

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

INSTALACIÓN Y PUESTA MARCHA DE MONGODB EN WINDOWS

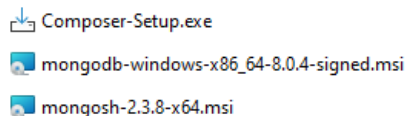
Hay que **instalar**:

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

Nota: Se puede instalar de forma independiente, aunque, en nuestro caso, se instala desde el propio archivo del Community Server

- *Composer*, un Gestor de Dependencias
<https://getcomposer.org/download/>
- El *driver* de PHP para MongoDB. se hace utilizando **composer**. Es el driver que necesitamos para poder conectarse a un servidor MongoDB desde PHP

Archivos necesarios para instalar todo esto:



Orden de instalación:

1. Mongodb Community Server
2. Mongodb Shell
3. Composer
4. Instalar driver mongodb para php usando el gestor de paquetes Composer

INSTALACIÓN DE MONGODB

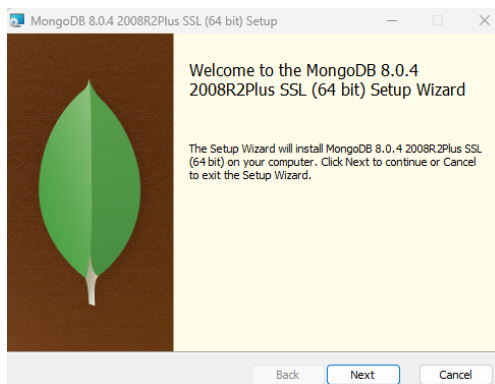
MongoDB Community Server

<https://www.mongodb.com/try/download/community>

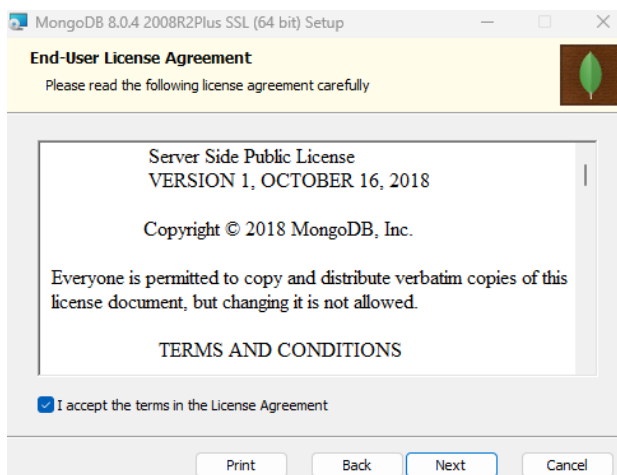
Descargamos el archivo de instalación:

 mongodb-windows-...64-8.0.4-signed.msi

y lo ejecutamos:



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: MongoDB

Account Password:

Service Name: MongoDB

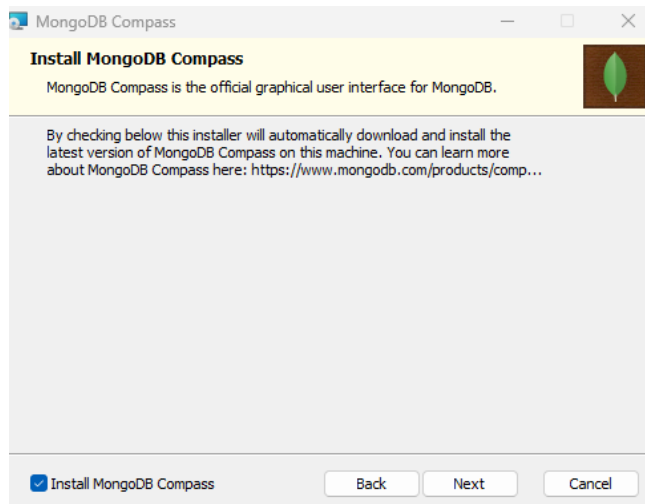
Data Directory: C:\Program Files\MongoDB\Server\8.0\data\

