



UNIVERSIDAD TECNOLÓGICA DE XICOTEPEC DE JUÁREZ

Universidad Tecnológica de Xicotepec de Juárez

- **Carrera:** Desarrollo de Software Multiplataforma
- **Materia:** Bases de Datos para Cómputo en la Nube
- **Profesor:** M.T.I. Marco Antonio Hernández Ramírez
- **Grado:** Quinto Cuatrimestre
- **Grupo:** B

Haziel Ortiz Ramírez
220087@utxicotepec.edu.mx

Contenido

Contenido	1
Introducción	3
Objetivo	5
Requerimientos de Hardware	6
Requerimientos de Software	6
Obtención de Instaladores	1
Proceso de Instalación	5
1.- Iniciar Instalador de MongoDB Community Server.....	5
2.- Aceptar Términos y Condiciones.....	6
3.- Elegir Tipo de Instalación.....	7
3.1.- Instalación personalizada.....	7
3.1.1.- Seleccionar qué instalar.....	8
4.- Configurar el Servicio de MongoDB.....	8
4.1.- Instalación de MongoD como Servicio.....	9
4.1.1.- Escoger entre tipo de servicio en la red.....	9
4.2.- Instalación como aplicación.....	10
5.- Comenzar la instalación de MongoDB.....	11
5.1- Instalar también MongoDB Compass.....	11
5.2.- Permisos de administrador.....	12
6.- Instalación finalizada.....	12
6.1- Comprobar servicio (En caso de haberse instalado como servicio)	13
7.- Comprobar instalación en “Aplicaciones Instaladas”.....	13
Instalación Mongosh (Shell)	14
1.- Ejecutar instalador Mongosh.....	14
2.- Inicio del instalador.....	14
3.- Instalar para todos los usuarios o solo el que lo ejecuta	14
3.1.- Instalación para todos los usuarios.....	15
.....	15
4.- Comenzar Instalación.....	15
5.- Terminar instalación.....	16

6.- Comprobar instalación.....	16
Configuración de MongoDB	17
1.- Conexión por defecto.....	17
1.1.- Conexión a MongoD como Servicio.....	17
1.2.- Conexión a MongoDB como Aplicación.....	19
1.2.2.- Ejecutar MongoD.....	20
1.2.3.- Conectarse con Mongosh.....	21
2.- Configurar un puerto personalizado.....	22
2.1.- Comprobar de nuevo la conexión.....	22
3.- Crear un usuario.....	23
3.1.- Crear Superusuario.....	24
4.- Configurar un puerto por defecto.....	25
5.- Activar autenticación por defecto.....	26
6.- Activar conexión remota.....	26
Verificación del Servicio	27
Cargar Archivo de Configuración.....	27
Conectarse con mongosh.....	27
Creación de una Base de Datos	28
Crear una colección	29
Insertar Datos	30
Consultar Datos	30
Conexión Remota	31
Conclusiones	33

Introducción

Dentro del desarrollo de software en general, siempre ha existido la necesidad de poder conservar los datos con el fin de que estos mismos sean usados ya sea para llevar un control, proceso o análisis que den lugar a información importante tanto para los usuarios como los desarrolladores.

Gracias a esta capacidad de conservar datos, es que las aplicaciones que usamos constantemente, nos ofrecen funcionalidades tan fascinantes como los sistemas de recomendaciones, la conservación de nuestra actividad durante el uso de los sistemas, entre muchas otras más.

A todo esto se le conoce como **persistencia de datos**, que con el masivo crecimiento que han tenido todos los sistemas, es que se han ido desarrollando nuevas formas de poder almacenar los datos. Gracias a esto es que surgen inicialmente formas básicas y primitivas como las **estructuras de datos** como las listas, arreglos, tuplas, etc... Siendo estas la base de tecnologías y mecanismos más avanzados como las bases de datos **SQL** (Structured Query Language - Lenguaje Estructurado de Consultas).

Estas bases de datos fueron una gran solución a la persistencia de datos, sobre todo porque ofrecen una gran robustez ante grandes cantidades de datos, permiten definir un diseño y esquema en el que se manejarán los datos, desde su inicio, su proceso y su fin. De esto surgieron tecnologías como MySQL, PostgreSQL, SQL Server, Oracle SQL, etc...

Durante muchos años, estas bases de datos fueron lo más usado y la mejor opción, pues crecieron hasta poder lograr tener un ecosistema más completo y avanzado de datos, en los que desde el mismo lenguaje SQL, se podían crear mecanismos de seguridad para los datos, automatizar tareas, procesar datos, definir estructuras avanzadas y sobre todo, garantizar la **integridad de los datos**.

Hasta que en algún punto estas bases de datos, comenzaron a quedarse cortas en rendimiento para las grandes cantidades de datos y consultas que se hacían constantemente a estas, pues nunca fueron pensadas para soportar tanta carga, haciendo que empresas tecnológicas desarrollaran sus propias soluciones como Amazon con DynamoDB.

Con esto surgen bases de datos que priorizaban más la **eficiencia y rendimiento** sobre la seguridad e integridad de los datos.

Dando lugar a las actuales famosas **Bases de Datos NoSQL** (Not Only SQL), las cuales se dividen en varios tipos de acuerdo a su manera

de organizar, administrar y almacenar los datos, entre los que destacan los sistemas de; grafos, documentales, columnares, y clave-valor.

La primer base de datos NoSQL se estima que fue creada alrededor de los años 90, cuando **Carlo Strozzi** desarrolló su propio SGDB (Sistema Gestor de Base de Datos), el cual no usaba SQL, en lugar de eso guardaba sus tablas dentro de archivos ASCII y ejecutaba Scripts UNIX para consultar los datos.

Actualmente han tomado gran importancia y se han ido modernizando hasta volver un debate sobre sí SQL o NoSQL, aunque esto es lo que menos debería importar, sabiendo que ambos tienen propósitos diferentes y que incluso pueden ser usadas ambas en un mismo sistema (una práctica común en grandes sistemas).

Objetivo

Este documento planea ser un manual para aquellas personas que buscan entrar en materia con las bases de datos NoSQL, específicamente **MongoDB**, siendo una guía de su instalación y configuración inicial de este SGDB para bases de datos basadas en documentos y que actualmente es uno de las mayores referencias de estas bases de datos NoSQL, sobre todo por su popularidad en los llamados “stacks” de JavaScript como el MERN, MEAN o MEVN.

MongoDB, no requiere de muchos conocimientos avanzados para ser instalado, sobre todo si no es el primer gestor de bases de datos que instala. Aún con ello, este manual prioriza que pueda ser fácilmente comprendido para que así tenga nula experiencia, logre instalar MongoDB en su sistema y aprenda a dar un paso más allá al configurarlo, evitando así quedarse con la configuración que suele traer por defecto, la cual es suficiente, pero es buena práctica realizarle configuraciones tanto por fines de seguridad, como con fines didácticos.

Durante esta instalación se abordarán temas como prácticas de seguridad en la configuración de MongoDB, además de que conocerá las diferentes formas que existen para tener instalado MongoDB en un sistema Windows superior a las versiones de Windows 8, 8.1. Todos los recursos serán gratuitos y obtenidos desde el sitio oficial de MongoDB para evitar que estos hayan sido manipulados por terceros, dando un inicio en la guía desde la obtención de instaladores, cómo ejecutarlos y explicar cada opción que aparecen durante el proceso de instalación de MongoDB Community Server en su última versión a 17 de enero de 2024, la versión 7.05 que está disponible como versión estable en mongodb.com.

