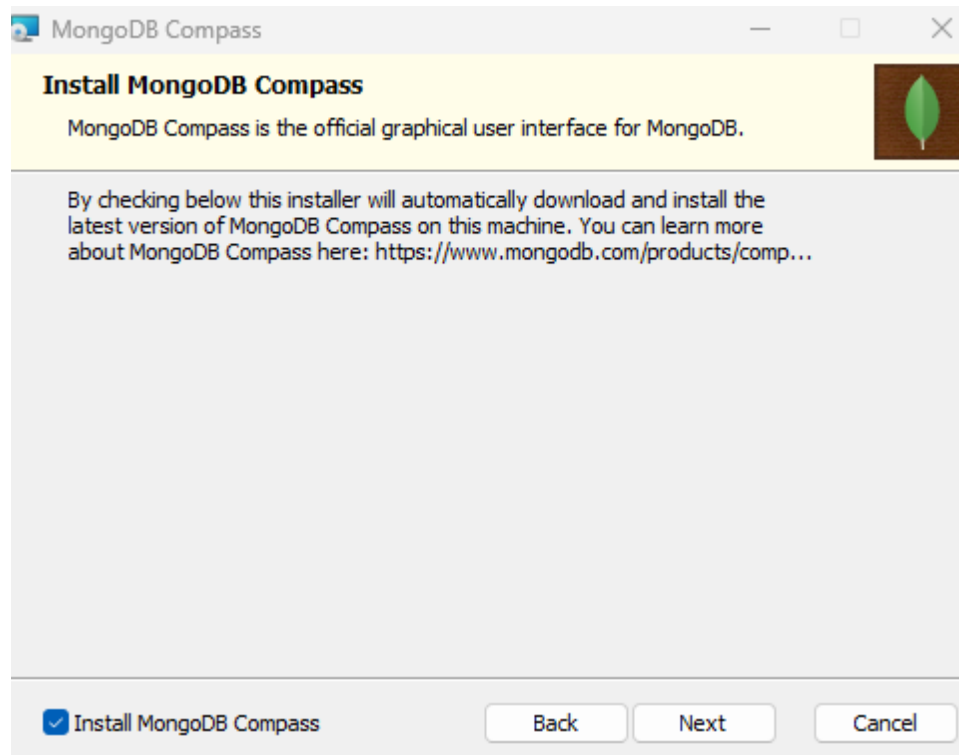
Data Directory:

Log Directory:

< Back   Next >   Cancel

Dejamos las opciones por defecto, es decir, que se instala MongoDB como un Servicio