Log Directory: C:\Program Files\MongoDB\Server\8.0\log\

< Back Next > Cancel

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

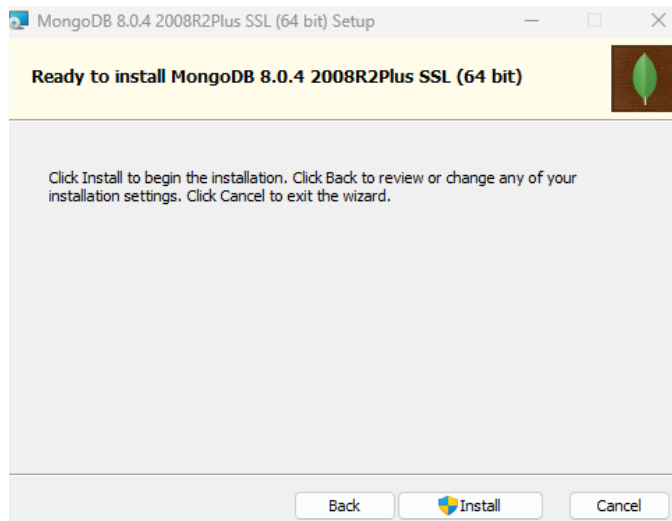
NEXT



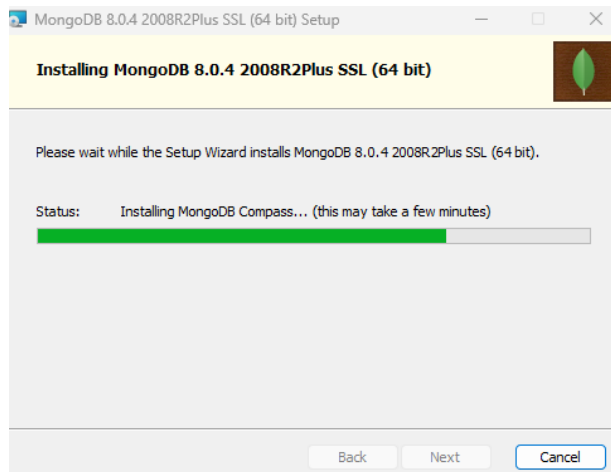
NEXT

Permite que se instale también **MONGODB COMPASS**

Decimos que Sí

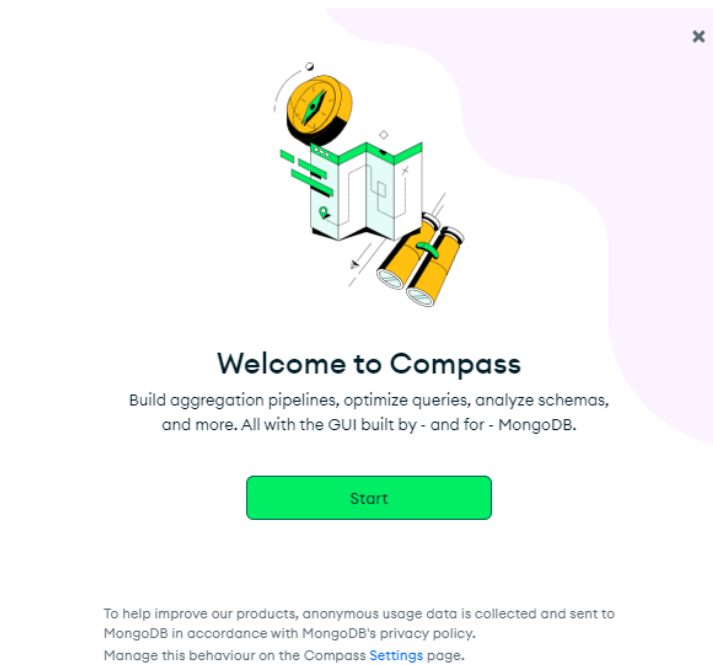


INSTALL



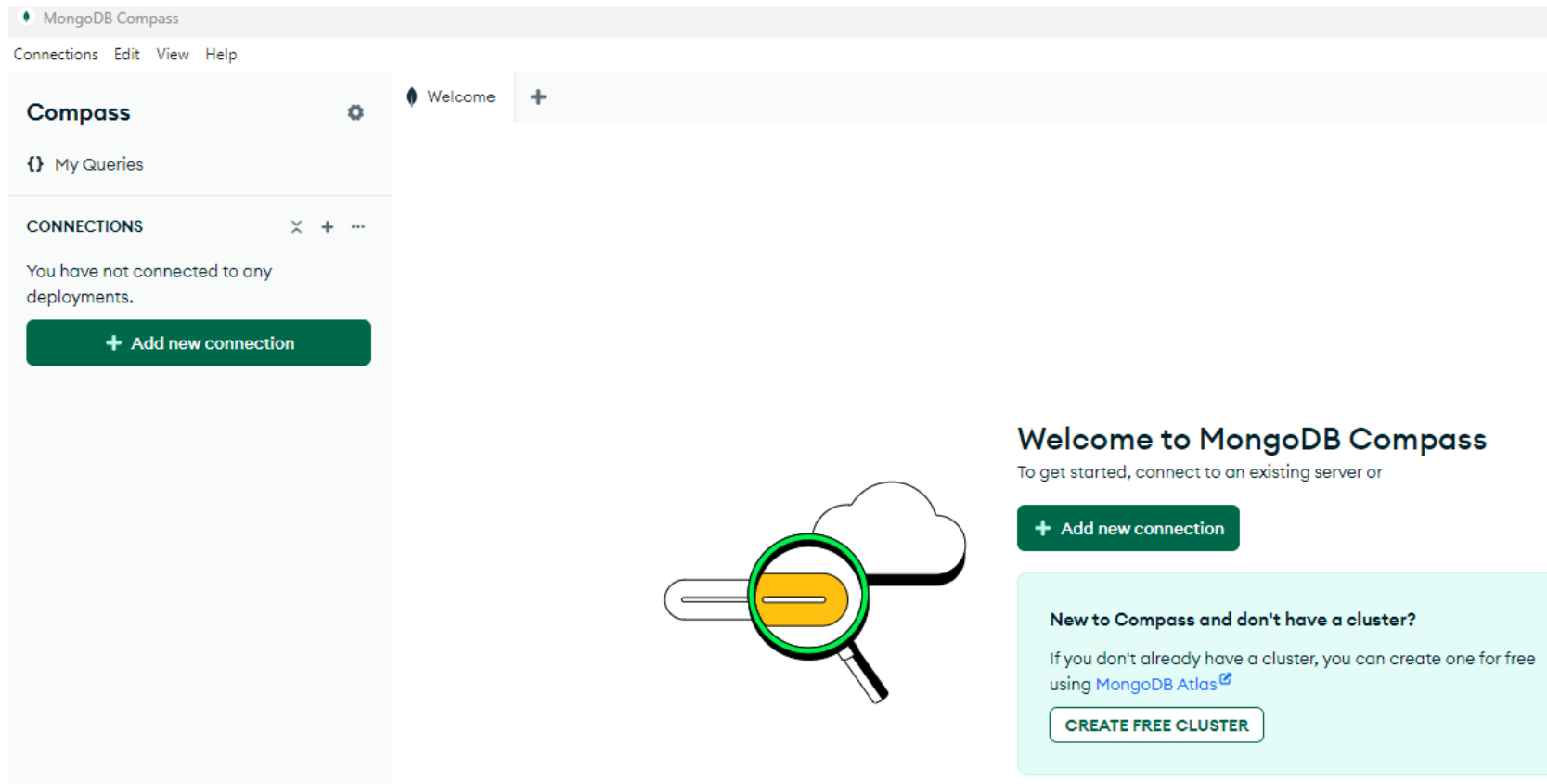
NEXT

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](#) 