Además de garantizar la instalación y configuración de MongoDB Community Server, también se busca explicar la instalación de su cliente gráfico oficial **MongoDB Compass**, así como el cliente oficial de línea de comandos (CLI), **Mongosh**. Con temas extra a abordar no tan comunes en las guías actuales, donde no se habla de Mongod Daemon y Mongo Sharded, temas presentes en la última versión de MongoDB, así como una breve introducción y primeros pasos en Mongo para probar el sistema, crear tu primer base de datos con el fin de conocer conceptos como: Documentos, Colecciones y Registros. Permitiendo a nuevos desarrolladores, dar sus primeros pasos en uno de los sistemas gestores de bases de datos NoSQL más populares del momento y una habilidad técnica muy importante en la actualidad para tener un perfil profesional más moderno, actualizado e interesante.

Requerimientos de Hardware

Antes de instalar MongoDB, es importante conocer los requerimientos mínimos y recomendados en cuestión de hardware por MongoDB para utilizar este sistema sin problemas de rendimiento o fallas por falta de recursos

Requisitos mínimos:

- RAM: 4GB
- Procesador: 2 núcleos
- Almacenamiento: 10 GB

Requisitos recomendados:

- RAM: 8 GB
- Procesador: 2 núcleos
- Almacenamiento: SSD 20 GB

Requerimientos de Software

Para el tema de **Sistemas Operativos**, MongoDB está disponible para MacOS, Windows y basados en Linux, sobre todo en los sistemas basados en x64 bits, más específicamente en la siguiente lista:

Linux:

- Ubuntu
- Debian
- CentOS
- Red Hat Enterprise Linux (RHEL)
- SUSE Linux Enterprise Server (SLES)

Windows:

- Windows Server
- Windows 10
- Windows 8
- Windows 7

MacOS

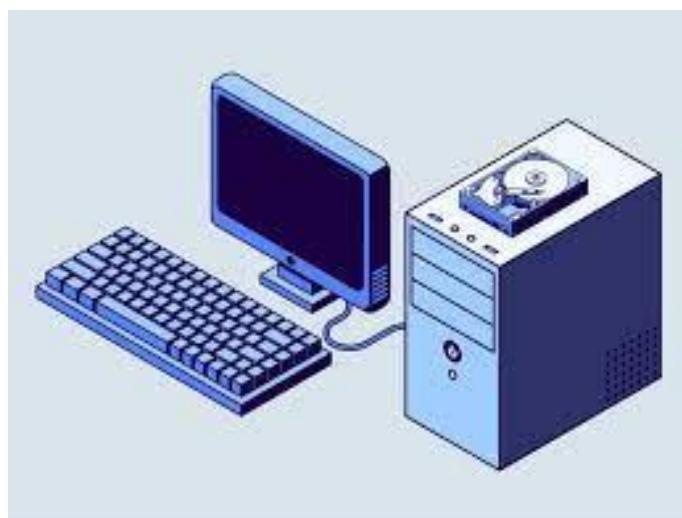


Imagen 1, Equipo de Cómputo

Obtención de Instaladores

Descargar MongoDB es un proceso sencillo y facilitado gracias a su web oficial en el apartado de [Downloads](#).

Los paquetes y programas que debemos descargar son tres principales:

- MongoDB Community Server
- Mongosh
- MongoDB Compass

Dentro del apartado de descargas para su versión [community](#) podemos encontrar:

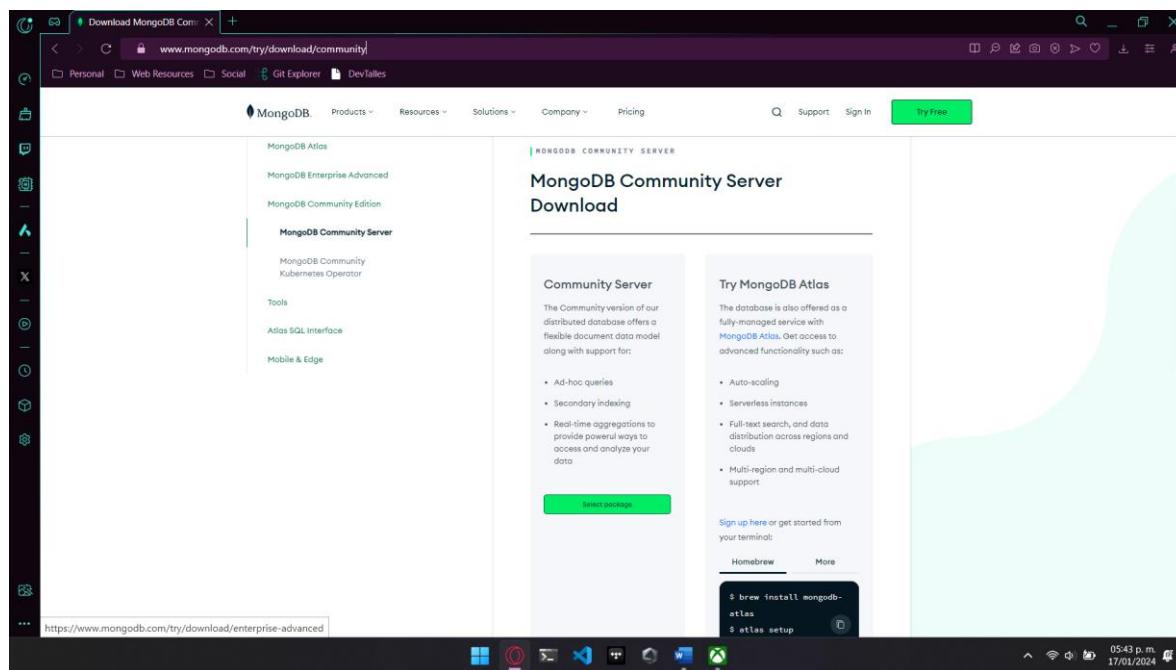


Imagen 2, Página de Descargas de MongoDB

Solo es necesario seleccionar en “Select Package” y se desplegará un menú para seleccionar la versión, formato y plataforma que requerimos:

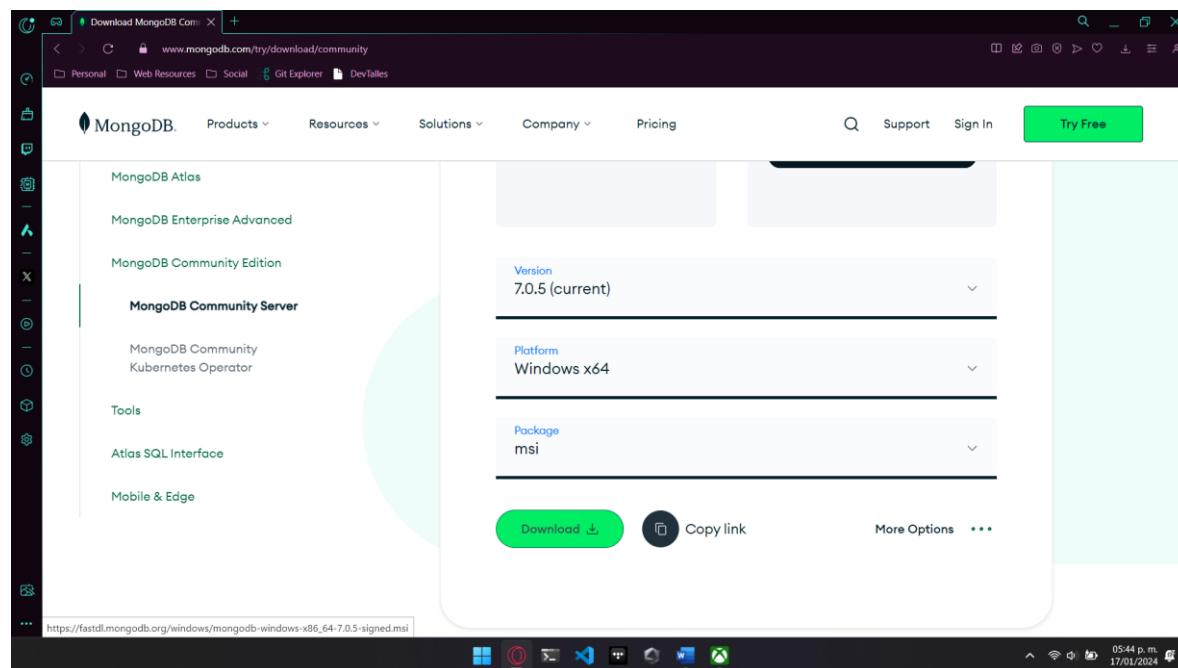


Imagen 3, Menú de descarga de MongoDB CS

En este caso se usará el paquete directo .msi (MicroSoft Installer) para poder ejecutarlo directamente sin tener que descomprimirlo. Al presionar “Download” comenzará la descarga del paquete:

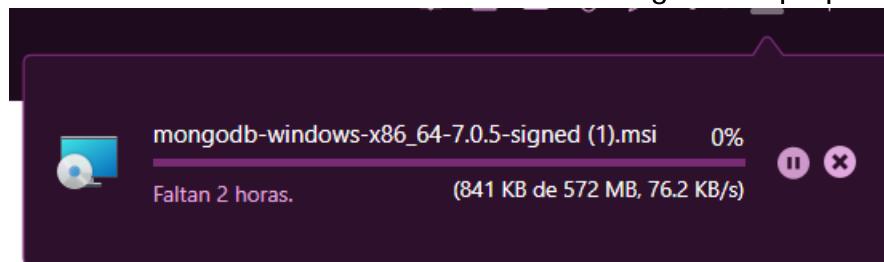


Imagen 4, Proceso de descarga de MongoDB

Por último es necesario descargar **Mongosh** para poder conectarse a MongoDB mediante su cliente **CLI**.

Dentro de la página encontramos lo siguiente:

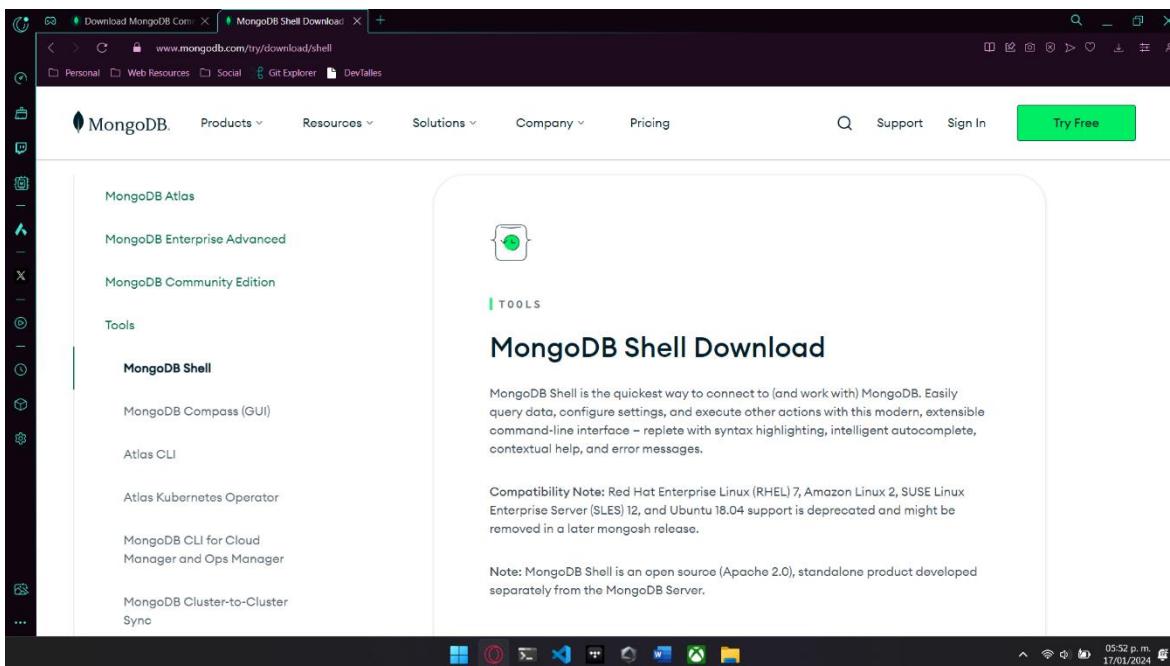


Imagen 5, Sitio oficial de descargas de Mongosh

Más abajo encontramos el menú para elegir el paquete que queremos descargar:

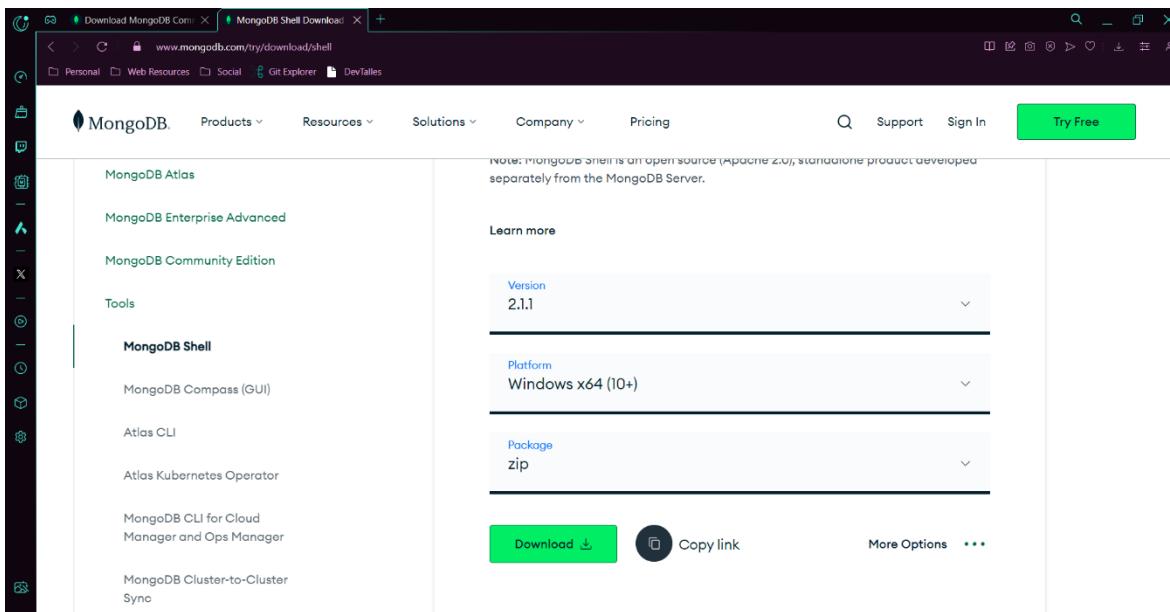


Imagen 6, menú de descarga de mongosh

La única recomendación, es seleccionar en “Package” el paquete **.msi** en lugar de **zip** para evitar tener que descomprimirlo.

