

## Práctica Grupal: Instalación de MongoDB

### 1. Instalación de MongoDB en Windows:

1.1 Ingrese a <https://www.mongodb.com/download-center/community>

1.2 Seleccione la pestaña “Windows”, la versión, y haga clic en el botón “Download (msi)”.

Select the server you would like to run:

**MongoDB Community Server**  
FEATURE RICH. DEVELOPER READY.

Version  
4.2.0 (current release) ▼

OS  
Windows x64 x64 ▼

Package  
MSI ▼

**Download**

[https://fastdl.mongodb.org/win32/mongodb-win32-x86\\_64-2012plus-4.2.0-signed.msi](https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2012plus-4.2.0-signed.msi)

1.3 Ejecute el instalador y siga los pasos que le indique el asistente de instalación.

**NO** INSTALARLO COMO SERVICIO:

MongoDB 4.0.3 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:

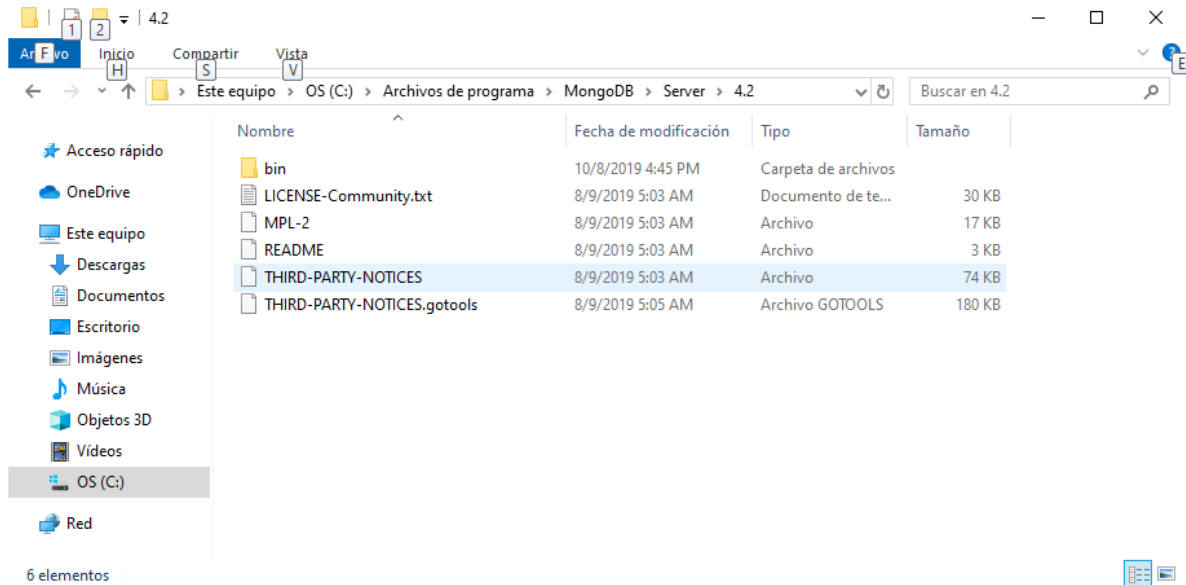
Service Name:

Data Directory:

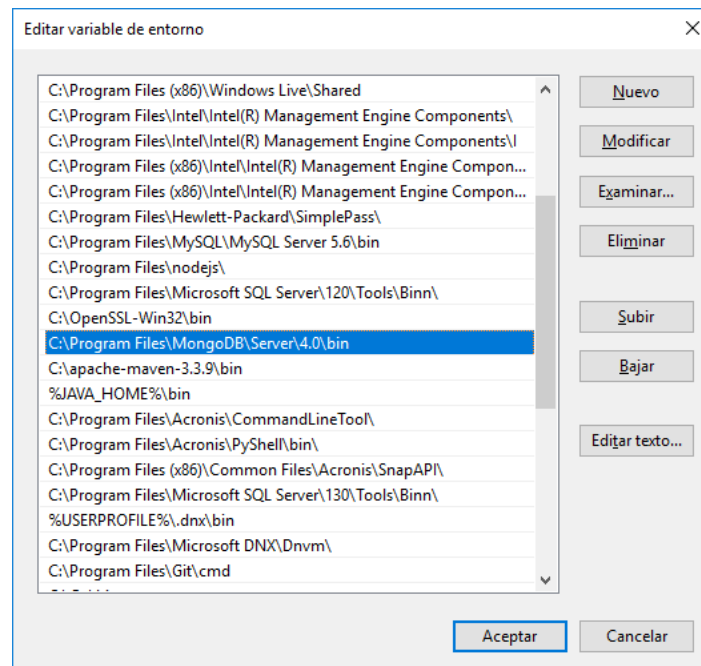
Log Directory:

< Back Next > Cancel

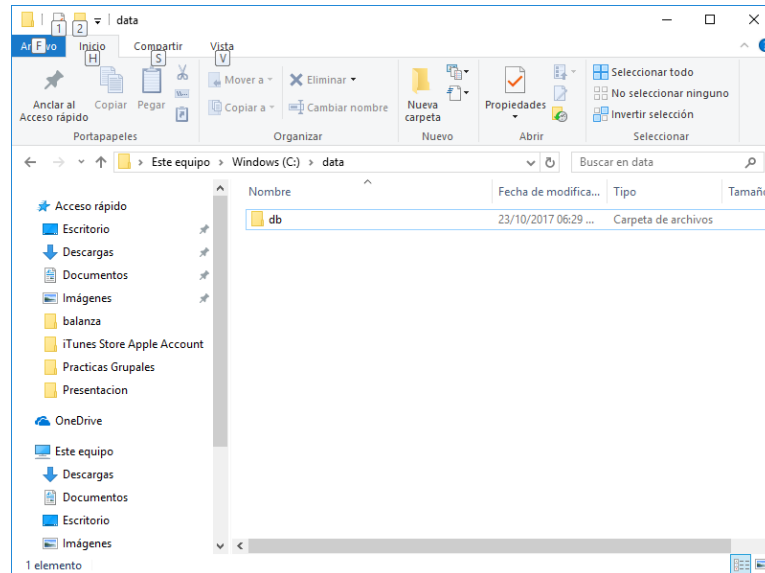
1.4 Asegúrese que la instalación fue correcta ingresando a la carpeta: “C:\Program Files\MongoDB\Server\4.2”, o en donde haya especificado su instalación.



1.5 Configure la ruta de los binarios de MongoDB en la variable PATH del sistema operativo, de tal forma que pueda ejecutar sus comandos desde cualquier ubicación al abrir una terminal:



1.6 Crear, sobre “C:” el directorio “data\db”, que es donde MongoDB generará los archivos relacionados a las bases de datos:



1.7 Dentro del directorio “bin” se encuentra “mongo” y “mongod”. Con “mongod” vamos a arrancar el servidor de base de datos, y con “mongo” vamos a tener acceso al Shell de mongo para ejecutar comandos:

```
Microsoft Windows [Versión 10.0.18362.356]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Jorge>mongod
2019-10-08T16:50:55.944-0500 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --ssl --tls
2019-10-08T16:50:57.168-0500 I CONTROL [initandlisten] MongoDB starting : pid=17112 port=27017 dbpath=C:\data\db\
2019-10-08T16:50:57.168-0500 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2019-10-08T16:50:57.169-0500 I CONTROL [initandlisten] db version v4.2.0
2019-10-08T16:50:57.169-0500 I CONTROL [initandlisten] git version: a4b751dcf51dd249c5865812b390cfd1c0129c30
```

```
C:\Users\Jorge>mongo
MongoDB shell version v4.2.0
connecting to: mongodb://127.0.0.1:27017/
Implicit session: session { "id" : UUID("e...") }
MongoDB server version: 4.0.6
```

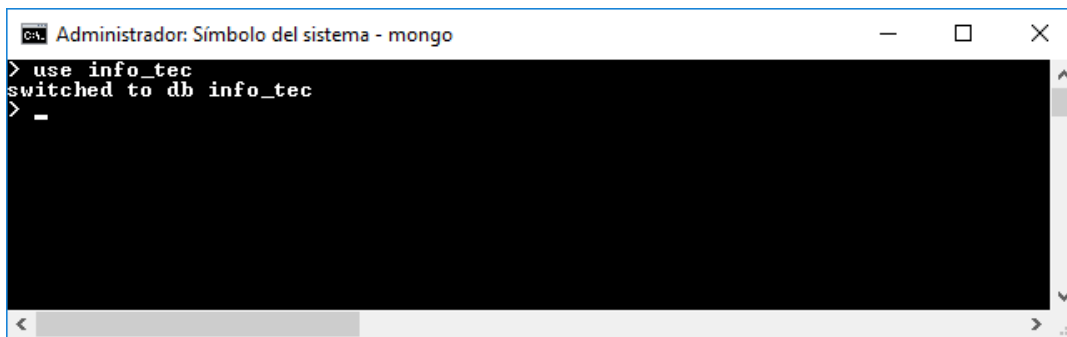
## 2. Instalación de MongoDB en OSX:

2.1 Siga los pasos indicados en la documentación de MongoDB en:  
<https://docs.mongodb.org/manual/tutorial/install-mongodb-on-os-x/>