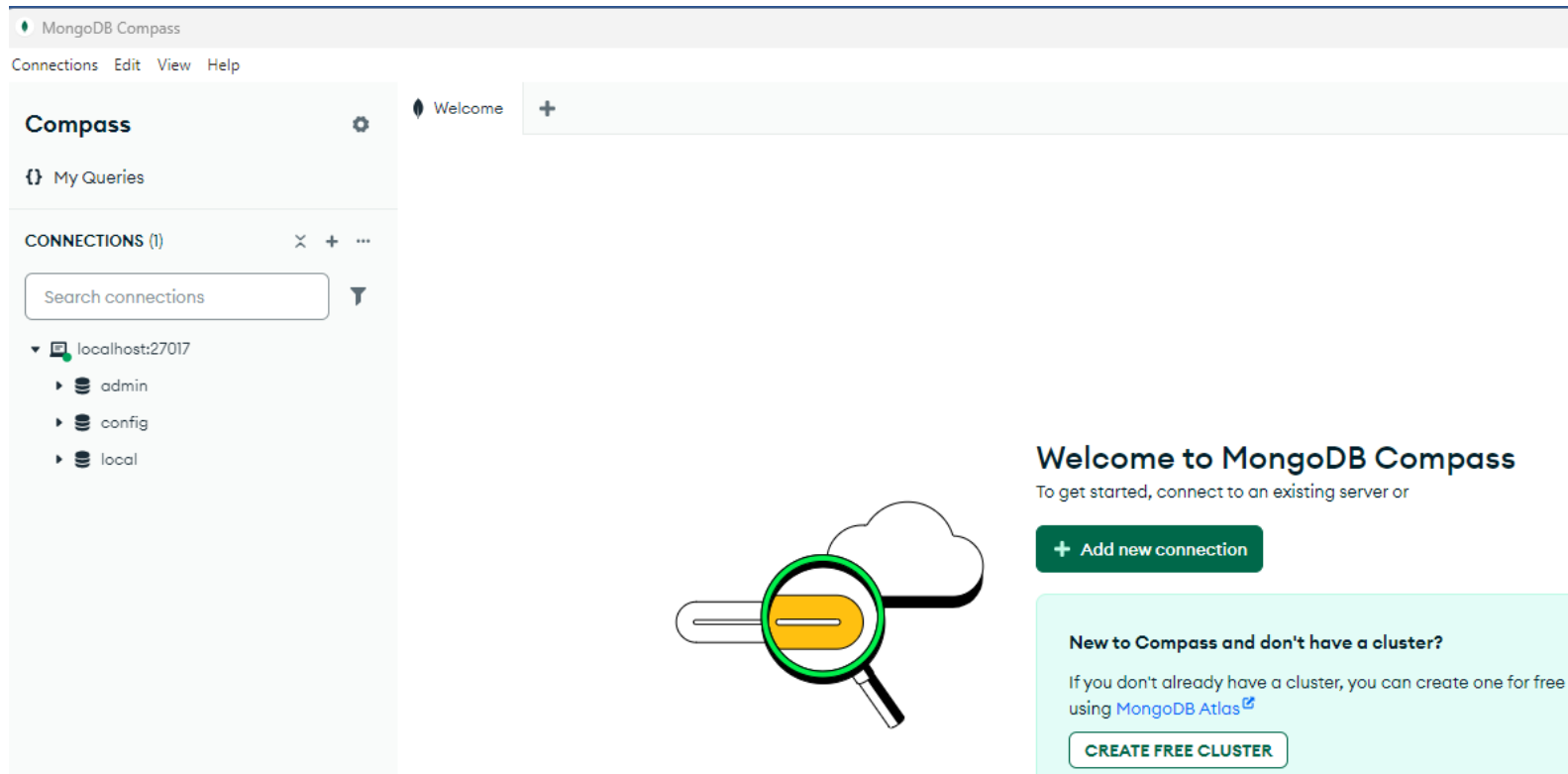
How do I format my connection string?

[See example](#) 

Pulsamos **Save & Connect**

Vemos:

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



Notas de la instalación de MongoDB:

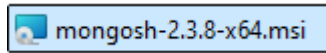
- El servidor escucha en el puerto 27017
- La ruta de instalación en Windows es C:\Program Files\MongoDB
- No es necesario utilizar usuario y contraseña para conectarse

INSTALAR MONGO SHELL (de MONGO DB)