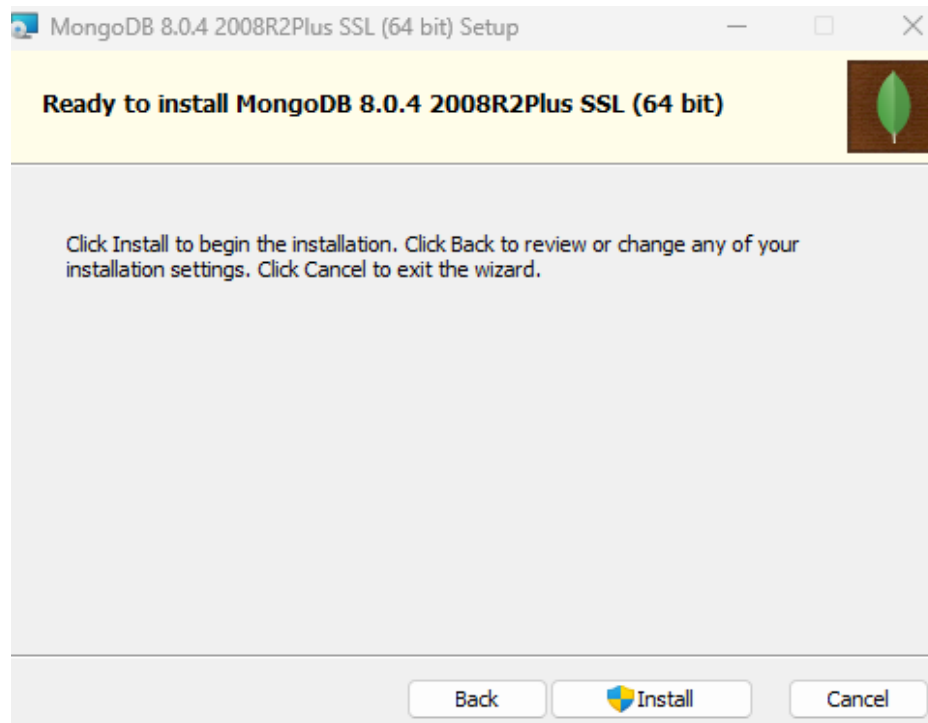
NEXT



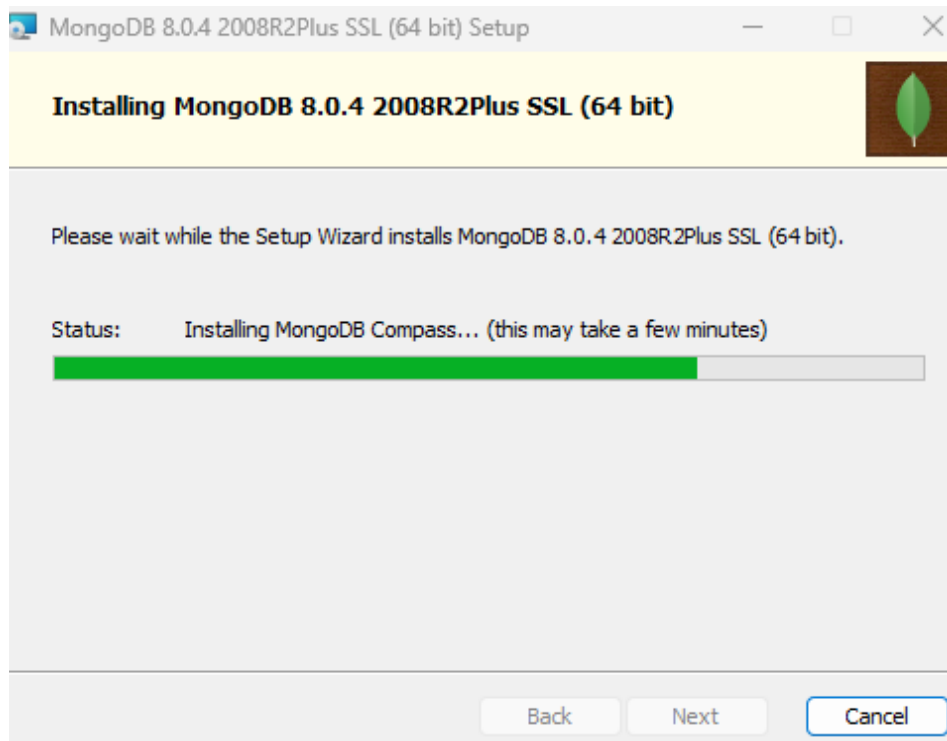
NEXT

Permite que se instale también MONGODB COMPASS

Decimos que Sí

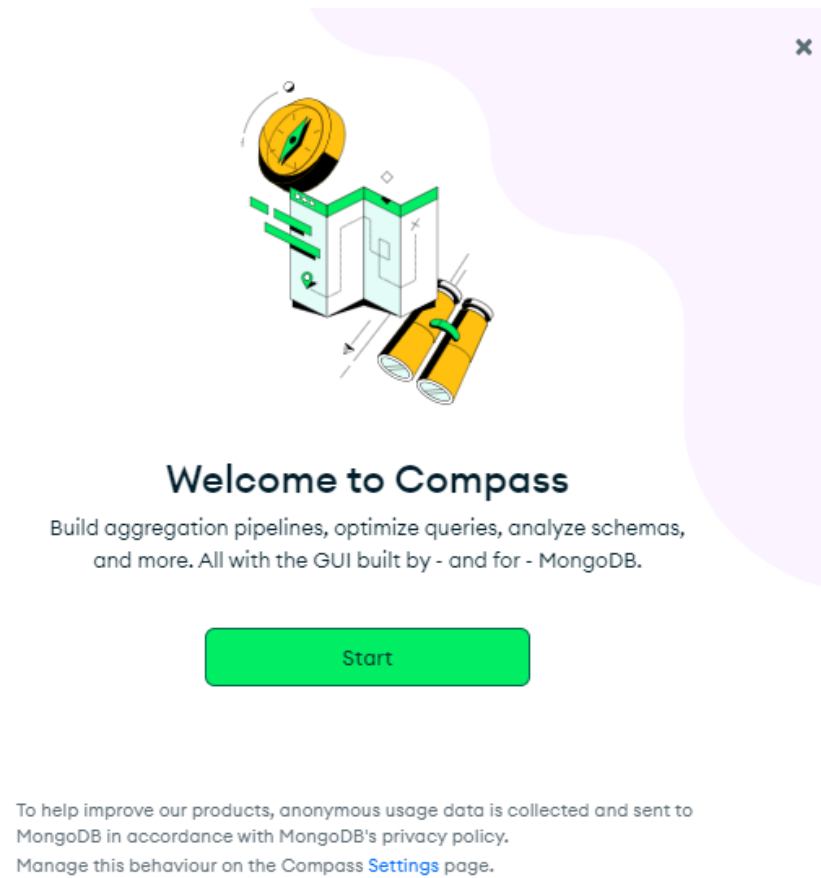


INSTALL



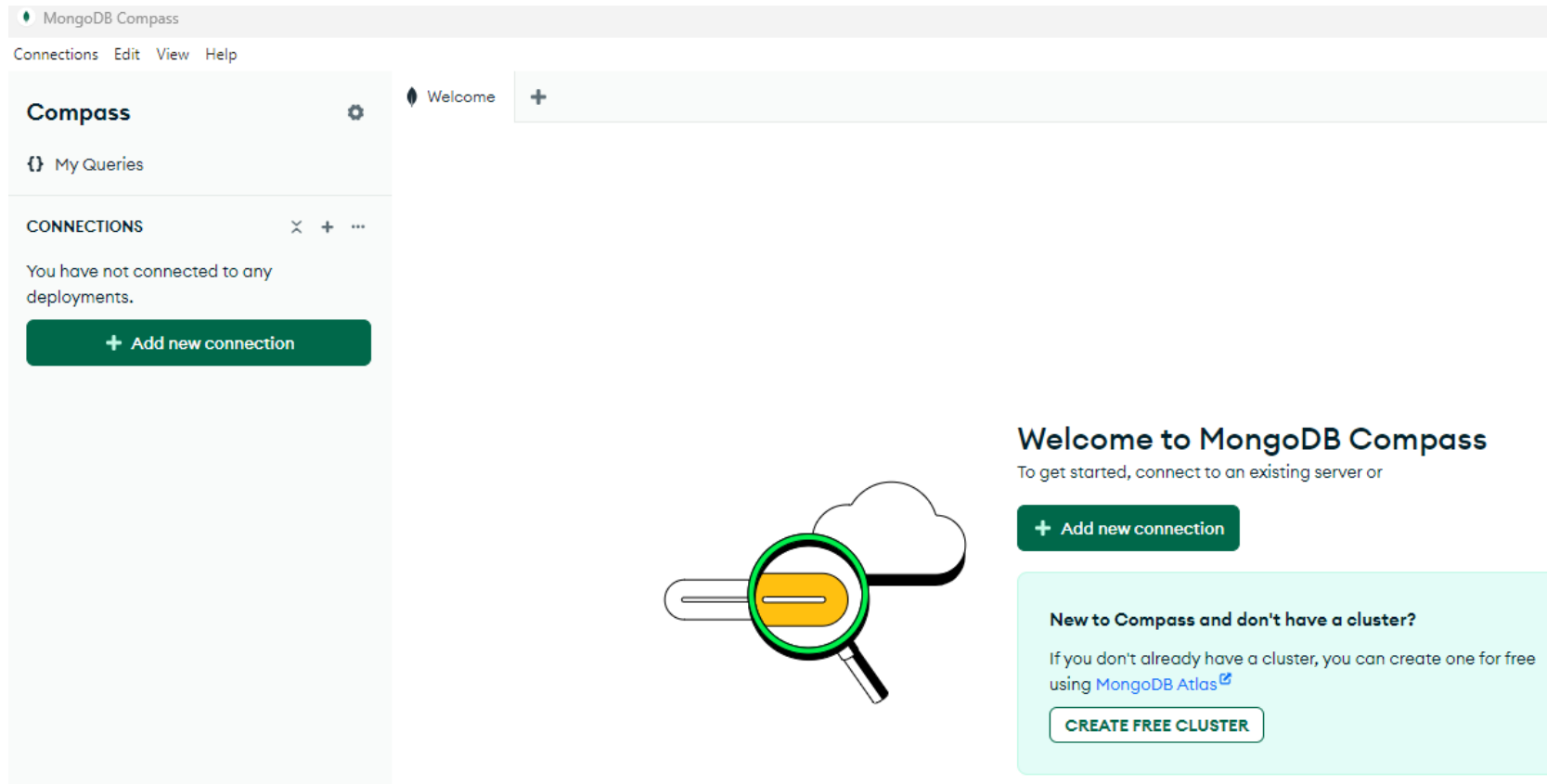
NEXT

FINISH



Ya está instalado.

Ahora habría que seleccionar START para arrancar el servicio MONGODB






Pulsamos *Add-New Connection*

## New Connection

Manage your connection settings

URI ⓘ

Edit Connection String 

mongodb://localhost:27017/

Name

Color

No Color ▼

☐ Favorite this connection

Favoriting a connection will pin it to the top of your list of connections

➤ Advanced Connection Options


Cancel

Save


Connect

Save & Connect

**How do I find my connection string in Atlas?**

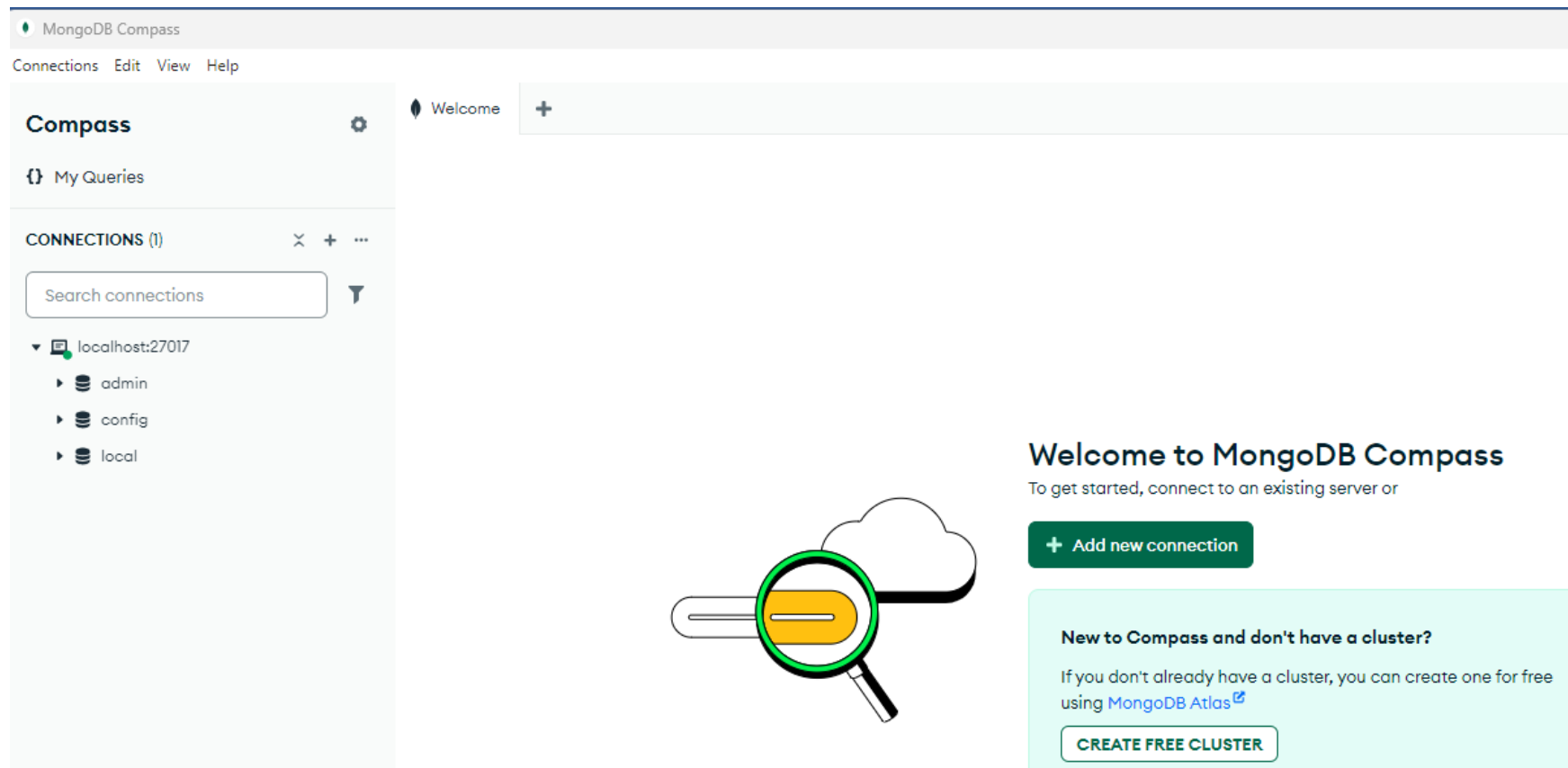
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.  
[See example](#) 