## 3. Creación y manipulación de una base de datos:

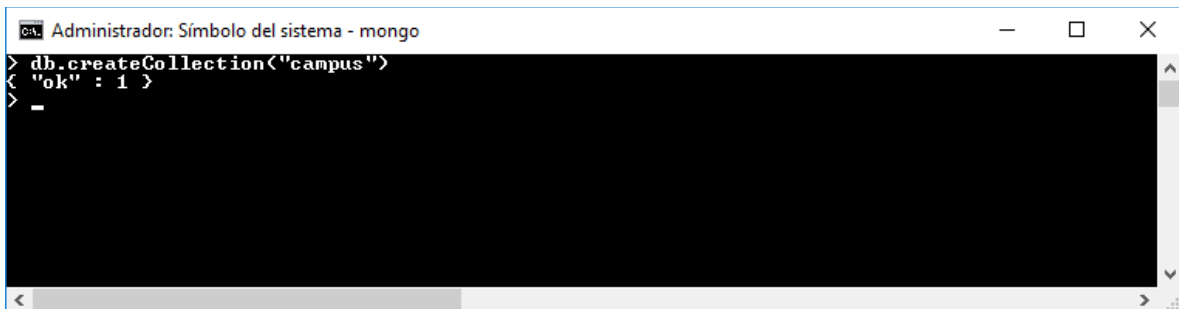
### 3.1 Genere una base de datos llamada *info\_tec*:

a. Ejecute el comando *use info\_tec*



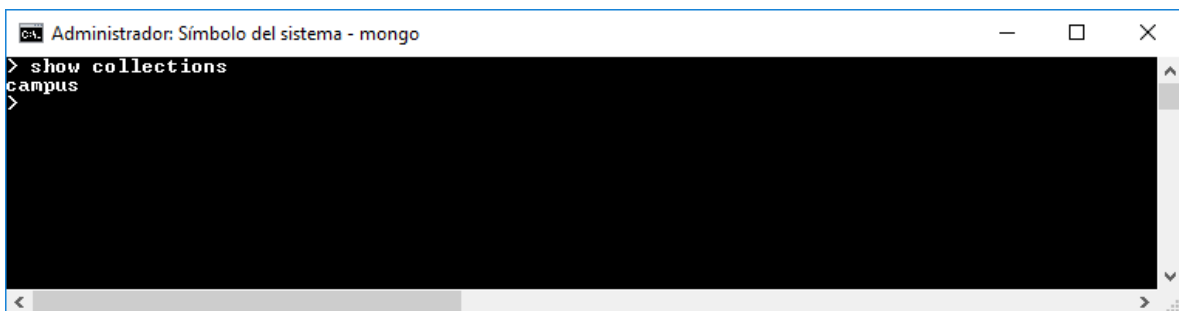
```
Administrador: Símbolo del sistema - mongo
> use info_tec
switched to db info_tec
> -
```

b. Cree una colección llamada *campus* con el comando *db.createCollection("campus")*:



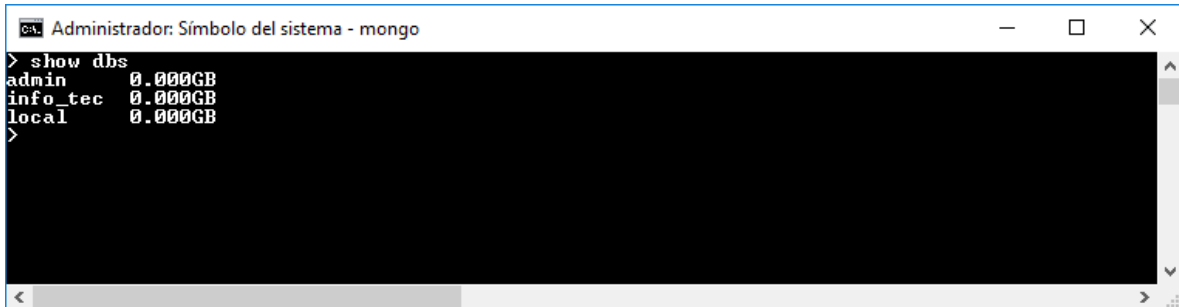
```
Administrador: Símbolo del sistema - mongo
> db.createCollection("campus")
{ "ok" : 1 }
> -
```

c. Valide la creación de la colección con el comando *show collections*:



```
Administrador: Símbolo del sistema - mongo
> show collections
campus
> -
```

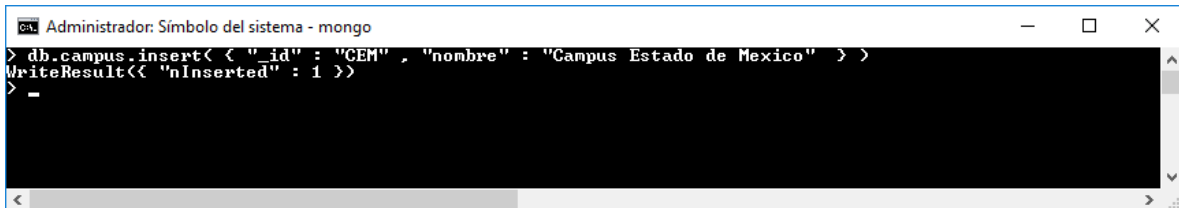
d. Valide la creación de la base de datos con el comando *show dbs*:



```
CA: Administrador: Símbolo del sistema - mongo
> show dbs
admin      0.000GB
info_tec   0.000GB
local      0.000GB
>
```