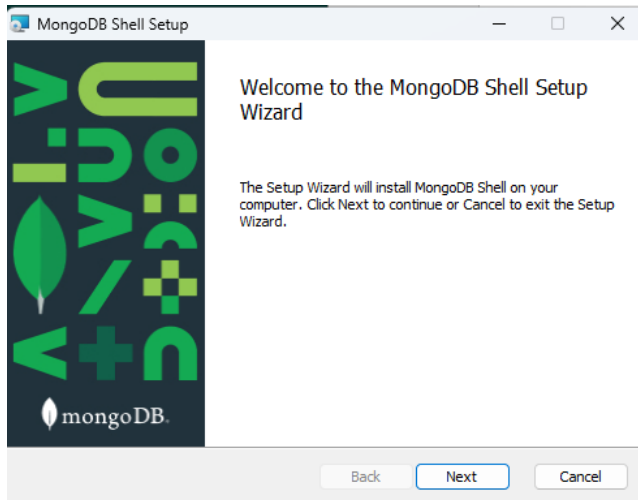
DESCARGAR MongoDB Shell Download

<https://www.mongodb.com/try/download/shell>

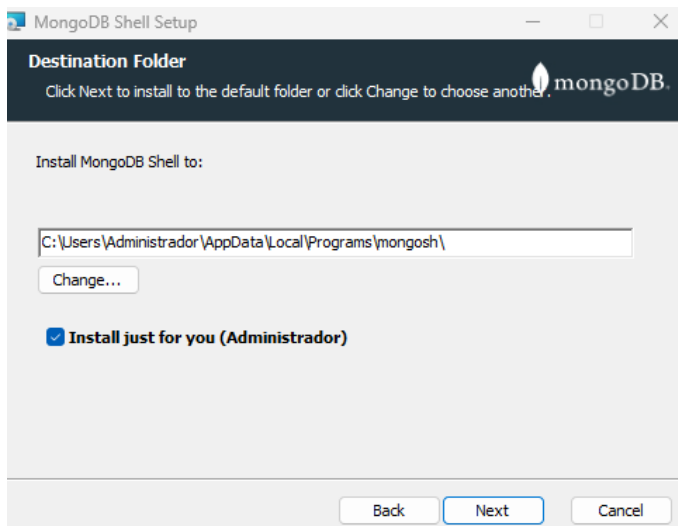
Ejecutar



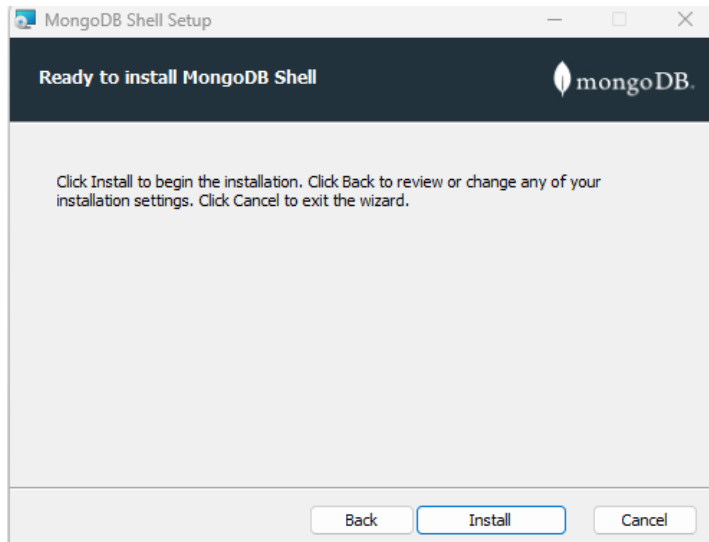
veremos



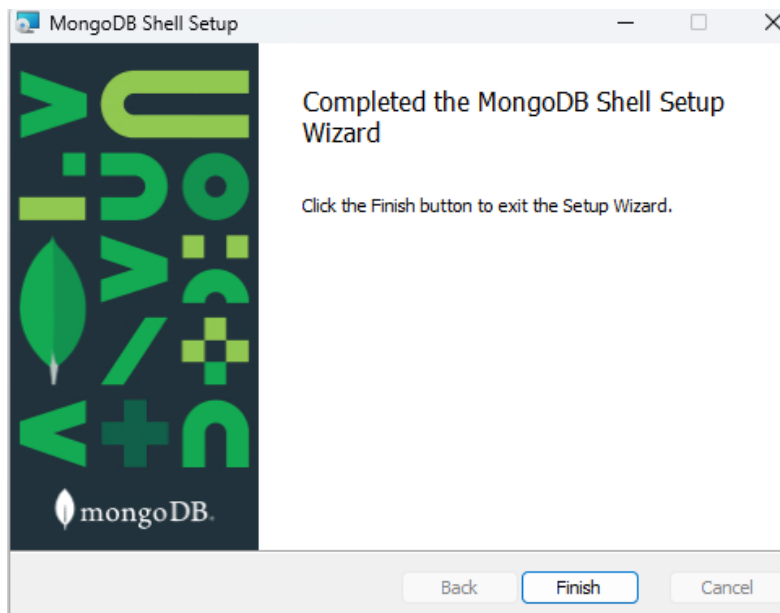
NEXT



NEXT



NEXT



FINISH

Ahora ya está instalado mongo Shell:

Abro ventana de comandos y escribo:

Mongosh

Veré:

```
C:\Users\Administrador>mongosh
Current Mongosh Log ID: 6784ce0dfbfe009ce9cb0ce1
Connecting to:      mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.3.8
Using MongoDB:      8.0.4
Using Mongosh:      2.3.8

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

To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/privacy-policy).
You can opt-out by running the disableTelemetry() command.

-----
The server generated these startup warnings when booting
2025-01-13T08:56:41.304+01:00: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
-----
```

Ya puedo empezar a interactuar con mongo

```
test> show databases;
admin    40.00 KiB
config   72.00 KiB
local    80.00 KiB
test>
```

Nota:

La ruta de instalación del servicio mongodb es
CD C:\Program Files\MongoDB\Server\8.0\bin>

Por si me interesa añadirla al path

CONEXIÓN A MONGODB DESDE CONSOLA. COMANDOS PARA INTERACTUAR CON 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> █
```

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
libroservidor 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
```