**How do I format my connection string?**

[See example](#) 

Pulsamos **Save & Connect**

Vemos:



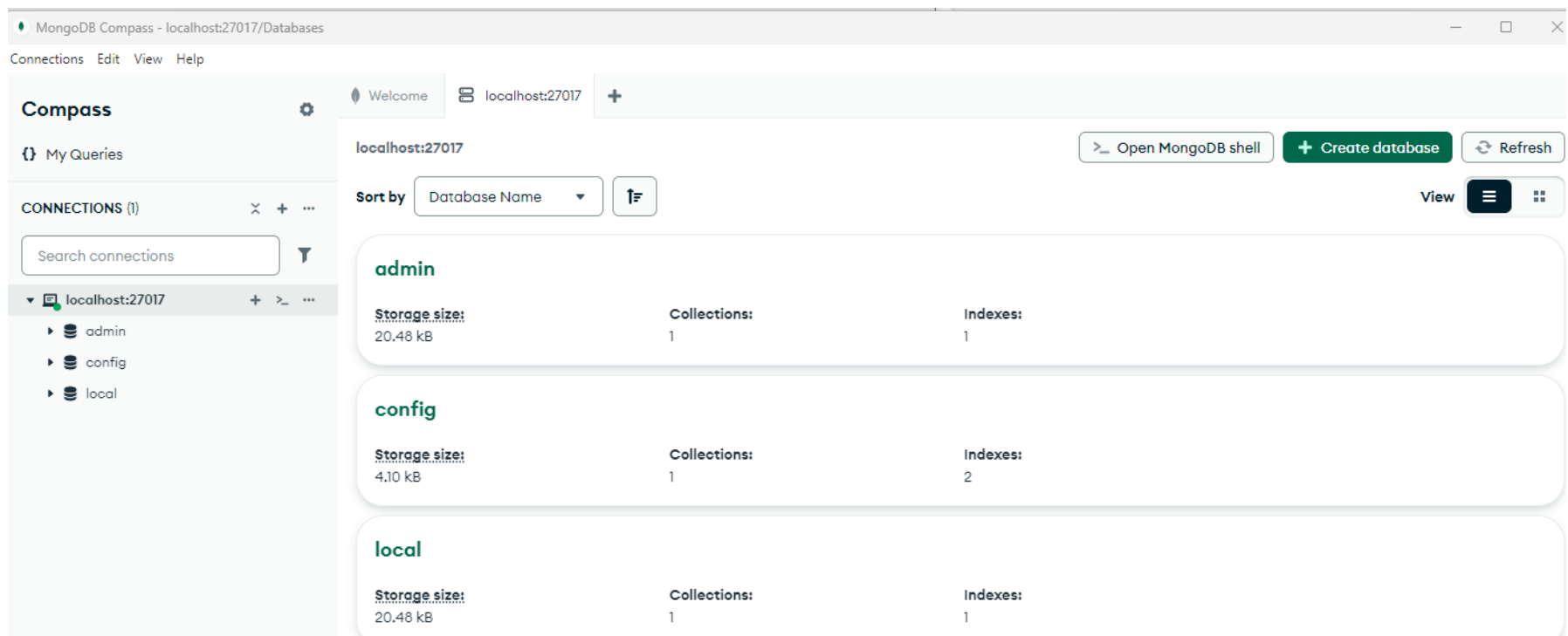
## YA INSTALADO MONGODB

Una vez instalado, el servidor se pone en marcha ejecutando **mongod.exe**, que se encuentra en el subdirectorio **bin** dentro de la ruta de instalación del programa (en Windows, la ruta por defecto es C:\Program Files\MongoDB\3.6\bin).

A continuación, se instala **MongoDB Compass**, que ofrece una interfaz gráfica para manejar el servidor. Al arrancar la aplicación, lo primero que hay que hacer es conectarse al servidor. Si se ha seguido la configuración por defecto, no hace falta cambiar nada, solo pulsar el botón **Connect**, en la esquina inferior derecha.

Al conectar, la aplicación muestra las bases de datos del servidor (llamado **MyCluster**).

Hay 3 bases de datos ya creadas: **admin**, **config** y **local**.



## CREAR BD MONGODB

Para crear una nueva se usa el botón *CREATE DATABASE*

Al pulsarlo, se abre una ventana que pide el nombre de la nueva base de datos y también el de una colección. Es obligatorio introducir ambos campos.

Para poder utilizar los ejemplos posteriores hay que crear una base datos llamada **libroservidor** y una colección llamada **usuarios**.

Al volver a la pantalla principal aparecerá la nueva base de datos, y si se pulsa sobre su nombre se verá también la colección **usuarios**.

MongoDB Compass - localhost:27017/libroservidor

Compass

My Queries

CONNECTIONS (10)

Search connections

- localhost:27017
  - admin
  - config
  - libroservidor**
    - localidades
    - prueba
    - usuarios
  - local
  - userblogdb
- localhost:27017
- localhost:27017

libroservidor +

localhost:27017 > libroservidor

Sort by Collection Name

localidades			
Storage size:	Documents:	Avg. document size:	Indexes:
20.48 kB	1	732.00 B	1
prueba			
Storage size:	Documents:	Avg. document size:	Indexes:
20.48 kB	4	92.00 B	1
usuarios			
Storage size:	Documents:	Avg. document size:	Indexes:
20.48 kB	3	104.00 B	1

## INSTALACIÓN DE COMPOSER

*Composer* es un gestor de paquetes y dependencias para PHP. Sirve para descargar y gestionar las librerías y dependencias de un proyecto PHP de manera automatizada.

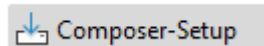
Cuando se desarrolla una aplicación o sitio web en PHP, a menudo se utilizan diferentes librerías o frameworks de terceros para añadir funcionalidades adicionales. Composer permite gestionar estas dependencias de manera sencilla, descargándolas y manteniéndolas actualizadas de manera automática.

Una vez instalado Composer. Tras usarlo en nuestro proyecto, se crea un archivo llamado "composer.json" en el raíz del proyecto, en el que se especifican las dependencias necesarias y se ejecuta el comando "composer install" para descargar las dependencias y crear un archivo "vendor/autoload.php" que se puede incluir en el código de la aplicación para cargar automáticamente las dependencias.

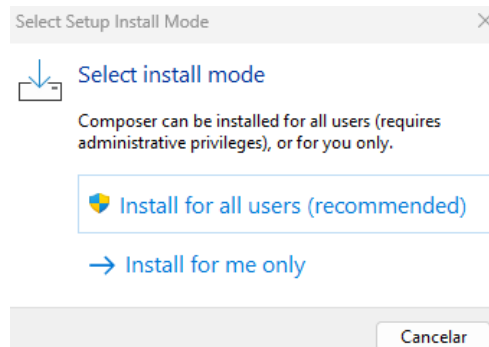
Enlace:

<https://getcomposer.org/download/>

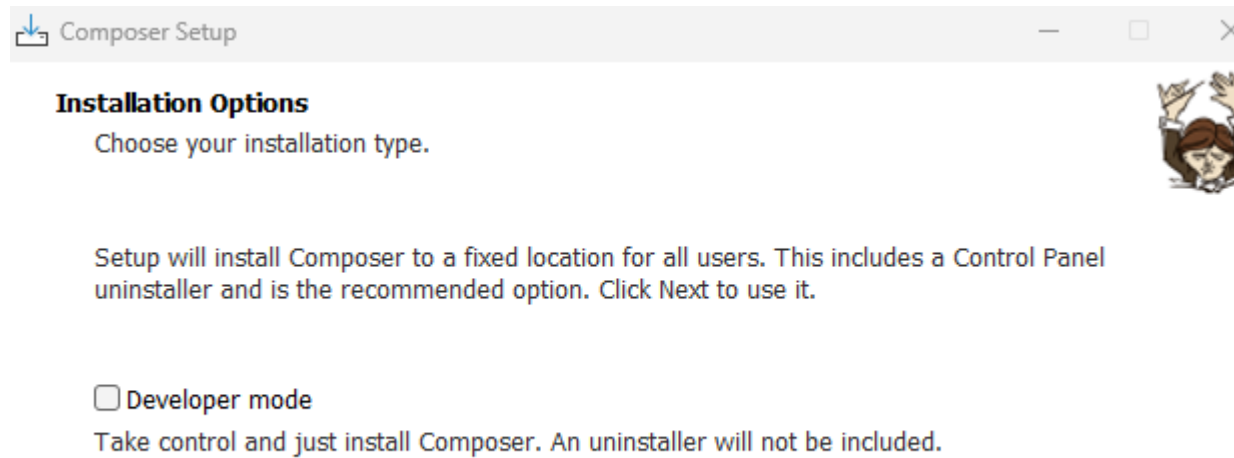
Descargamos *composer.exe* (composer Setup):