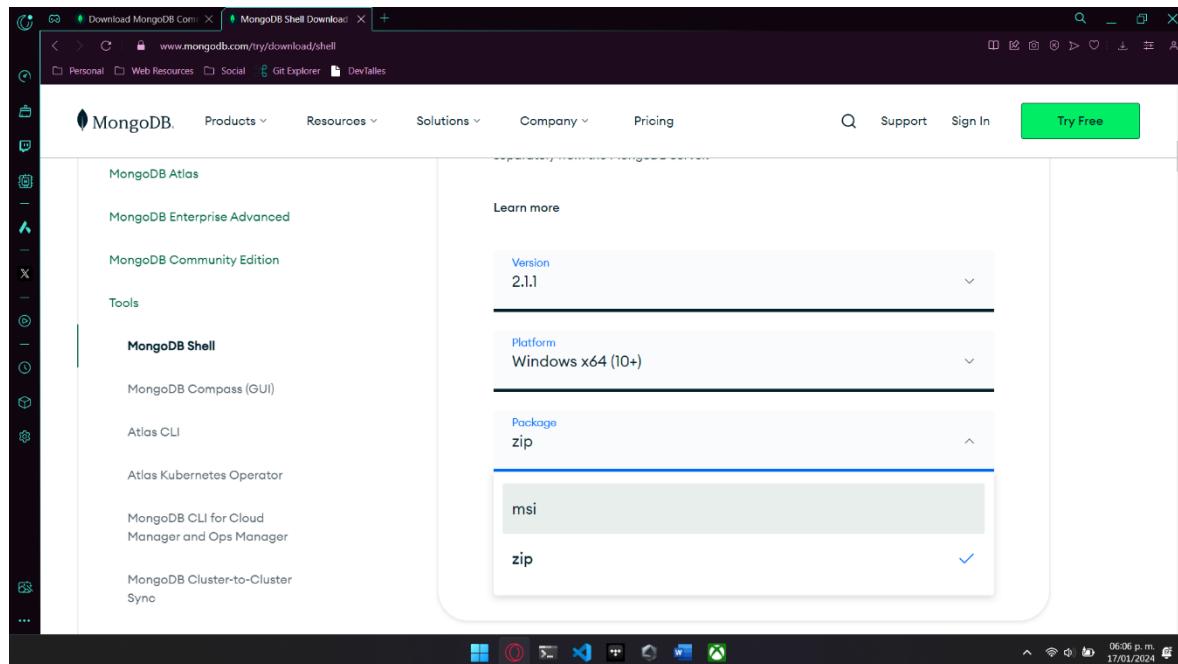


Imagen 7, selección de tipo de paquete a descargar

Al iniciar la descarga, debería salir algo como:

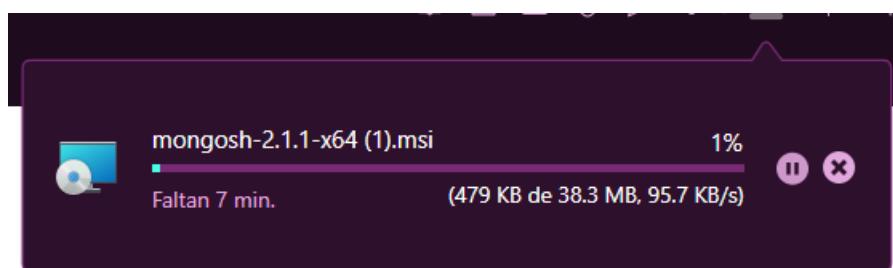


Imagen 8, proceso de descarga de mongosh

Proceso de Instalación

Una vez contamos ya con los instaladores necesarios para MongoDB (tanto el community server 7.05 como el Mongosh 2.1, opcionalmente el instalador de MongoDB Compass, este viene incluido en Community Server), podemos proseguir con la ejecución e instalación de MongoDB.

 mongodb-windows-x86_64-7.0.5-signed.msi	08/01/2024 06:38 p. m.	Paquete de Windo...	585,804 KB
 mongosh-2.1.1-x64.msi	07/01/2024 11:27 a. m.	Paquete de Windo...	39,216 KB

Imagen 9, vista de los paquetes descargados

Al instalar MongoDB Community Server, tenemos dos maneras de tener instalado este sistema y es cómo **Servicio** o como **Aplicación**. Ambos permiten usar correctamente y sin problemas a MongoDB, la principal diferencia recae en que al tenerlo instalado como Aplicación, este estará constantemente levantado y corriendo en segundo plano esperando conexiones, mientras que instalarlo como Servicio, implica levantar manualmente su proceso Daemon o demonio el cual se encarga de esperar conexiones. Ambos serán abordados cuando llegue el momento de elegir una opción.

1.- Iniciar Instalador de MongoDB Community Server

Primero ejecutaremos el instalador de MongoDB Community Server, el cual suele tener el siguiente nombre:

 mongodb-windows-x86_64-7.0.5-signed.msi	08/01/2024 06:38 p. m.	Paquete de Windo...	585,804 KB
---	------------------------	---------------------	------------

Imagen 10, paquete descargado de MongoDB Community Server

Una vez lo ejecutemos, comenzará a cargar el instalador y mostrará la siguiente ventana emergente:

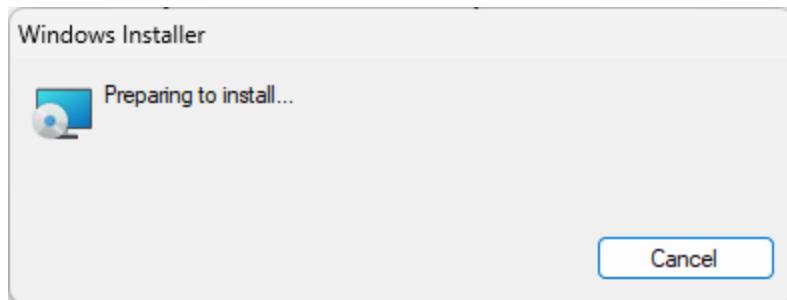


Imagen 11, cargando el instalador

Una vez el instalador haya terminado de cargar, mostrará la siguiente ventana de bienvenida al instalador donde solo es necesario pulsar en **Next**:



Imagen 12, inicio del instalador de MongoDB

2.- Aceptar Términos y Condiciones

Para continuar con la instalación donde aceptaremos los **Términos y Condiciones** para poder continuar con la instalación:

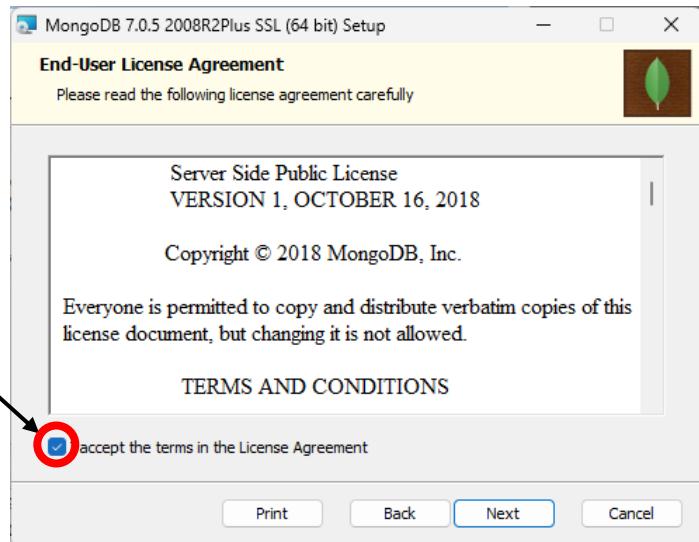


Imagen 13, términos y condiciones de MongoDB

3.- Elegir Tipo de Instalación

Posteriormente nos mostrará dos opciones para instalar MongoDB, en las que se puede observar **Complete** o **Custom**. Complete implica instalar automáticamente todo lo necesario de MongoDB, como MongoDB Compass, Mongo Router y Mongo Server.