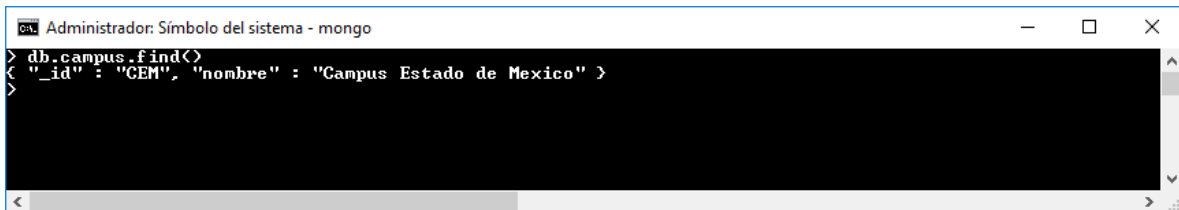
3.3 Genere un nuevo registro en la colección *campus*:

a. Ejecute el comando *db.campus.insert( { "\_id" : "CEM" , "nombre" : "Campus Estado de México" } )*:



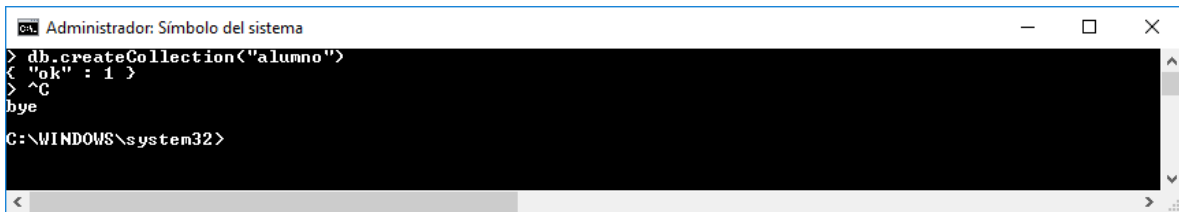
```
CA: Administrador: Símbolo del sistema - mongo
> db.campus.insert( { "_id" : "CEM" , "nombre" : "Campus Estado de Mexico" } )
WriteResult<{ "nInserted" : 1 }>
> -
```

b. Valide la inserción del registro con el comando *db.campus.find()*:



```
CA: Administrador: Símbolo del sistema - mongo
> db.campus.find(<>)
{ "_id" : "CEM", "nombre" : "Campus Estado de Mexico" }
>
```

3.4 Cree una colección llamada *alumno* con el comando `db.createCollection("alumno")`:



```
Administrador: Símbolo del sistema
> db.createCollection("alumno")
< { "ok" : 1 }
> ^C
bye
C:\WINDOWS\system32>
```

3.5 Genere nuevos registros dentro de la colección *alumno* con los siguientes comandos:

- Genere un archivo de texto con el siguiente contenido y guárdelo como “alumnos.js”

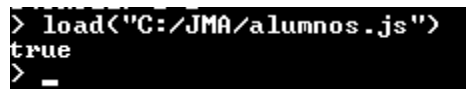
```
db.alumno.insert( { "campus" : "CEM" , "nombre" : "Juan Perez" , "matricula" :  
"00954558" , "materias" : [ { "_id" : "TC3052" , "nombre" : "Laboratorio de Desarrollo  
de Aplicaciones Web" } , { "_id" : "TC2026" , "nombre" : "Desarrollo de Aplicaciones  
Web" } , { "_id" : "TC2025" , "nombre" : "Programacion Avanzada" } ] } )
```

```
db.alumno.insert( { "campus" : "CEM" , "nombre" : "Pedro Gonzalez" , "matricula" :  
"00954548" , "materias" : [ { "_id" : "TC3052" , "nombre" : "Laboratorio de Desarrollo  
de Aplicaciones Web" } , { "_id" : "TC2025" , "nombre" : "Programacion Avanzada" } ]  
} )
```

```
db.alumno.insert( { "campus" : "CEM" , "nombre" : "Hugo Ramirez" , "matricula" :  
"00954538" , "materias" : [ { "_id" : "TC3052" , "nombre" : "Laboratorio de Desarrollo  
de Aplicaciones Web" } , { "_id" : "TC2026" , "nombre" : "Desarrollo de Aplicaciones  
Web" } ] } )
```