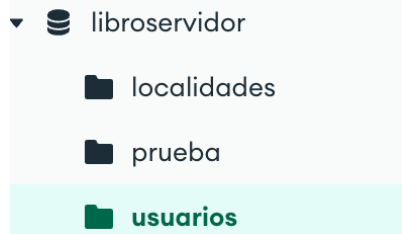
```
libroservidor> show collections
```

```
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 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.

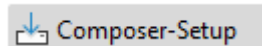
Cuando se desarrolla una aplicación o sitio web en PHP, a menudo se utilizan diferentes librerías o frameworks de terceros para añadir funcionalidades adicionales. Composer permite gestionar estas dependencias de manera sencilla, descargándolas y manteniéndolas actualizadas de manera automática.

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.

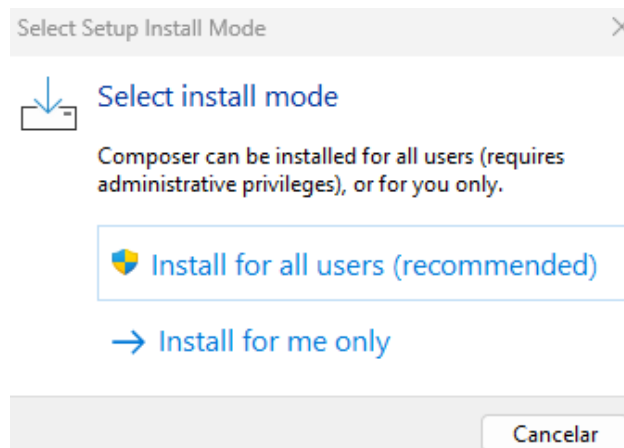
Enlace:

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

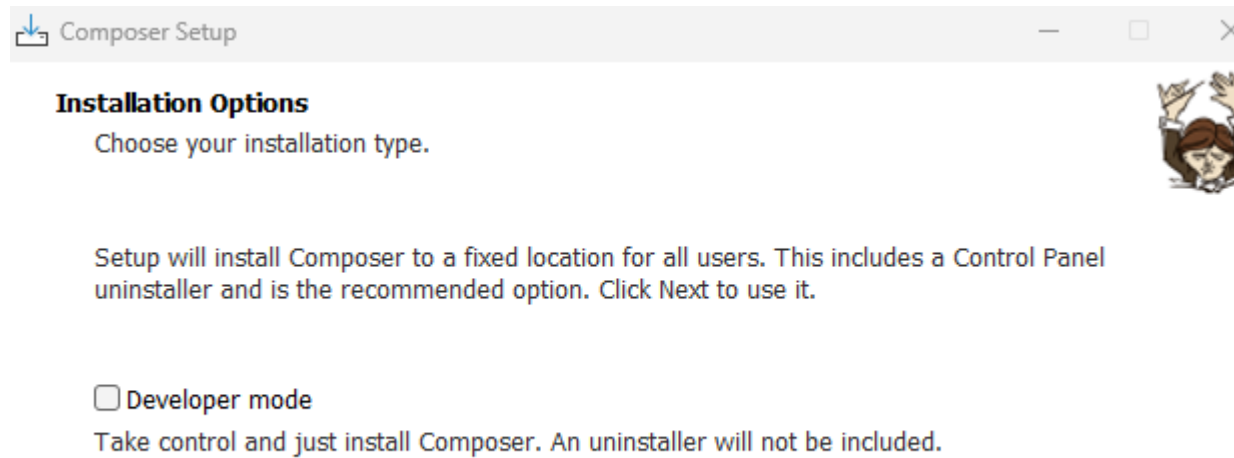
Descargamos *composer.exe* (composer Setup):



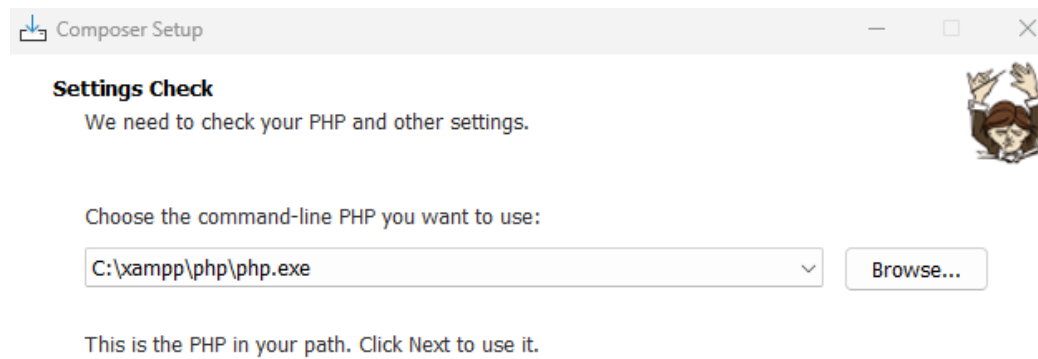
Lo ejecutamos



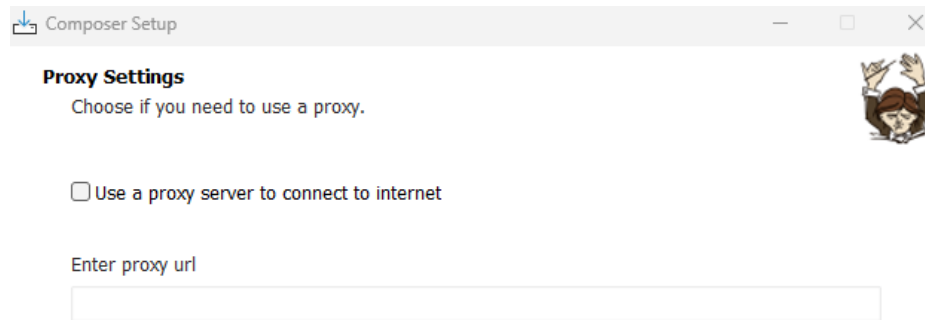
Seleccionamos instalar para todos los usuarios



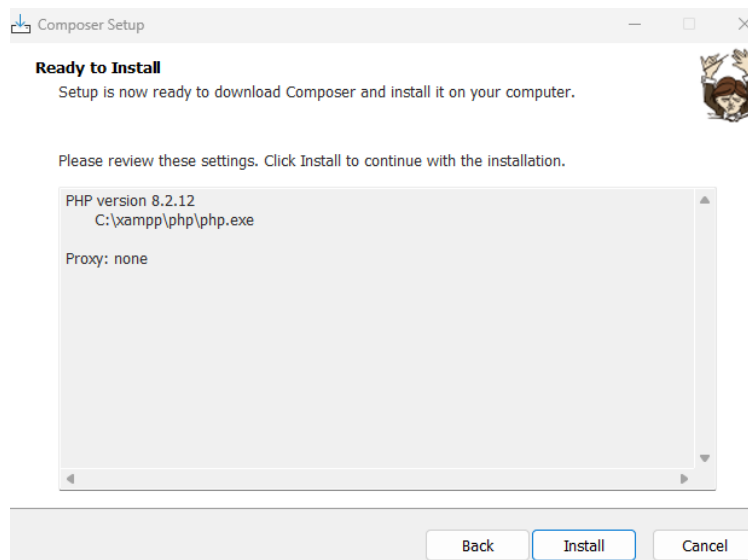
NEXT



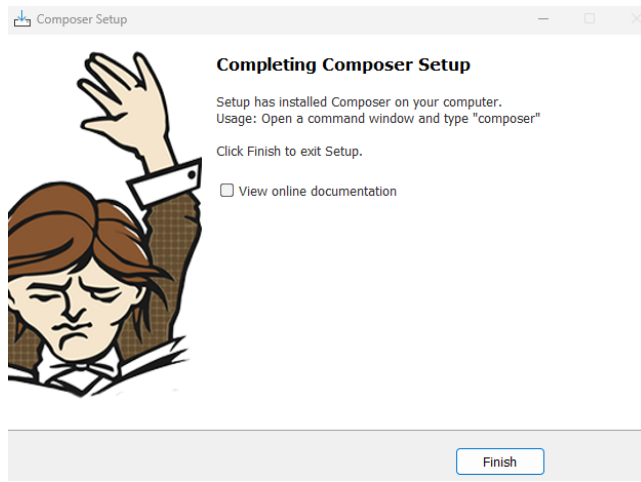
NEXT



NEXT



INSTALL



FINISH

Comandos de Composer

Tabla con algunos de los comandos principales de Composer con su respectiva finalidad:

Comando	Descripción
composer install	Instala las dependencias del proyecto según lo especificado en el archivo composer.json.
composer update	Actualiza las dependencias del proyecto a las últimas versiones compatibles.
composer require	Agrega una nueva dependencia al proyecto y la guarda automáticamente en composer.json.
composer remove	Elimina una dependencia del proyecto y la elimina del archivo composer.json.
composer show	Muestra información sobre las dependencias instaladas en el proyecto.
composer dump-autoload	Regenera el archivo de autocarga autoload.php.
composer validate	Verifica si el archivo composer.json es válido y no contiene errores de sintaxis.
composer self-update	Actualiza Composer a la última versión disponible.
composer require --dev	Agrega una dependencia de desarrollo al proyecto.
composer clear-cache	Limpia la caché de Composer.

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, siguiente:

```
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
```

INSTALACIÓN DE MONGO COMPASS EN WINDOWS (si no lo hemos instalado desde Community Server)

Si en la instalación de **MongoDB Community Server** no hemos instalado Compass, podemos hacerlo con su instalador individual

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

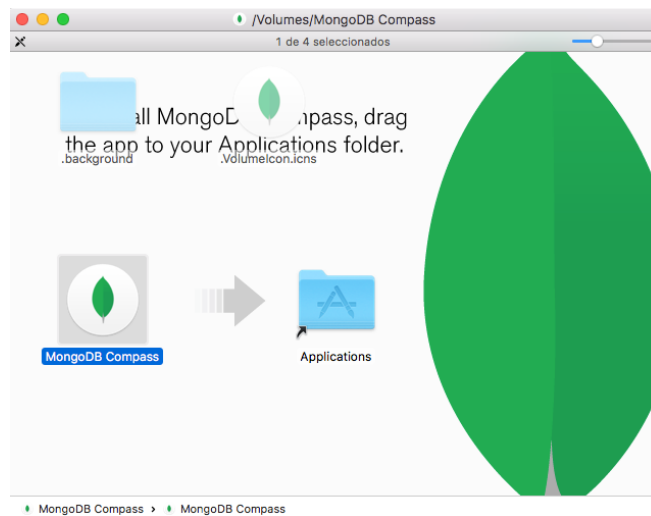
Seguimos el link:

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

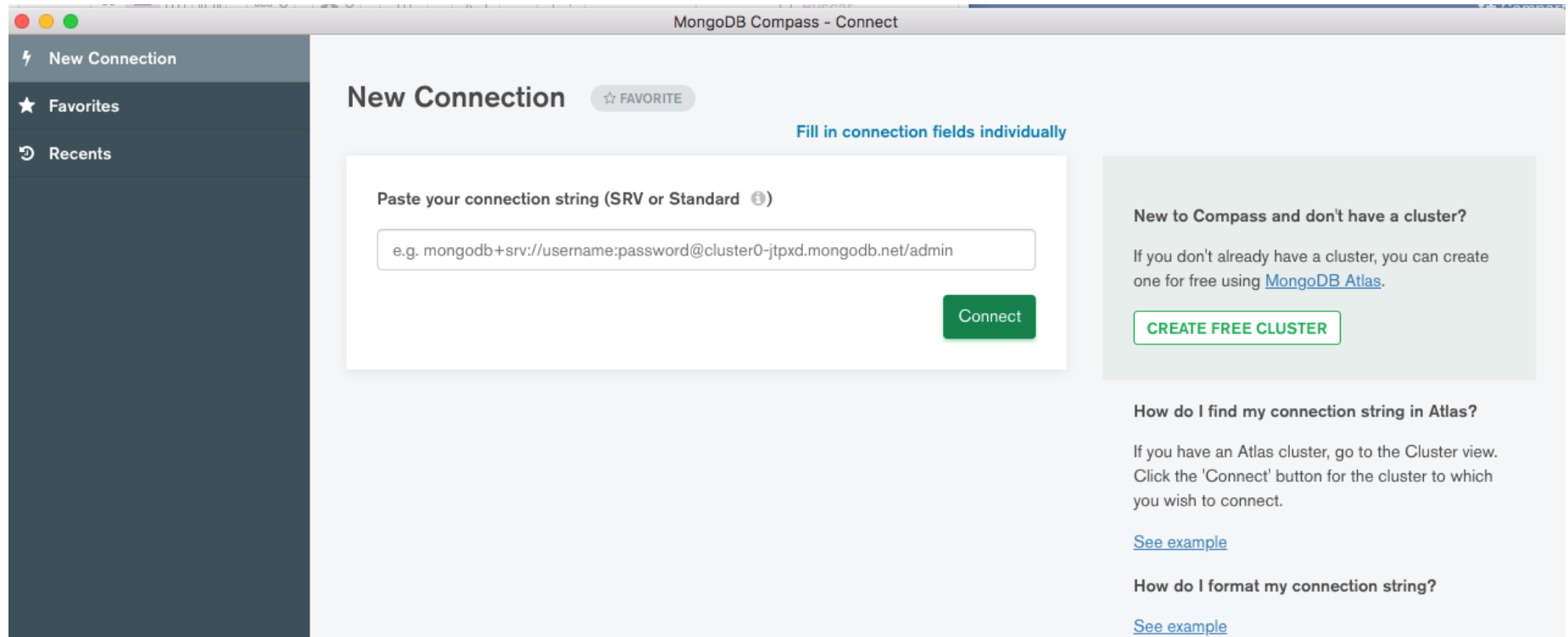
Descargo archivo versión actual. Ejemplo:

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

Lo ejecutamos:

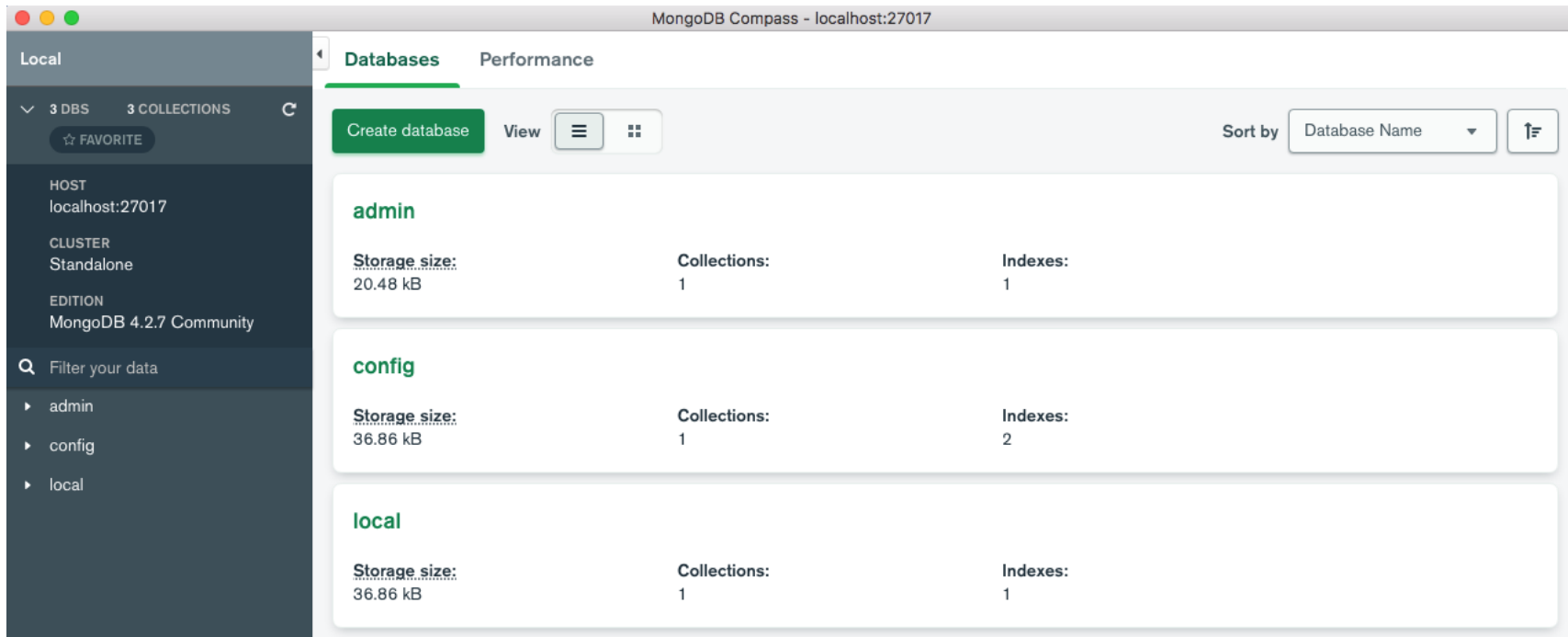


Ejecutamos la Aplicación:



Entramos en New Connection:

Vemos que hay tres bases de datos ya existentes



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.