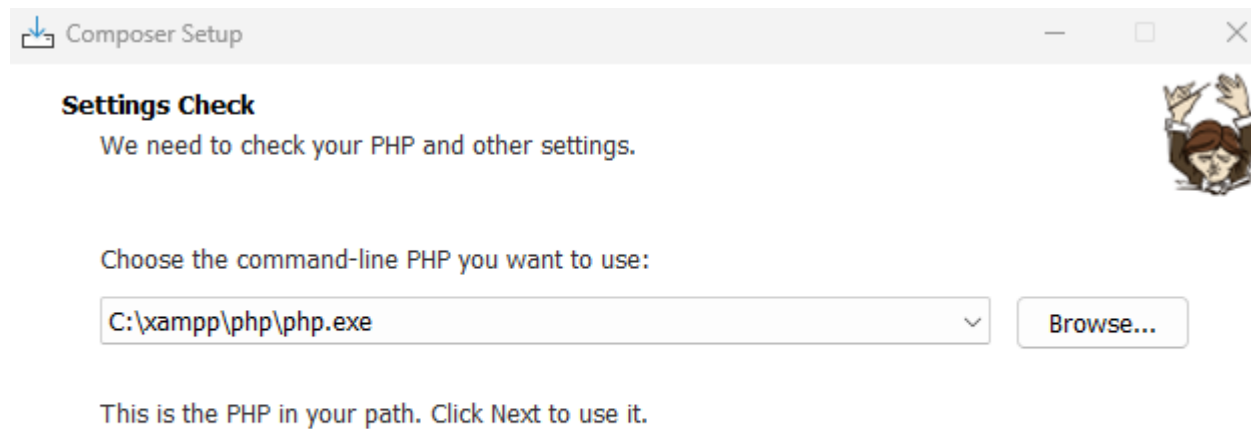
Lo ejecutamos



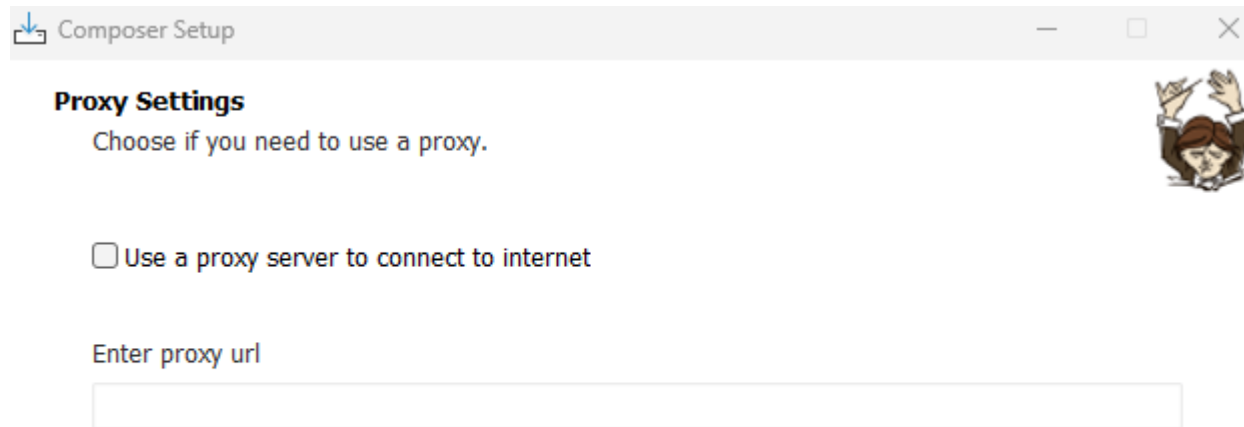
Seleccionamos instalar para todos los usuarios



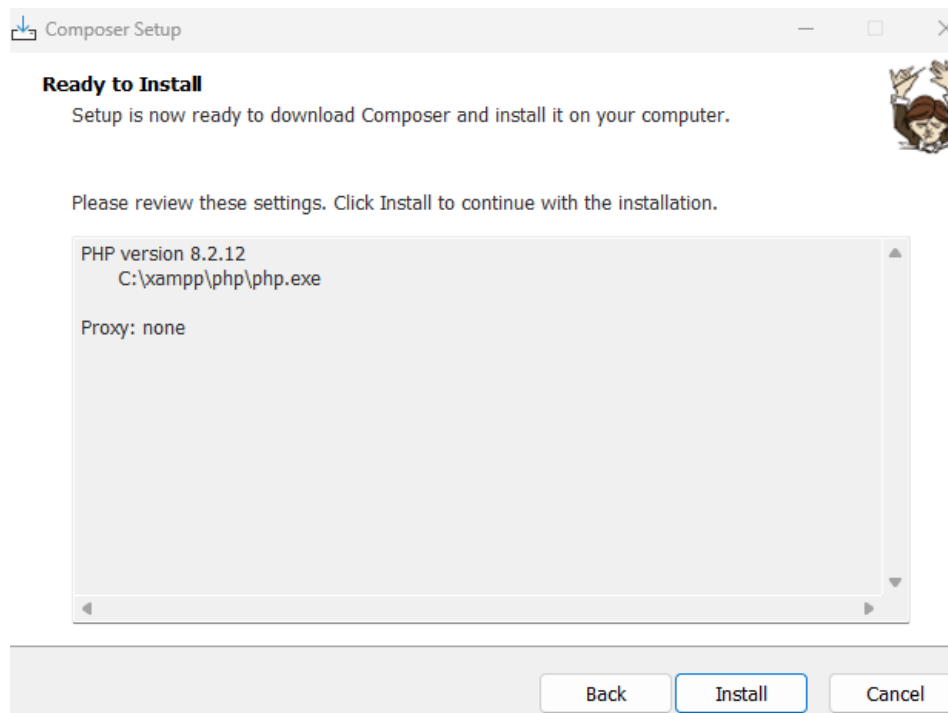
NEXT



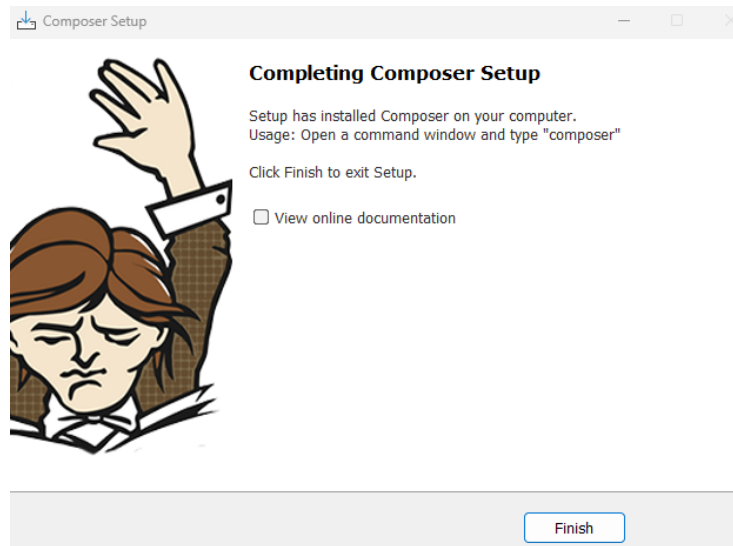
NEXT



NEXT



## INSTALL



## FINISH



## INSTALAR DRIVER MONGODB

Finalmente, para poder conectarse a un servidor MongoDB desde PHP hay que instalar el driver correspondiente, lo que se hace utilizando **composer**.

Se debe ejecutar el comando:

```
composer require mongodb/mongodb
```

## ERRORES AL INSTALAR DRIVER MONGODB

Si al introducir comando

```
composer require mongodb/mongodb
```

Me da error:

```
C:\Users\Administrador>composer require mongodb/mongodb
Cannot use mongodb/mongodb's latest version 1.20.0 as it requires ext-mongodb ^1.20.0 which is missing from your platform.
Cannot use mongodb/mongodb 1.7.2 as it requires php ^7.0 which is not satisfied by your platform.
Cannot use mongodb/mongodb 0.1.0 as it requires ext-phongo >=0.1.2 which is missing from your platform.
```

Hago lo siguiente:

Compruebo mi versión de php:

```
php -v
```

Me devuelve:

PHP 8.2.12

Voy a página:

<https://pecl.php.net/package/mongodb/1.20.1/windows>

Descargo el archivo dll [8.2 Thread Safe \(TS\) x64](#) :

Copio este archivo a la ruta de php (C/xampp/php/ext)

Modifico archivo *php.ini* (está en la ruta a C/xampp/php/ ).