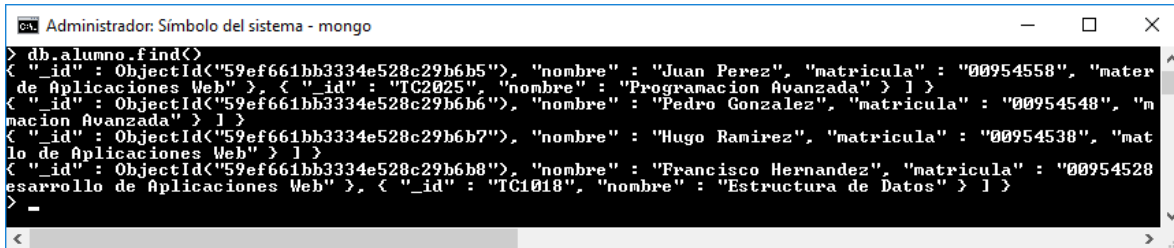
```
db.alumno.insert( { "campus" : "CEM" , "nombre" : "Francisco Hernandez" ,  
"matricula" : "00954528" , "materias" : [ { "_id" : "TC3052" , "nombre" : "Laboratorio  
de Desarrollo de Aplicaciones Web" } , { "_id" : "TC2026" , "nombre" : "Desarrollo de  
Aplicaciones Web" } , { "_id" : "TC1018" , "nombre" : "Estructura de Datos" } ] } )
```

- Ejecute el comando `load("DIRECTORIO/alumnos.js")`:



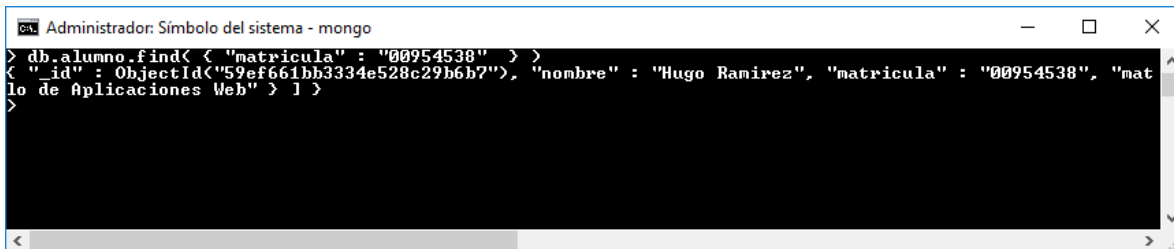
```
> load("C:/JMA/alumnos.js")
true
> -
```

3.6 Valide la creación de los registros con el comando `db.alumno.find()`:



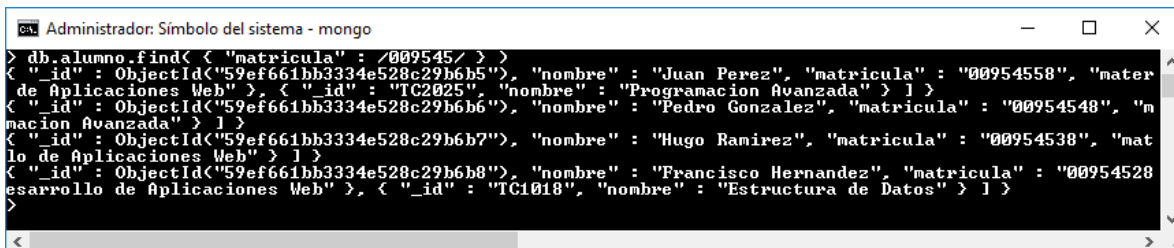
```
Administrador: Símbolo del sistema - mongo
> db.alumno.find()
{ "_id" : ObjectId("59ef661bb3334e528c29b6b5"), "nombre" : "Juan Perez", "matricula" : "00954558", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b6"), "nombre" : "Pedro Gonzalez", "matricula" : "00954548", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b7"), "nombre" : "Hugo Ramirez", "matricula" : "00954538", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b8"), "nombre" : "Francisco Hernandez", "matricula" : "00954528", "materias" : { "_id" : "TC1018", "nombre" : "Estructura de Datos" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b9"), "nombre" : "Estructura de Datos", "matricula" : "00954528", "materias" : { "_id" : "TC1018", "nombre" : "Estructura de Datos" } }
```

3.7 Consulte el alumno cuya matrícula es igual a “00954538” con el comando `db.alumno.find( { “matricula” : “00954538” } )`:



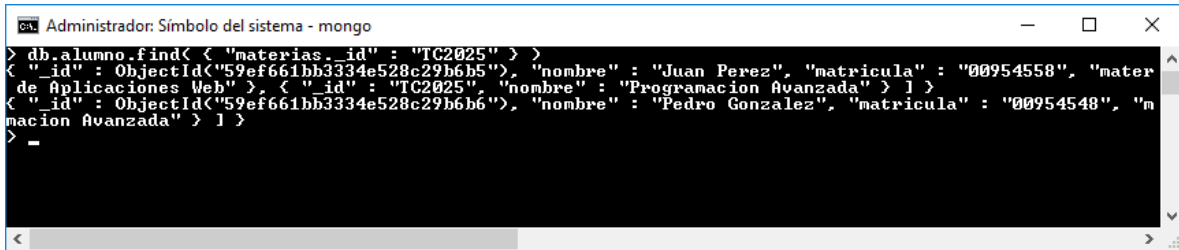
```
Administrador: Símbolo del sistema - mongo
> db.alumno.find( { "matricula" : "00954538" } )
{ "_id" : ObjectId("59ef661bb3334e528c29b6b7"), "nombre" : "Hugo Ramirez", "matricula" : "00954538", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
```