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



Create Database

Database Name

libroservidor

Collection Name

usuarios

☐ Capped Collection

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

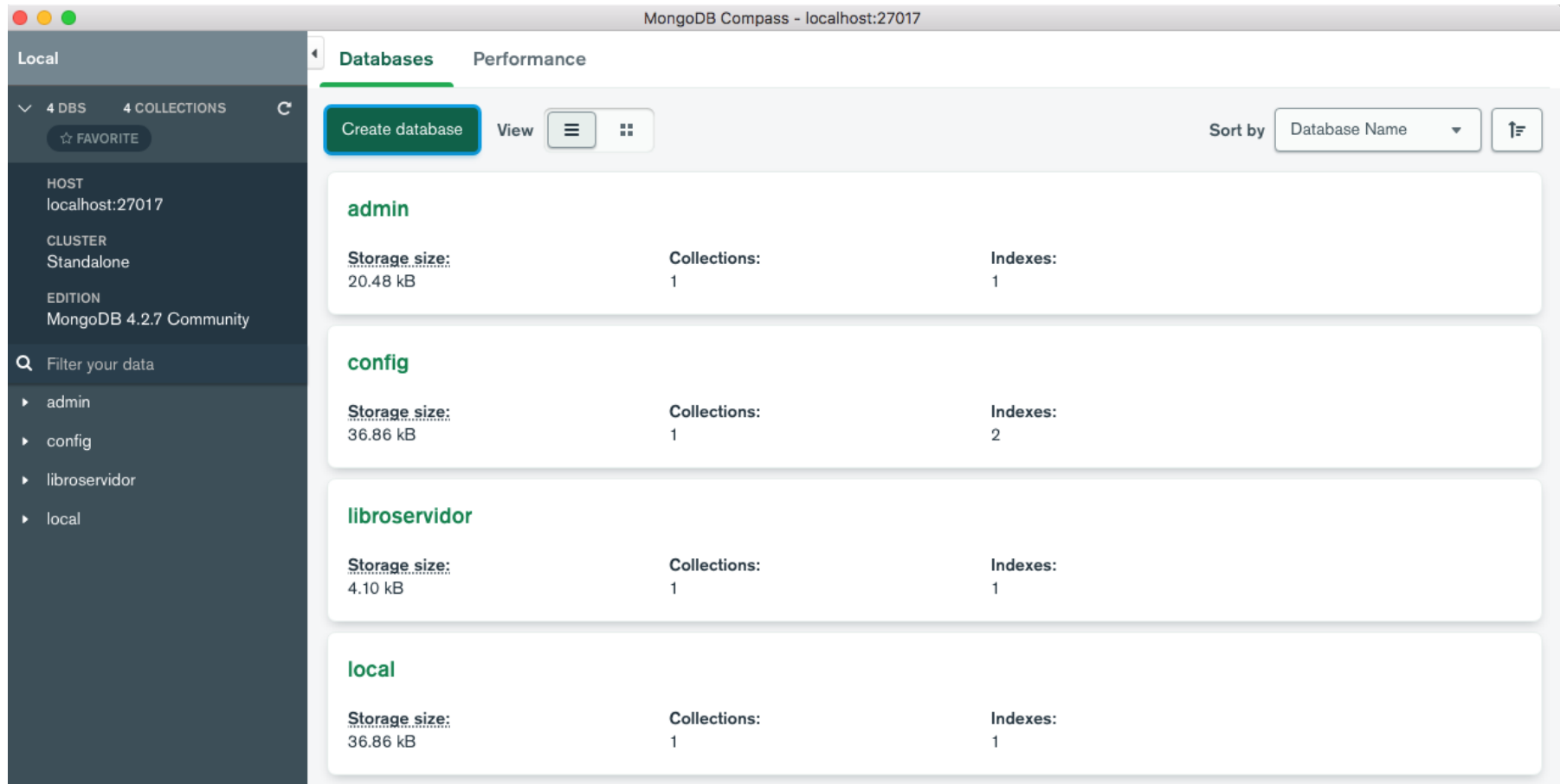
☐ Use Custom Collation

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:

Para poder utilizar los ejemplos posteriores hay que crear una base de 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

libroservidor +

localhost:27017 > libroservidor

Sort by Collection Name

Search connections

CONNECTIONS (10)

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

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

PRUEBA DE CONEXIÓN DESDE PHP A MONGODB:

Recordemos que hemos añadido el **driver de mongo**

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
```