Añado al final la línea:

```
; HABILITAR MONGODB  
extension=mongodb
```

Bajo el servicio apache

Compruebo que ahora está mongo, tecleando:

```
php -m | findstr mongodb
```

si veo:

```
C:\Users\Administrador>php -m | findstr mongodb  
mongodb
```

Es que ya está driver de *mongodb* instalado

## PRUEBA DE MONGODB DESDE NETBEANS

Crea proyecto php en netbeans

Crea archivo *pruebaMongo.php*

Añade en el código *phpinfo()*;

Comprobamos que se ha instalado el driver mongodb, ya que busco la cadena mongodb y veo

**mongodb**

MongoDB support	enabled
MongoDB extension version	1.20.1
MongoDB extension stability	stable
libbson bundled version	1.28.1
libmongoc bundled version	1.28.1
libmongoc SSL	enabled
libmongoc SSL library	OpenSSL
libmongoc crypto	enabled
libmongoc crypto library	libcrypto

Abre un terminal en dicho proyecto

Ejecuto comando:

*composer require mongodb/mongodb*

```
d:\PROJECTS_PHP\PHP3>composer require mongodb/mongodb
```

## CONEXIÓN A MONGODB DESDE CONSOLA

Para conectarse desde CONSOLA a MONGODB, escribimos en el terminal:

***mongosh***

Ya estamos dentro de MongoDB, vemos:

```
admin@MacBook-Air-de-Admin T3_AplicWebPHP % mongosh
Current Mongosh Log ID: 66b5ef382c624871aa353a7e
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutM
S=2000&appName=mongosh+2.2.12
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017
admin@MacBook-Air-de-Admin T3_AplicWebPHP % brew services restart mongodb-community
=> Successfully started `mongodb-community` (label: homebrew.mxcl.mongodb-community)
admin@MacBook-Air-de-Admin T3_AplicWebPHP % mongosh
Current Mongosh Log ID: 66b5ef4714102e601cc48664
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutM
S=2000&appName=mongosh+2.2.12
Using MongoDB:      7.0.8
Using Mongosh:      2.2.12

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
```

---

**The server generated these startup warnings when booting**  
2024-08-09T12:28:20.602+02:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

```
test> █
```

## COMANDOS PARA INTERACTUAR CON MONGODB DESDE CONSOLA

Podemos interactuar con Mongo desde un terminal o desde un entorno gráfico, como por ejemplo *Compass*

En este apartado vamos a hacerlo todo desde consola.

<https://geekflare.com/es/mongodb-queries-examples/>

Para ver los comandos disponibles:

- *help*
- *db.help*

Para ver las bases de datos que se han creado escribo:

*show dbs*

**Nota** : show va en minúsculas

Veo (inicialmente):

```
[> show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
> █
```

En mi caso veo, pues ya creé una nueva colección:

```
test> show dbs;
admin      40.00 KiB
config     60.00 KiB
libroserver 112.00 KiB
local      144.00 KiB
```

**Para usar una Colección:**

*use libroservidor*

Veo:

```
test> use libroservidor
switched to db libroservidor
libroservidor> 
```

**Para ver estadísticas:**

*db.stats*

```
libroservidor> db.stats
[Function: stats] AsyncFunction {
  apiVersions: [ 0, 0 ],
  returnsPromise: true,
  serverVersions: [ '0.0.0', '999.999.999' ],
  topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],
  returnType: { type: 'unknown', attributes: {} },
  deprecated: false,
  platforms: [ 'Compass', 'Browser', 'CLI' ],
  isDirectShellCommand: false,
  acceptsRawInput: false,
  shellCommandCompleter: undefined,
  help: [Function (anonymous)] Help
}
```

**Para consultar en qué colección estamos:**

*db*

```
libroservidor> db
libroservidor
```

## CREAR UNA COLECCIÓN DESDE CONSOLA

```
db.createCollection(Nombre,Opciones)
```

ejemplo:

```
db.createCollection(prueba)
```

Vemos que se ha creado

```
show collections
```

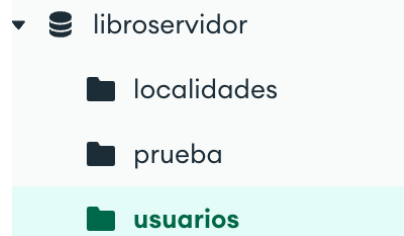
```
libroservidor> show collections
```

```
localidades
```

```
prueba
```

```
usuarios
```

Ahora lo visualizamos desde COMPASS():



## INSERTAR EN MONGODB DESDE CONSOLA

```
db.[coleccion].insert( [documento en formato JSON] ) ;
```

ejemplo:

Insertamos en la colección PRUEBA un nuevo registro (un documento)

### inserta un registro (un documento en una colección):

```
db.prueba.insertOne ({  
  firstname: "Marta",  
  lastname: "Casas",  
  Address: "Calle Petunias 232, Ciempozuelos, Madrid, 28350"  
});
```

Vemos que se ha creado:

```
libroservidor> db.prueba.insertOne ({  
...     firstname: "Marta",  
...     lastname: "Casas",  
...     Address: "Calle Petunias 232, Ciempozuelos, Madrid, 28350"  
...     });  
{  
  acknowledged: true,  
  insertedId: ObjectId('66b5fa3e14102e601cc48665')  
}
```



**inserta varios registros (varios documentos en una colección):**

- función *insertMany*
- se usa un array, por eso hay que poner los corchetes

```
db.prueba.insertMany ([
  {
    firstname: "Nombre1",
    lastname: "Ape1",
    Address: "Dir1"
  },
  {
    firstname: "Nombre2",
    lastname: "Ape2",
    Address: "Dir2"
  },
  {
    firstname: "Nombre3",
    lastname: "Ape3",
    Address: "Dir3"
  }
]);
```

Vemos que se ha creado:

```
libroservidor> db.prueba.insertMany ([
...     {
...         firstname: "Nombre1",
...         lastname: "Ape1",
...         Address: "Dir1"
...     },
...     {
...         firstname: "Nombre2",
...         lastname: "Ape2",
...         Address: "Dir2"
...     },
...     {
...         firstname: "Nombre3",
...         lastname: "Ape3",
...         Address: "Dir3"
...     }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66b5fc1b14102e601cc48666'),
    '1': ObjectId('66b5fc1b14102e601cc48667'),
    '2': ObjectId('66b5fc1b14102e601cc48668')
  }
}
```

## VISUALIZAR REGISTROS DE UNA COLECCIÓN DESDE CONSOLA

```
db.[coleccion].find();  
db.[coleccion].find().pretty();
```

ejemplo:

```
db.prueba.find();
```

Vemos el contenido:

```
libroservidor> db.prueba.find();  
[  
  {  
    _id: ObjectId('66b5fa3e14102e601cc48665'),  
    firstname: 'Marta',  
    lastname: 'Casas',  
    Address: 'Calle Petunias 232, Ciempozuelos, Madrid, 28350'  
  },  
  {  
    _id: ObjectId('66b5fc1b14102e601cc48666'),  
    firstname: 'Nombre1',  
    lastname: 'Ape1',  
    Address: 'Dir1'  
  },  
  {  
    _id: ObjectId('66b5fc1b14102e601cc48667'),  
    firstname: 'Nombre2',  
    lastname: 'Ape2',  
    Address: 'Dir2'  
  },  
  {  
    _id: ObjectId('66b5fc1b14102e601cc48668'),  
    firstname: 'Nombre3',  
    lastname: 'Ape3',  
    Address: 'Dir3'  
  }  
]
```

## **SALIR DE MONGODB DESDE CONSOLA**

Para salir escribimos:

*quit*

*o*

*exit*

## INSTALACIÓN DE MONGO COMPASS EN WINDOWS:

**Compass** es un entorno gráfico para interactuar con las bases de datos Mongo

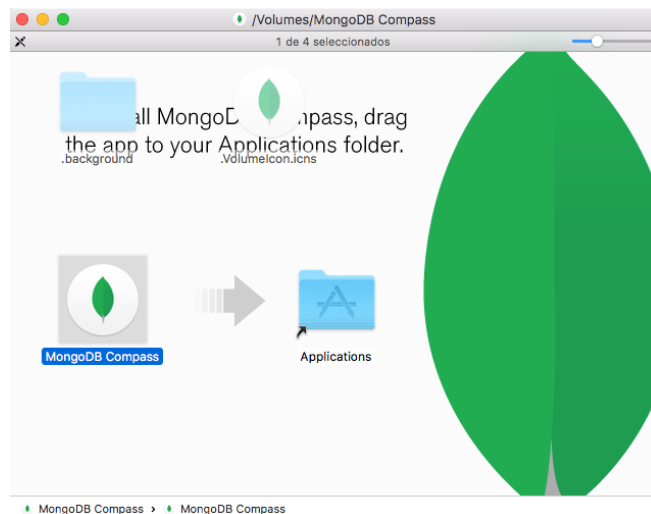
Seguimos el link:

<https://www.mongodb.com/docs/compass/current/install/>

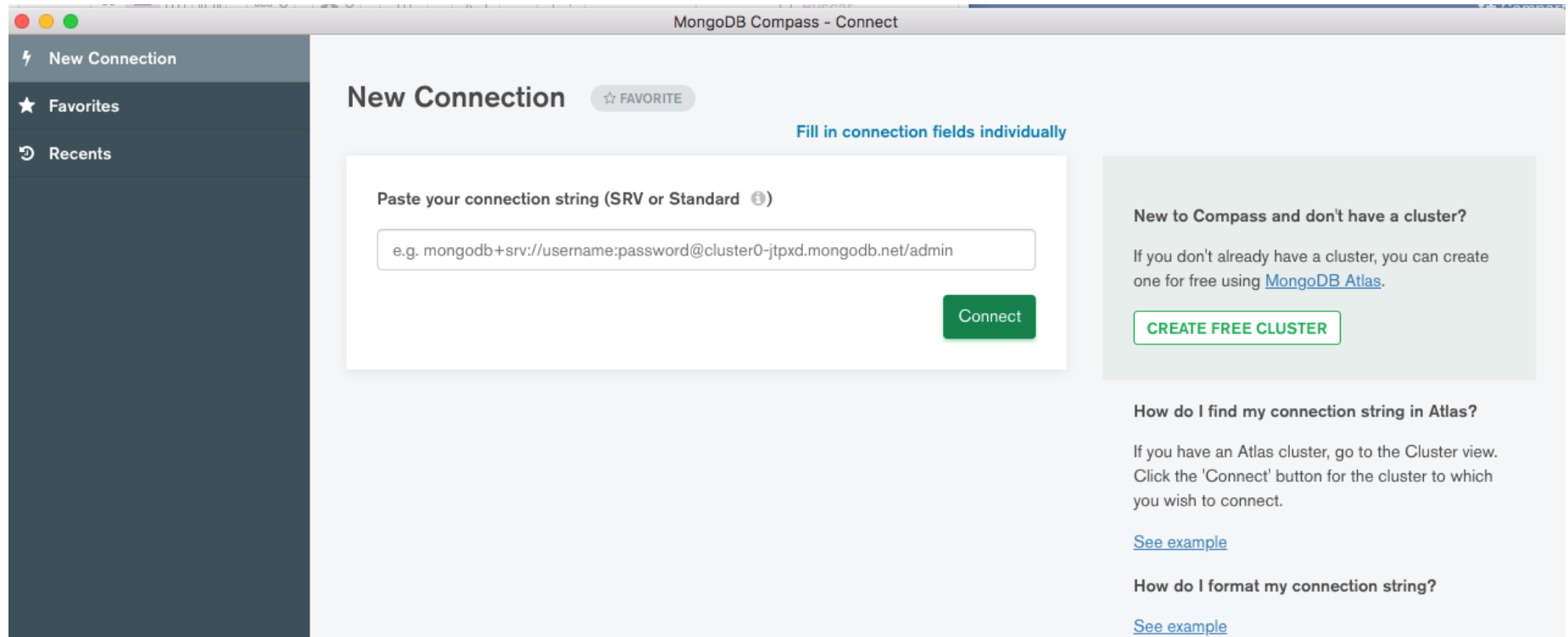
Descargo archivo versión actual. Ejemplo:

<https://downloads.mongodb.com/compass/mongosh-2.3.2-x64.msi>

Lo ejecutamos:

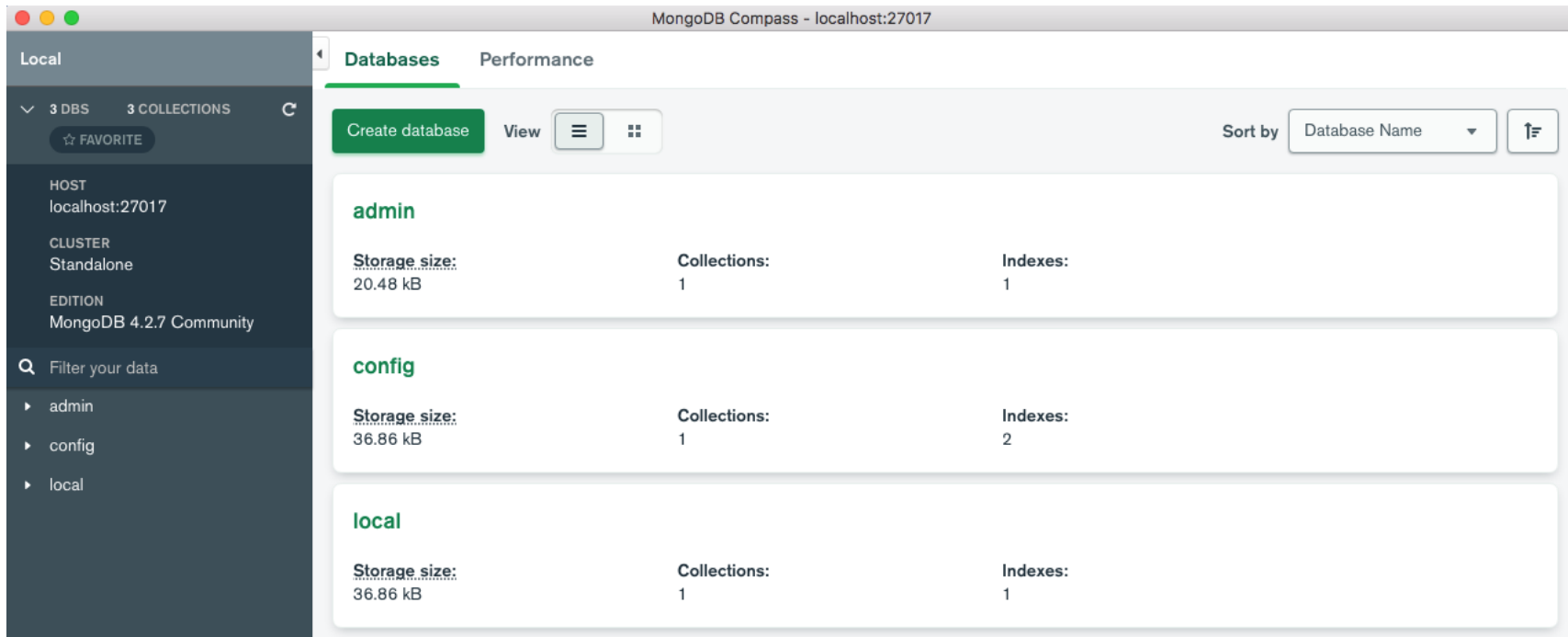


Ejecutamos la Aplicación:



## Entramos en New Connection:

Vemos que hay tres bases de datos ya existentes



Creamos nueva Base de datos, llamada *libroservidor* y una colección llamada *usuarios*:



## Create Database

Database Name

Collection Name

☐ Capped Collection

Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. ⓘ

☐ Use Custom Collation

Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. ⓘ

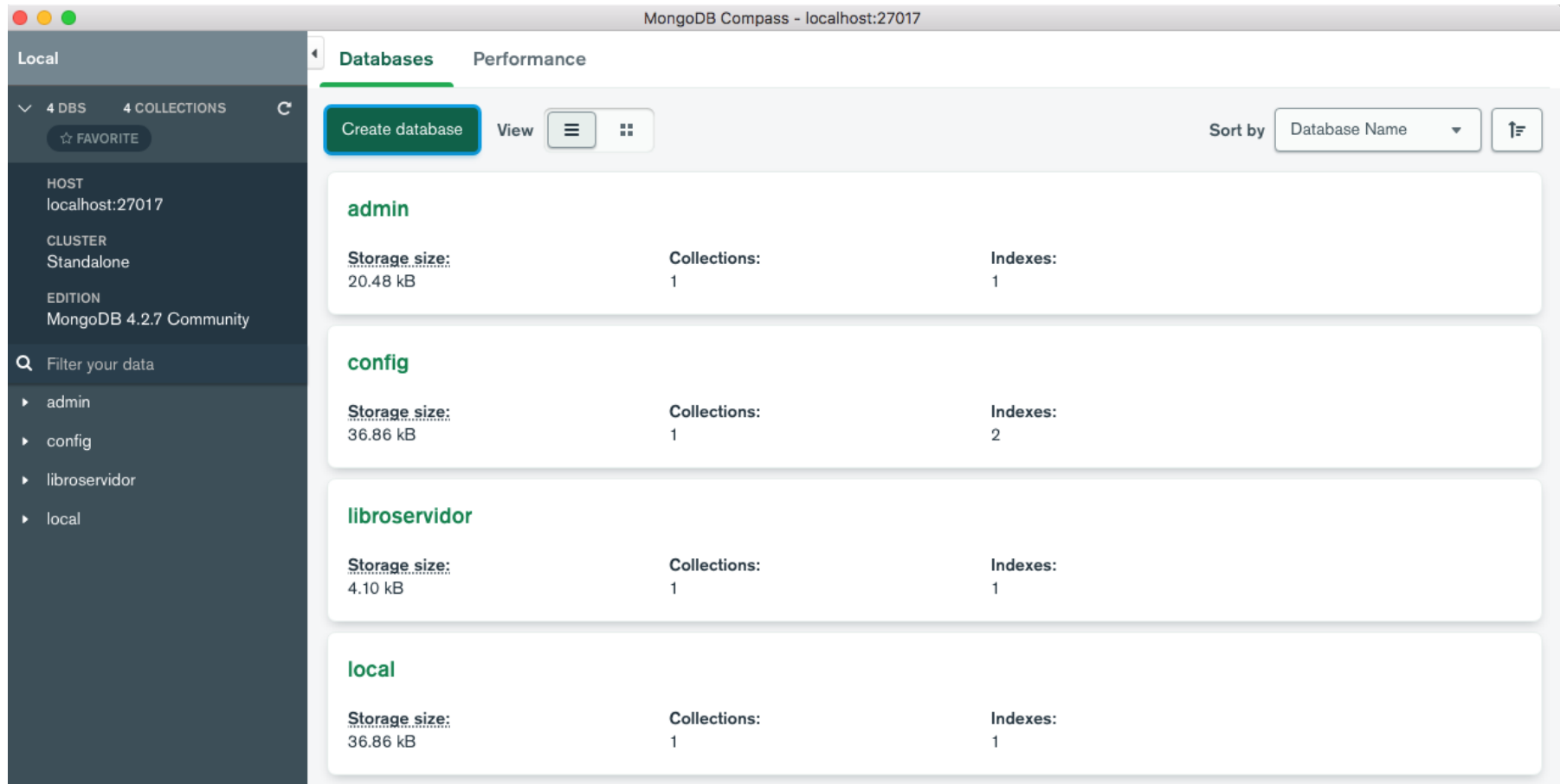
---

Cancel

Create Database



Se ha creado. Vemos:



La seleccionamos:

Finalmente, añadimos el **driver**

Para poder conectarse a un servidor MongoDB desde PHP hay que instalar el driver correspondiente, lo que se hace utilizando **composer**.

Se debe ejecutar el comando:

```
composer require mongodb/mongodb
```

## PRUEBA DE CONEXIÓN DESDE PHP A MONGODB:

### CONEXIÓN

archivo : *conectaMongoDB.php*

```
$Fich_autoloadPhp = 'vendor/autoload.php';

require_once $Fich_autoloadPhp;
try {
    $cadenaConexion = 'mongodb://127.0.0.1:27017';

    $cliente = new MongoDB\Client($cadenaConexion);
    $bd = $cliente->libroservidor;
    echo 'Conectado';
} catch (Exception $e) {
    echo 'error';
    print ($e);
}
```

### Ejecución:

Desde línea de comandos. Estando situados en la carpeta donde se encuentra el archivo fuente

*php conectaMongoDB.php*

## INSERCIÓN

archivo : *insertaMongoDB.php*

```
require 'vendor/autoload.php';

try {
    $cadenaConexion = 'mongodb://127.0.0.1:27017';

    $cliente = new MongoDB\Client($cadenaConexion);
    // Selecciona la base de datos
    $bd = $cliente->libroservidor;
    echo 'Conectado';

    // Selecciona la colección llamada "usuarios" de la BD "libroservidor"
    $usuarios = $bd->usuarios;

    // Inserta un nuevo usuario en la colección
    $usuarios->insertOne(["_id" => "123",
        "usuario" => "antonio",
        "edad" => "27",
        "fechaInscripcion" => "2024-12-20"]);

    // Lee todos los documentos de la colección
    $cursor = $usuarios->find()->toArray();

    // Recorre el array de documentos
    foreach ($cursor as $usuario) {
        var_dump($usuario);
        //print_r($usuario);
    }
} catch (Exception $e) {
    print ($e);
}
```

### Ejecución:

Desde línea de comandos. Estando situados en la carpeta donde se encuentra el archivo fuente

*php insertaMongoDB.php*

## Contenido inicial:

```
_id: "11001113"  
alumno : "Chema Alonso"  
edad : 27  
fechaInscripcion : "2020-05-21T17:18:52.125Z"
```

---

```
_id: "11001013333383"  
alumno : "Will Smith"  
edad : 28  
fechaInscripcion : "2020-05-23T06:22:43.753Z"
```

---

```
_id: "11001999"  
alumno : "Nelson Hernandez"  
fechaInscripcion : "2020-07-23T06:22:43.753Z"
```

## Después:

```
_id: "11001113"  
alumno : "Chema Alonso"  
edad : 27  
fechaInscripcion : "2020-05-21T17:18:52.125Z"
```

---

```
_id: "11001013333383"  
alumno : "Will Smith"  
edad : 28  
fechaInscripcion : "2020-05-23T06:22:43.753Z"
```

---

```
_id: "11001999"  
alumno : "Nelson Hernandez"  
fechaInscripcion : "2020-07-23T06:22:43.753Z"
```

---

```
_id: "123"  
usuario : "antonio"  
edad : "27"  
fechaInscripcion : "2024-12-20"
```

## ACTUALIZACIÓN

archivo : *actualizaMongoDB.php*

```
require 'vendor/autoload.php';

try {
    $cadenaConexion = 'mongodb://127.0.0.1:27017';

    $cliente = new MongoDB\Client($cadenaConexion);
    // Selecciona la BD
    $bd = $cliente->libroservidor;
    echo 'Conectado';

    // Selecciona la colección
    $usuarios = $bd->prueba;

    echo 'Actualiza Usuarios' . "<br>";
    $updateResult = $bd->prueba->updateOne
    (
        ['firstname' => 'Nombre2'],
        ['$set' => ['lastname' => 'Salas']]
    );
} catch (Exception $e) {
    print ($e);
}
```

### Ejecución:

Desde línea de comandos. Estando situados en la carpeta donde se encuentra el archivo fuente

*php actualizaMongoDB.php*

## Contenido inicial:

---

```
_id: ObjectId('66b5fc1b14102e601cc48666')  
firstname : "Nombre1"  
lastname : "Ape1"  
Address : "Dir1"
```

---

```
_id: ObjectId('66b5fc1b14102e601cc48667')  
firstname : "Nombre2"  
lastname : "Ape2"  
Address : "Dir2"
```

## Después:

---

```
_id: ObjectId('66b5fc1b14102e601cc48667')  
firstname : "Nombre2"  
lastname : "Salas"  
Address : "Dir2"
```

---

## CONSULTA

archivo : *selectMongoDB.php*

```
$Fich_autoloadPhp = 'vendor/autoload.php';  
//require_once __DIR__ . $Fich_autoloadPhp;  
require_once $Fich_autoloadPhp;  
  
try {  
    $cadenaConexion = 'mongodb://127.0.0.1:27017';  
  
    $cliente = new MongoDB\Client($cadenaConexion);  
    $bd = $cliente->libroservidor;  
    echo 'Conectado';  
  
    // devuelve todos los usuarios  
    echo 'Todos los usuarios' . "<br>";  
    $usuarios = $bd->usuarios->find();  
  
    foreach ($usuarios as $usuario) {  
        //var_dump($usuario);  
        print_r($usuarios);  
    }  
} catch (Exception $e) {  
    print ($e);  
}
```

Ejecución:

*php selectMongoDB\_ALU.php*

**Salida:**

```
class MongoDB\Model\BSONDocument#13 (1) {  
    private $storage =>  
    array(4) {  
        '_id' =>  
            string(8) "11001113"  
        'alumno' =>  
            string(12) "Chema Alonso"  
        'edad' =>  
            int(27)  
        'fechaInscripcion' =>  
            string(24) "2020-05-21T17:18:52.125Z"  
    }  
}
```

.....