3.8 Consulte el alumno cuya matrícula contiene la cadena “009545” con el comando `db.alumno.find( { “matricula” : /009545/ } )`:



```
Administrador: Símbolo del sistema - mongo
> db.alumno.find( { "matricula" : /009545/ } )
{ "_id" : ObjectId("59ef661bb3334e528c29b6b5"), "nombre" : "Juan Perez", "matricula" : "00954558", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b6"), "nombre" : "Pedro Gonzalez", "matricula" : "00954548", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b7"), "nombre" : "Hugo Ramirez", "matricula" : "00954538", "materias" : { "_id" : "TC2025", "nombre" : "Programacion Avanzada" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b8"), "nombre" : "Francisco Hernandez", "matricula" : "00954528", "materias" : { "_id" : "TC1018", "nombre" : "Estructura de Datos" } }
{ "_id" : ObjectId("59ef661bb3334e528c29b6b9"), "nombre" : "Estructura de Datos", "matricula" : "00954528", "materias" : { "_id" : "TC1018", "nombre" : "Estructura de Datos" } }
```

3.9 Consulte los alumnos que pertenecen a la materia “TC2025” (Programación Avanzada) con el comando `db.alumno.find( { “materias._id” : “TC2025” } )`:



```
Administrador: Símbolo del sistema - mongo
> db.alumno.find(< < "materias._id" : "TC2025" > >
< "_id" : ObjectId("59ef661bb3334e528c29b6b5"), "nombre" : "Juan Perez", "matricula" : "00954558", "mater
de Aplicaciones Web" >, < "_id" : "TC2025", "nombre" : "Programacion Avanzada" > 1 >
< "_id" : ObjectId("59ef661bb3334e528c29b6b6"), "nombre" : "Pedro Gonzalez", "matricula" : "00954548", "m
macion Avanzada" > 1 >
> _
```

3.10 Un ejemplo para el escenario en el que tengan un arreglo de IDs en una colección y se quiera consultar la información de los documentos de la colección correspondiente a esos IDs:

```
//Creamos la base de datos
use videojuegosdb

//Creamos la coleccion de consolas
db.createCollection("consolas")

//Insertamos una consola
db.consolas.insert({"_id":"ps4", "nombre":"Playstation 4", "juegos":[0,1,2]})

//Creamos la coleccion de videojuegos
db.createCollection("juegos")

//Insertamos 4 juegos
db.juegos.insert({"_id":0, "nombre":"Juego PS4 1", "developer":"Developer 1"})
db.juegos.insert({"_id":1, "nombre":"Juego PS4 2", "developer":"Developer 2"})
db.juegos.insert({"_id":2, "nombre":"Juego PS4 3", "developer":"Developer 3"})
db.juegos.insert({"_id":3, "nombre":"Juego PS4 4", "developer":"Developer 4"})

//https://docs.mongodb.com/manual/reference/method/cursor.forEach/
//La funcion find() regresa un cursor, el cual tiene un metodo forEach que nos
// ayuda a iterar el cursor. Aqui, lo que hacemos es obtener el arreglo de
// IDs de juegos de una consola en especifico, y por cada ID realizamos
// la busqueda sobre la coleccion de juegos.
db.consolas.find({"_id":"ps4"},{"_id":0, "juegos":1}).forEach(function(consola){
    print(tojson(consola));
    consola.juegos.forEach(function(idJuego){
        print(idJuego);
        db.juegos.find({"_id":idJuego}).forEach(function(documento){
            print(tojson(documento));
        });
    });
});
```



```
    });  
  }  
}
```