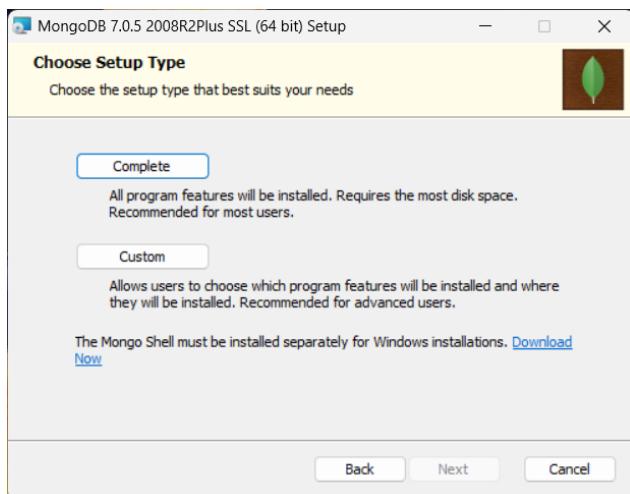


Imagen 14, elegir tipo de instalación

3.1.- Instalación personalizada

Sí seleccionamos **Custom**, nos mostrará un árbol de opciones con los complementos y programas que cuenta el instalador:

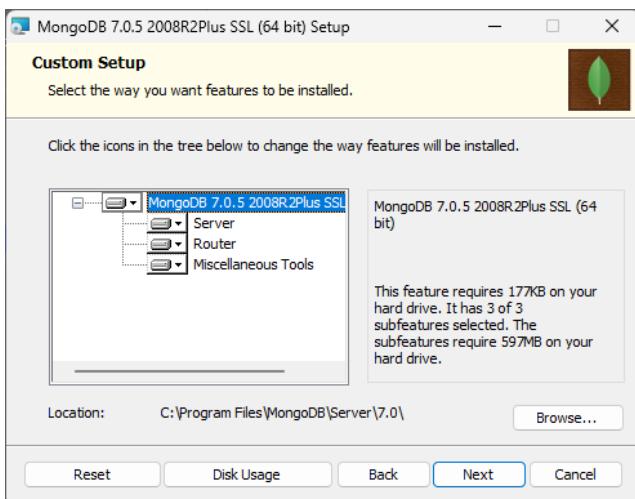


Imagen 15, selección manual de paquetes a instalar

3.1.1.- Seleccionar qué instalar

Para elegir que instalar y qué no, solo es necesario pulsar en cada opción del árbol y dará una lista de opciones, entre las que podemos ver:

- Instalar en el disco local
- Instalar completamente en el disco local
- Instalar para correr desde una red.
- Instalar completamente para correr desde una red
- Instalar cuando sea requerido
- Desactivar instalación

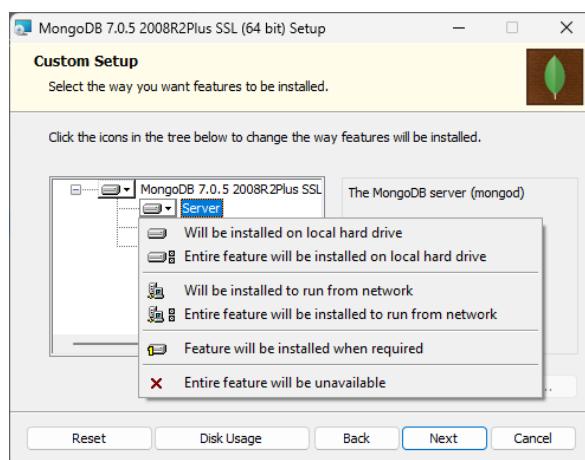


Imagen 16, opciones para cada paquete a instalar

4.- Configurar el Servicio de MongoDB

Este punto es muy importante, pues aquí definimos si el proceso Daemon o Demonio de Mongo (MongoD) será instalado como **Aplicación** o como un **Servicio**, previamente se comentó un poco sobre las diferencias.

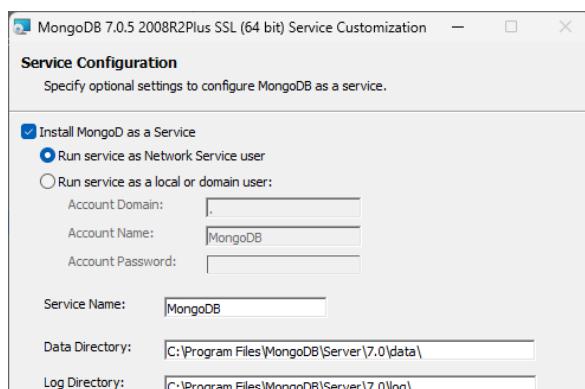


Imagen 17, configuración de MongoDB como servicio

4.1.- Instalación de MongoDB como Servicio

En esta pantalla podemos observar varias opciones, para comenzar, si requerimos instalarlo como un Servicio que esté constantemente levantado, dejaremos seleccionada la casilla de “*Install MongoDB as a Service*”

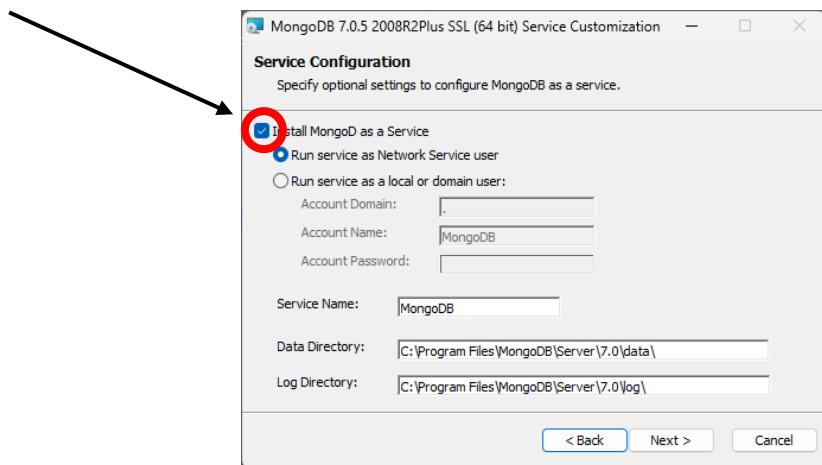


Imagen 18, configuración como servicio

4.1.1.- Escoger entre tipo de servicio en la red

MongoDB nos permite elegir entre ejecutar el servicio como un usuario en la red o configurar el servicio como un usuario de red local o con dominio personalizado.