## CONSULTA con WHERE

archivo : *selectWhereMongoDB.php*

```
$Fich_autoloadPhp = 'vendor/autoload.php';  
//require_once __DIR__ . $Fich_autoloadPhp;  
require_once $Fich_autoloadPhp;  
  
try {  
    $cadenaConexion = 'mongodb://127.0.0.1:27017';  
  
    $cliente = new MongoDB\Client($cadenaConexion);  
    $bd = $cliente->libroservidor;  
    echo 'Conectado';  
  
    echo "alumnos 'Will Smith'?.?<br>";  
    $usuarios = $bd->usuarios->find(['alumno' => 'Will Smith']);  
    foreach ($usuarios as $usuario) {  
        var_dump($usuario);  
    }  
  
} catch (Exception $e) {  
    print ($e);  
}
```

Ejecución:

*php selectWhereMongoDB\_ALU.php*

**Salida:**


```
class MongoDB\Model\BSONDocument#13 (1) {  
    private $storage =>  
    array(4) {  
        '_id' =>  
        string(14) "11001013333383"  
        'alumno' =>  
        string(10) "Will Smith"  
        'edad' =>  
        int(28)  
        'fechaInscripcion' =>  
        string(24) "2020-05-23T06:22:43.753Z"  
    }  
}
```

## ERRORES DE CONEXIÓN A MONGO DB CUANDO PENSAMOS QUE ESTÁ INSTALADO

Para verificar que tienes la extensión de MongoDB para PHP instalada en tu sistema, crea un archivo llamado `phpinfo.php` con el siguiente contenido:

```
<?php
phpinfo();
```

Vemos:

PHP Version 8.2.4	
	
System	Darwin MacBook-Air-de-Admin.local 23.2.0 Darwin Kernel Version 23.2.0: Wed Nov 15 21:54:55 PST 2023; root:xnu-10002.61.3~2/RELEASE_ARM64_T8122 x86_64
Build Date	Apr 6 2023 04:09:35
Build System	Darwin xampps-Mac.local 14.5.0 Darwin Kernel Version 14.5.0: Wed Jul 29 02:26:53 PDT 2015; root:xnu-2782.40.9~1/RELEASE_X86_64 x86_64
Configure Command	'./configure' '--prefix=/Applications/XAMPP/xamppfiles' '--with-apxs2=/Applications/XAMPP/xamppfiles/bin/apxs' '--with-config-file-path=/Applications/XAMPP/xamppfiles/etc' '--with-mysql=mysqlnd' '--enable-inline-optimization' '--disable-debug' '--enable-bcmath' '--enable-calendar' '--enable-ctype' '--enable-ftp' '--enable-gd-native-ttf' '--enable-magic-quotes' '--enable-shmop' '--disable-sigchild' '--enable-sysvsem' '--enable-sysvshm' '--enable-wddx' '--with-gdbm=/Applications/XAMPP/xamppfiles' '--with-jpeg-dir=/Applications/XAMPP/xamppfiles' '--with-png-dir=/'

Buscamos la sección *mongodb*. Si existe es que está instalada

Si no está instalada, puedes instalarla usando `pecl`:

```
pecl install mongodb
```

Vemos algo como:

```
Terminal - ...=true&serverSelectionTimeoutMS=2000 × Output
admin@MacBook-Air-de-Admin T3_AplicWebPHP % pecl install mongodb
PHP Warning: Module "mongodb" is already loaded in Unknown on line 0

Warning: Module "mongodb" is already loaded in Unknown on line 0
WARNING: channel "pecl.php.net" has updated its protocols, use "pecl channel-update pecl.php.net" to update
downloading mongodb-1.20.1.tgz ...
Starting to download mongodb-1.20.1.tgz (2,106,641 bytes)
.....
```

Verifica dónde se está cargando la extensión

Puedes verificar desde dónde se está cargando ejecutando este comando en la terminal:

```
php --ini
```

Si encuentras la línea ***extension=mongodb.so*** repetida, coméntala o elimínala:

Guarda los cambios y reinicia el servidor PHP o el servidor web:

```
brew services restart php
```

## SERVICIO DE MONGODB DESDE NETBEANS

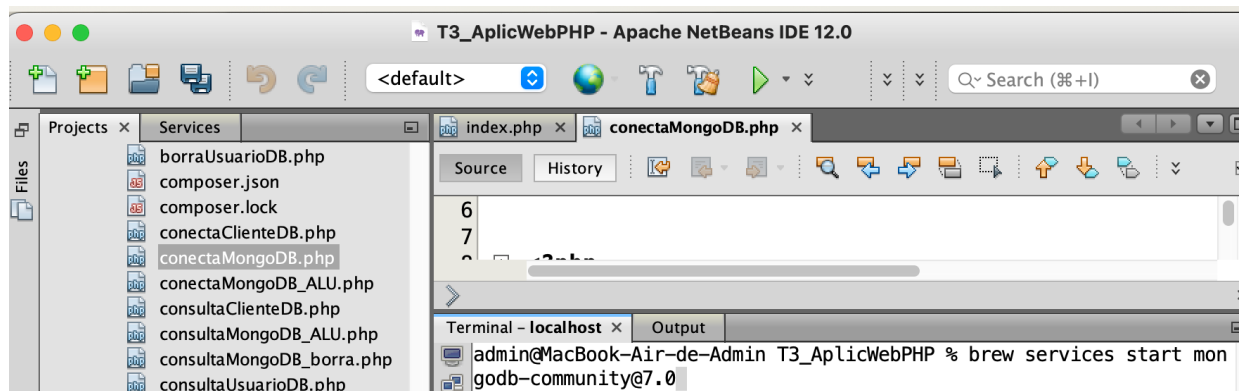
Puedo gestionar servicios *MongoDb* desde el terminal de Netbeans:

Entrar en NETBEANS.

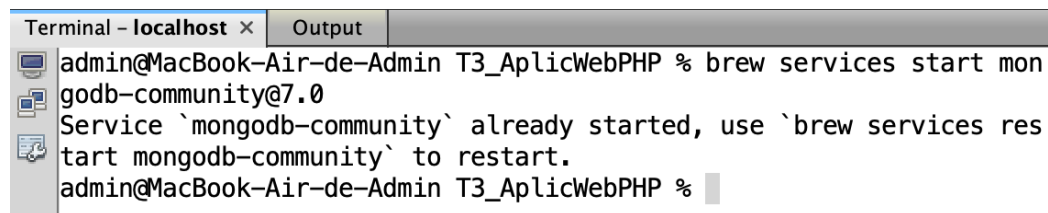
Abrir un terminal con WINDOW-IDE TOOLS-TERMINAL

Situarse en Terminal que acabamos de abrir y escribir

*brew services start mongodb-community*

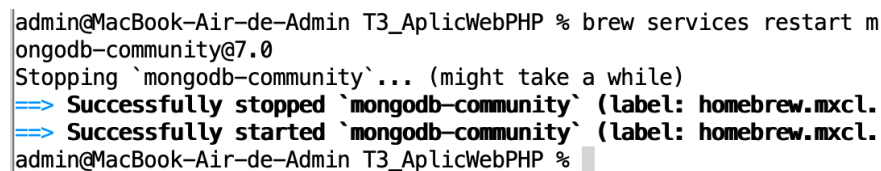


**Arrancar servicio mongodb:**



Para **rearrancar** el servicio mongo DB:

*brew services restart mongodb-community@7.0*



## CONFIGURAR NETBEANS PARA QUE CONECTE CON MONGO

NO FUNCIONA TODAVÍA

NETBEANS-PREFERENCES-PHP

interpreter inicialmente cuando todo funciona tenemos:

/Applications/XAMPP/xamppfiles/etc/php

Pregunto qué php estoy ejecutando:

which php

me dice:

/opt/homebrew/bin/php

## INSTALACIÓN DE MONGODB EN MAC ( Y luego EJECUTAR EN EL NAVEGADOR MONGODB )

Instalar mongodb en un terminal (teniendo conexión a INTERNET)

```
brew services start mongodb-community
```

Comienza a descargar cosas e instalar:

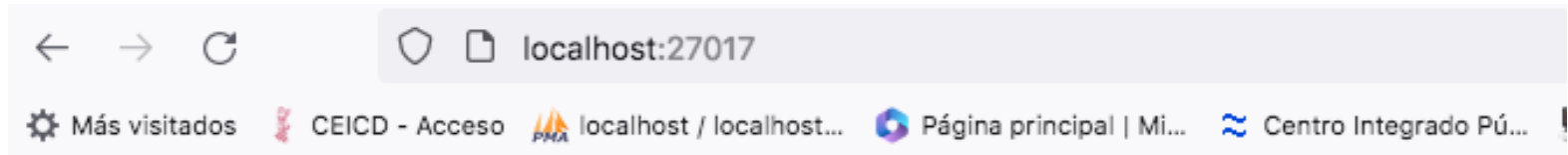
.....

```
==> Successfully stopped `mongodb-community` (label: homebrew.mxcl.mongodb-commu
==> Downloading https://formulae.brew.sh/api/cask_tap_migrations.json
curl: (6) Could not resolve host: formulae.brew.sh
```

Arranco navegador, en la url escribo:

<http://localhost:27017/>

Si veo la siguiente pantalla, es que **ha funcionado**, es que el servidor está esperando peticiones:



Si el servicio *mongodb* **no está arrancado** veré en el navegador algo similar a:



Es decir, esa pantalla sólo se ve si el servicio está arrancado. En caso contrario, al escribir

<http://localhost:27017/>

en el navegador verá la pantalla que se ha mostrado antes