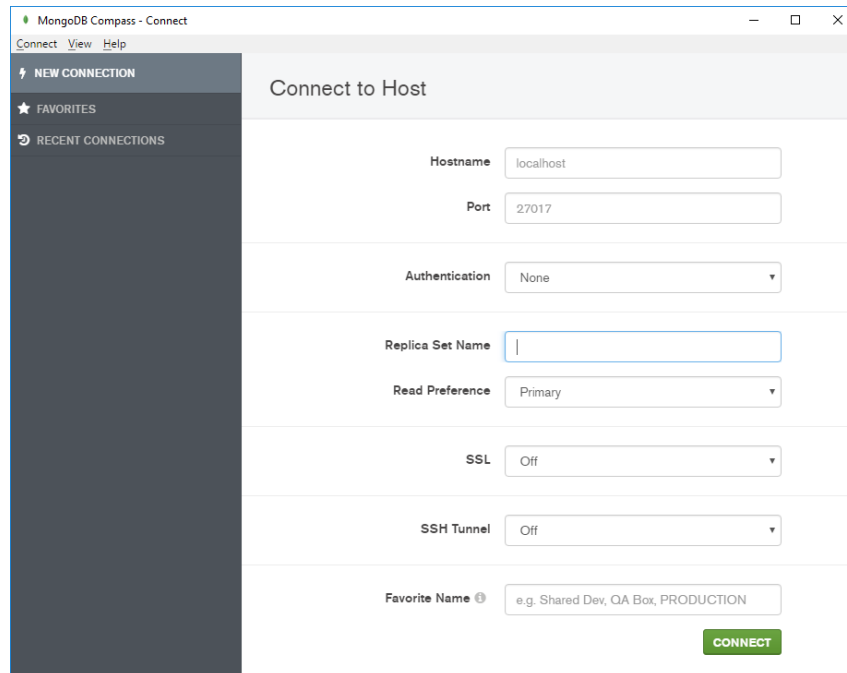
3.11 Un ejemplo para el escenario en el que tengan un arreglo de documentos Json como valor de una propiedad de la colección y se quiera consultar la información de esos documentos:

```
//Creamos la coleccion de consolas 2  
db.createCollection("consolas2")  
  
//Insertamos una consola  
db.consolas2.insert({"_id":"ps4", "nombre":"Playstation 4", "juegos":[{"_id":0,  
"nombre":"Juego PS4 1", "developer":"Developer 1"}, {"_id":1, "nombre":"Juego PS4 2",  
"developer":"Developer 2"}, {"_id":2, "nombre":"Juego PS4 3", "developer":"Developer  
3"}, {"_id":3, "nombre":"Juego PS4 4", "developer":"Developer 4"}]})  
  
db.consolas2.aggregate([  
  { $unwind: '$juegos'},  
  { $match: {'juegos.nombre': /1/i }},  
  { $project: { _id: 0, nombre: '$juegos.nombre', developer: '$juegos.developer' } }  
]);
```

#### 4. Compass: Cliente gráfico para MongoDB

##### 4.1 Abrir Compass, que se instaló junto con MongoDB

##### 4.2 Conectarse a mongo con las opciones por default:

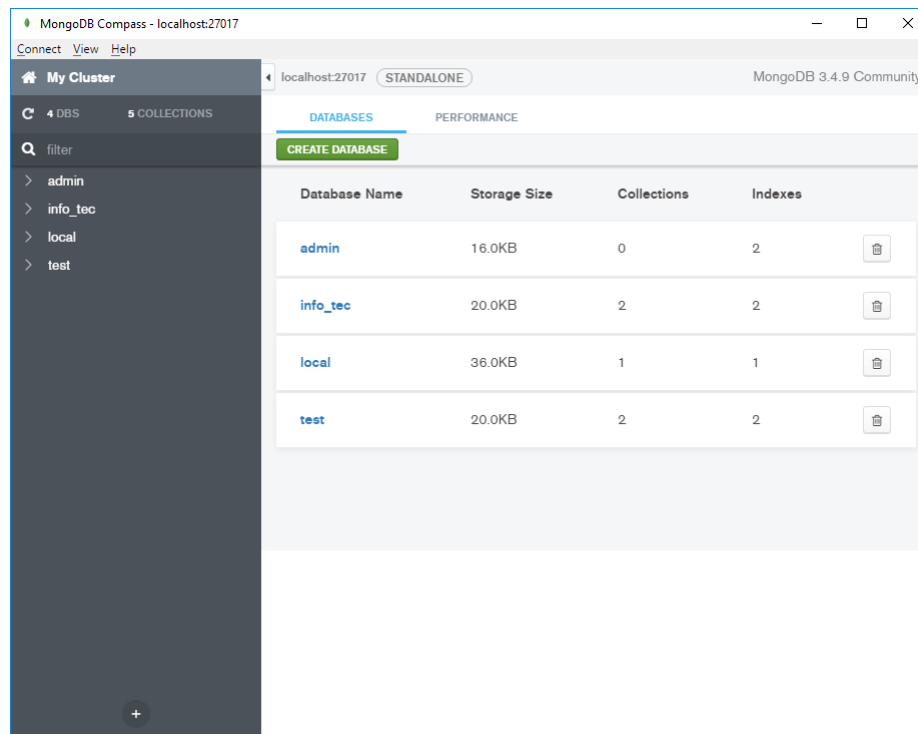


The screenshot shows the 'MongoDB Compass - Connect' window. On the left is a sidebar with 'NEW CONNECTION', 'FAVORITES', and 'RECENT CONNECTIONS'. The main area is titled 'Connect to Host' and contains the following fields:

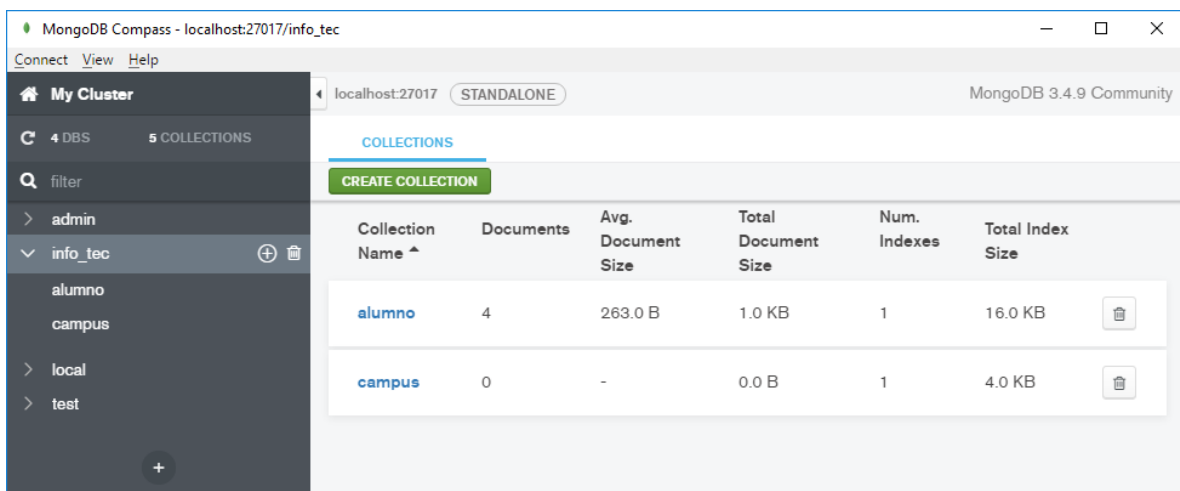
- Hostname: localhost
- Port: 27017
- Authentication: None (dropdown)
- Replica Set Name: (empty text box)
- Read Preference: Primary (dropdown)
- SSL: Off (dropdown)
- SSH Tunnel: Off (dropdown)
- Favorite Name: e.g. Shared Dev, QA Box, PRODUCTION (text box)

A green 'CONNECT' button is located at the bottom right of the form.

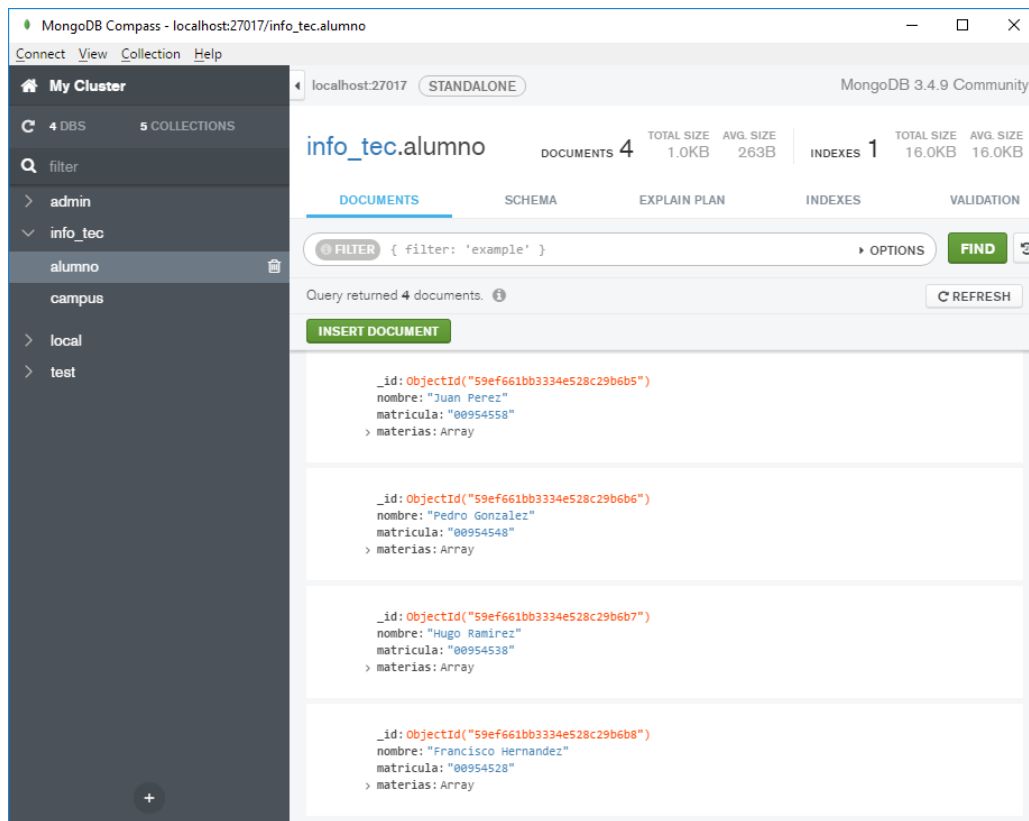
#### 4.3 Nos va mostrar las bases de datos creadas:



#### 4.4 Elijan la base *info\_tec* para visualizar las colecciones:



4.5 Elijan la colección *alumno* para visualizar los documentos:



4.4. Exploren la funcionalidad de la herramienta.