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 {
    // Cadena de conexión a MongoDB
    $cadenaConexion = 'mongodb://127.0.0.1:27017';
    // Crear una instancia del cliente de MongoDB
    $cliente = new MongoDB\Client($cadenaConexion);
    // Selecciona la base de datos
    $bd = $cliente->libroservidor;
    echo 'Conectado';

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

    // Inserta un nuevo usuario en la colección
    $resultado=$usuarios->insertOne([
        "_id" => "1234",
        "usuario" => "antonio",
        "edad" => "27",
        "fechaInscripcion" => "2025-01-20"
    ]);
    // Mensaje de éxito
    echo "Documento insertado con el ID: "
        . $resultado->getInsertedId(). "<br>";

} catch (Exception $e) {
    // captura y muestra error
    echo "Error: " . $e->getMessage();
}
```

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->usuarios;

    // Actualiza un usuario la colección
    echo 'Actualiza Usuarios'. "<br>";
    $updateResult = $usuarios->updateOne
    (
        [
            ['usuario'=>'antonio'],
            ['$set'=>['edad'=>'47',
                    'fechaInscripcion'=>'2025-12-25']]
        ]
    );

    echo "Documentos |modificados: "
        . $updateResult->getModifiedCount();
} 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: "123"  
usuario : "antonio"  
edad : "27"  
fechaInscripcion : "2024-12-25"
```

Después:

```
_id: "123"  
usuario : "antonio"  
edad : "47"  
fechaInscripcion : "2025-12-25"
```

CONSULTA

Sin Formato

archivo : *selectMongoDBprint_r.php* (o *selectMongoDBvar_dump.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 selectMongoDBprint_r.php

Salida:

.....

```
[currentDocument] => MongoDB\Model\BSONDocument Object
(
  [storage:ArrayObject:private] => Array
  (
    [_id] => 11001113
    [alumno] => Chema Alonso
    [edad] => 27
    [fechaInscripcion] => 2020-05-21T17:18:52.125Z
  )
)
```

Con Formato

archivo : *selectMongoDB.php*

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

try {
    // Cadena de conexión a MongoDB
    $cadenaConexion = 'mongodb://127.0.0.1:27017';
    // Crear una instancia del cliente de MongoDB
    $cliente = new MongoDB\Client($cadenaConexion);
    // Selecciona la base de datos
    $bd = $cliente->libroservidor;
    echo 'Conectado';

    // Consulta todos los documentos de la colección
    $usuarios = $bd->usuarios->find();

    echo "Listado de usuarios:<br>";

    // Itera sobre los documentos y muestra los datos
    foreach ($usuarios as $usuario) {
        echo "Nombre del alumno: " . $usuario['usuario'] . "<br>";
        echo "Edad: " . $usuario['edad'] . "<br>";
        echo "Fecha de Inscripción: " . date('Y-m-d',
            strtotime($usuario['fechaInscripcion'])) . "<br>";
        echo "<hr>";
    }
} catch (Exception $e) {
    // Captura y muestra el error
    echo "Error: " . $e->getMessage();
}
```

Ejecución:

php selectMongoDB.php

Salida:

```
ConectadoListado de usuarios:<br>Nombre del alumno: antonio<br>Edad: 27<br>Fecha de Inscripción: 2024-12-25<br><hr>Nombre del alumno: antonio2<br>Edad: 27<br>Fecha de Inscripción: 2024-12-20<br><hr>Nombre del alumno: antonio<br>Edad: 27<br>Fecha de Inscripción: 2025-01-20<br><hr>
```

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 INSTALACIÓN DE MONGODB

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
```

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