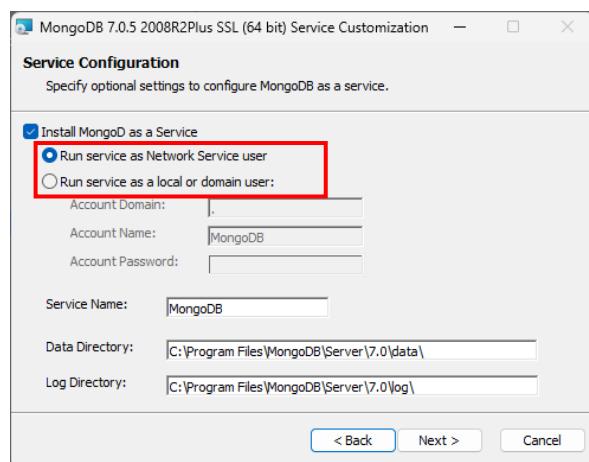


Imagen 19, opciones de usuario como servicio

4.2.- Instalación como aplicación

En caso de que se quiera ejecutar manualmente el Daemon o demonio MongoD, se debe desmarcar la casilla de “*Install MongoDB as a Service*”:

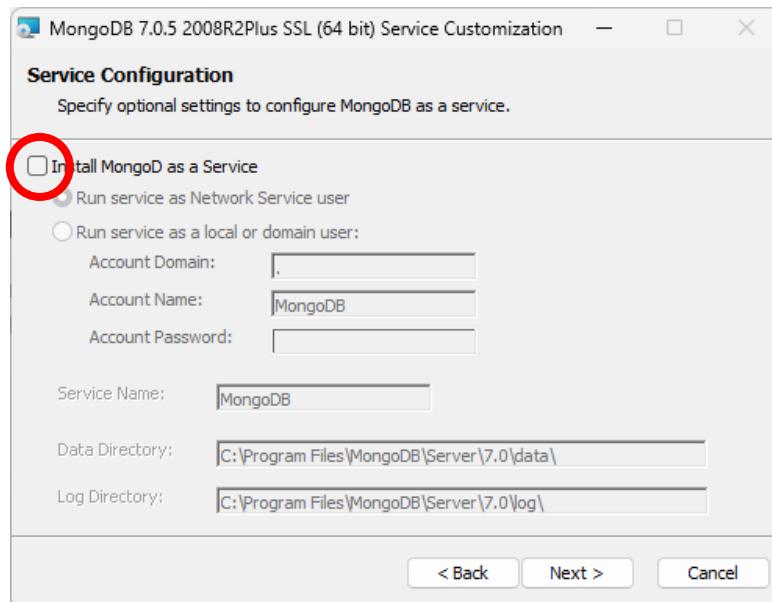


Imagen 20, selección de mongod como aplicación

5.- Comenzar la instalación de MongoDB

Después de todos estos pasos y “configuraciones” para instalar Mongo, nos aparecerá esta ventana:

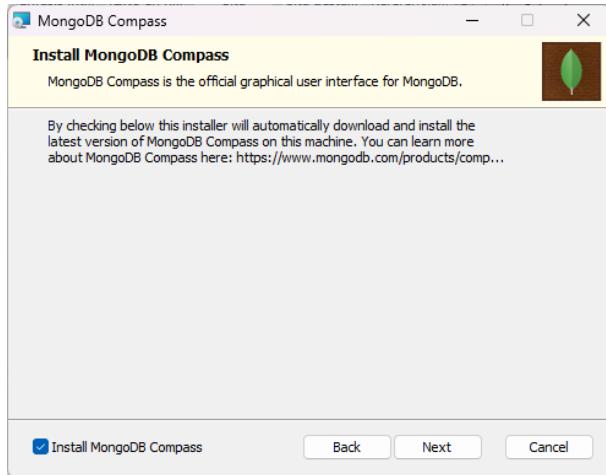


Imagen 21, inicio de la instalación

5.1- Instalar también MongoDB Compass

En este paso se nos marca por defecto la casilla de instalar también el cliente gráfico oficial de MongoDB, sin embargo en caso de requerir instalarlo por separado puede desmarcarse esta casilla sin afectar la instalación

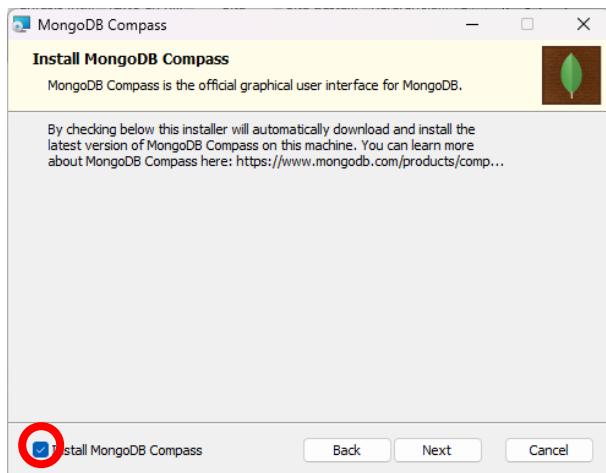


Imagen 22, instalar adicionalmente Mongo Compass

5.2.- Permisos de administrador

Por último, al dar “next” se nos solicitará continuar con permisos de administrador y que el asistente de instalación de MongoDB pueda realizar los cambios pertinentes en el sistema:

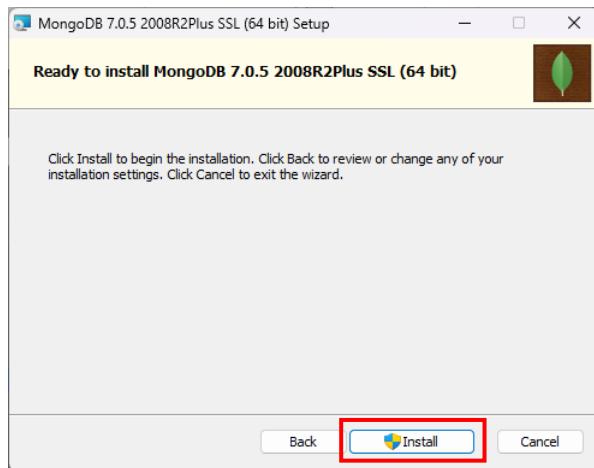


Imagen 23, solicitud de permisos de administrador

6.- Instalación finalizada

Luego de dar permisos de administrador y que terminará el proceso de instalación, el proceso terminará con la siguiente ventana:



Imagen 24, ventana de instalación finalizada

6.1- Comprobar servicio (En caso de haberse instalado como servicio)

Sí elegiste instalar MongoD como un servicio, debería poder ser comprobado que está constantemente ejecutándose desde el administrador de tareas:

7.- Comprobar instalación en “Aplicaciones Instaladas”

Luego de finalizar la instalación de MongoDB, al entrar en la configuración de Windows en su apartado de Aplicaciones>Aplicaciones instaladas, debería aparecer lo siguiente al buscar por “mongo”:

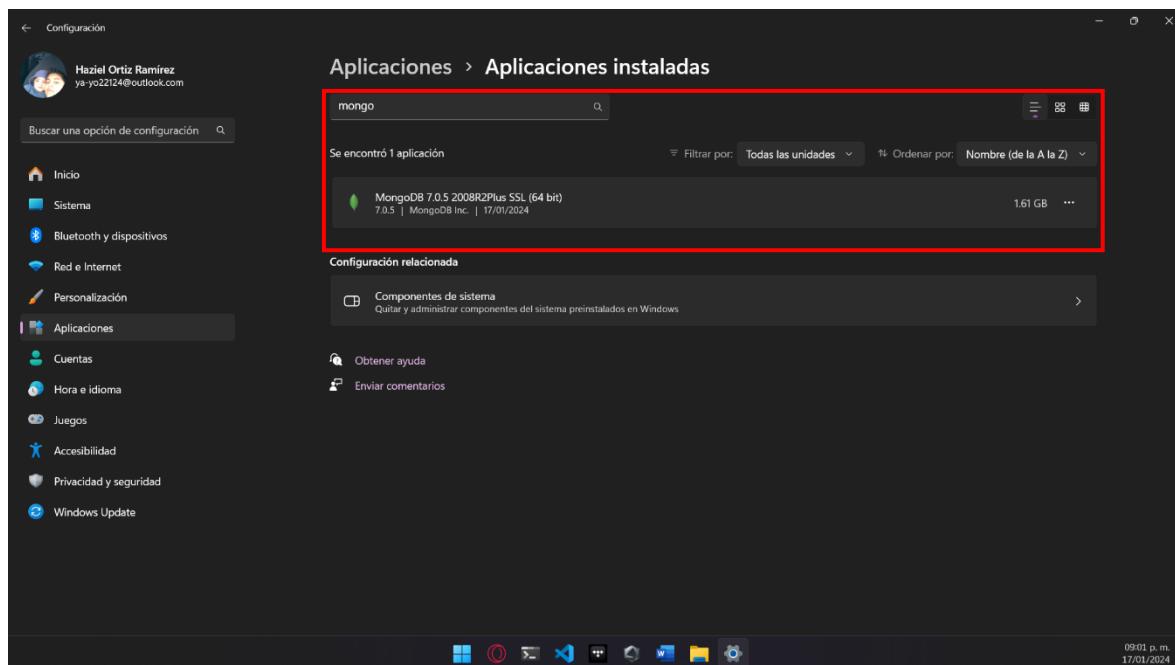


Imagen 25, configuración de Windows, mongodb instalado

Instalación Mongosh (Shell)

Mongosh es el cliente oficial de línea de comandos (CLI), el cual su instalación es bastante sencilla:

1.- Ejecutar instalador Mongosh

Lo primero es ubicar el instalador previamente descargado de mongosh, el cual debe verse algo parecido a:



Imagen 26, paquete instalador de mongosh

2.- Inicio del instalador

Al ejecutar el instalador, lo primero que vemos es una ventana emergente como la siguiente:

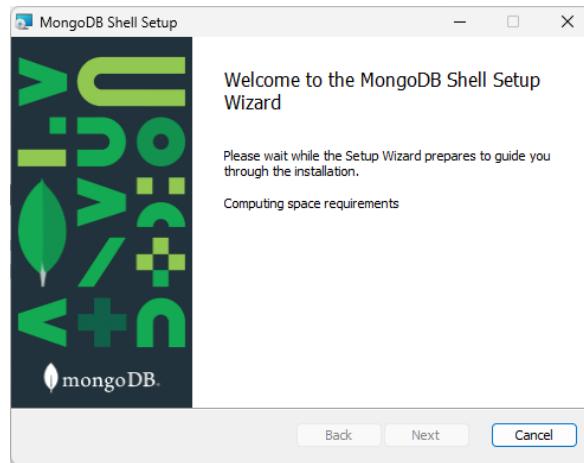


Imagen 27, ventana de bienvenida al instalador

3.- Instalar para todos los usuarios o solo el que lo ejecuta

El siguiente paso consiste en elegir la ruta de instalación de mongosh, la cual ya viene configurada por defecto en "C:\Users\<user>\AppData\Local\Programs\mongosh\" y además escoger si la instalación sea global o solo para tu usuario:

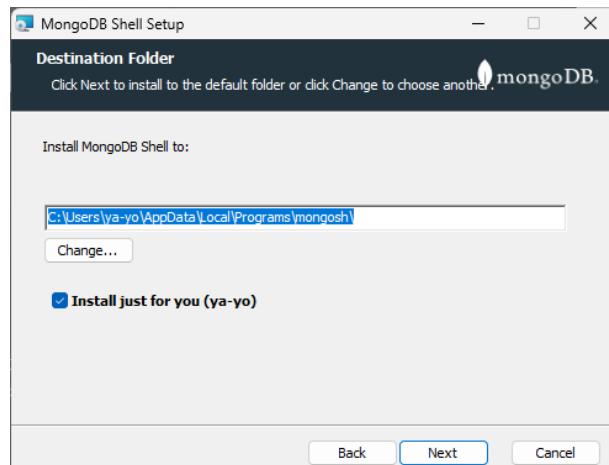


Imagen 28, ruta de instalación

3.1.- Instalación para todos los usuarios

Seleccionar esta opción requiere de permisos de Administrador, pues la ruta cambia a “C:\Program Files\mongosh”, la cual está protegida y por ello requiere permisos especiales:

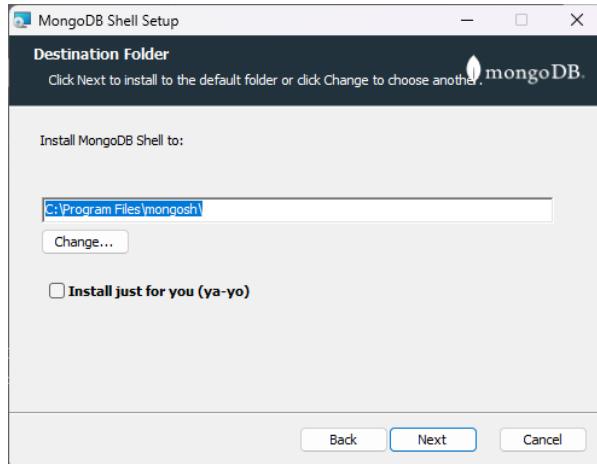


Imagen 29, opción para instalar en todos los usuarios

4.- Comenzar Instalación

Luego de lo anterior, comenzaremos la instalación al pulsar sobre “Install”:

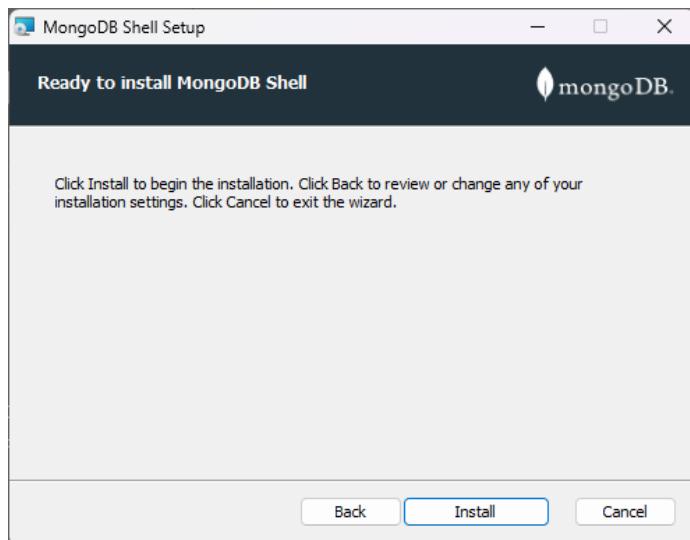


Imagen 30, iniciar instalación de mongosh

5.- Terminar instalación

Luego de que el proceso termine, se nos mostrará la siguiente ventana indicando su finalización:

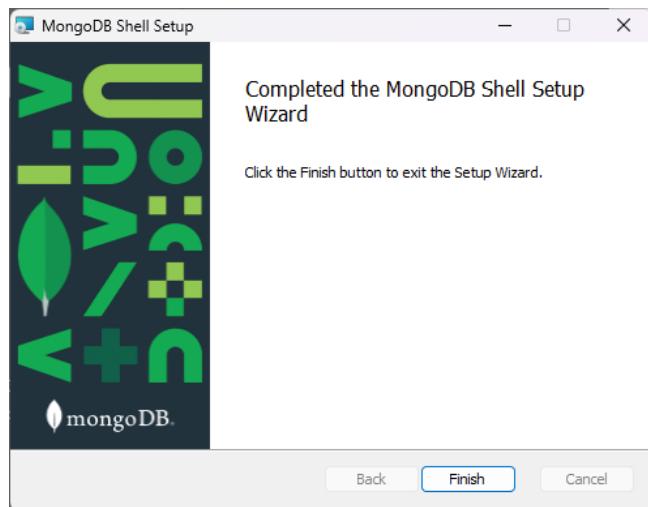


Imagen 31, finalización de la instalación

6.- Comprobar instalación

Sí nos vamos a la configuración de Windows en Aplicaciones>Aplicaciones instaladas, y buscamos por “mongo” ahora debería aparecer un programa nuevo:

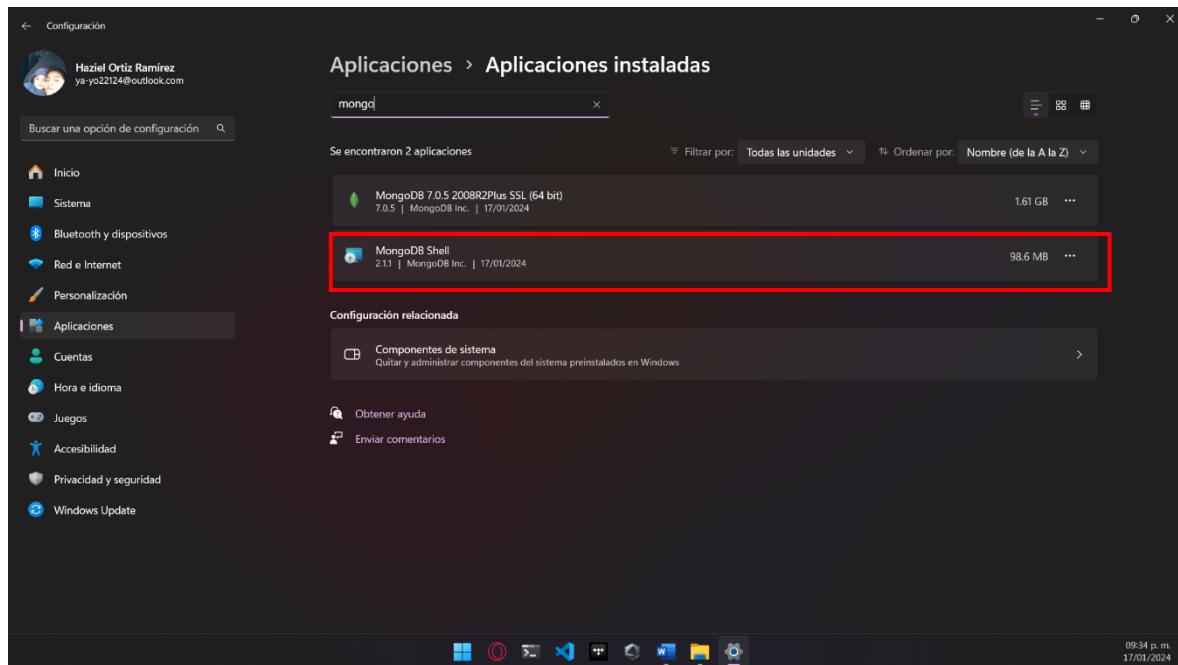


Imagen 32, comprobación de instalación de mongosh

Configuración de MongoDB

Luego de haber instalado MongoDB, MongoDB Compass y MongoSH, podemos seguir a algunos pasos extra de configuración que por defecto nos son establecidos como el Puerto, Usuario y Contraseña.

1.- Conexión por defecto

Antes de configurar MongoDB, primero debemos probar que este ya funcione y reciba conexiones, lo que depende si la instalación de MongoDB se realizó como **Servicio** o como **Aplicación**:

1.1.- Conexión a MongoDB como Servicio

Esta forma es más sencilla pues MongoDB está ejecutándose de fondo, esto puede comprobarse asistiendo al *Task Manager* de Windows:

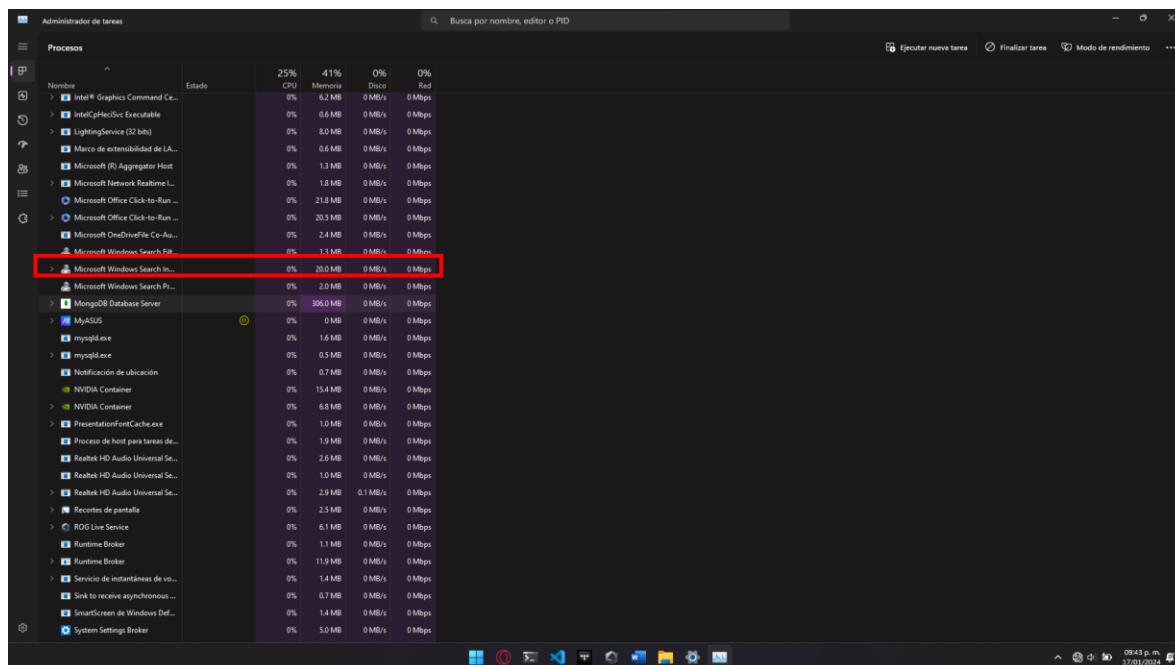
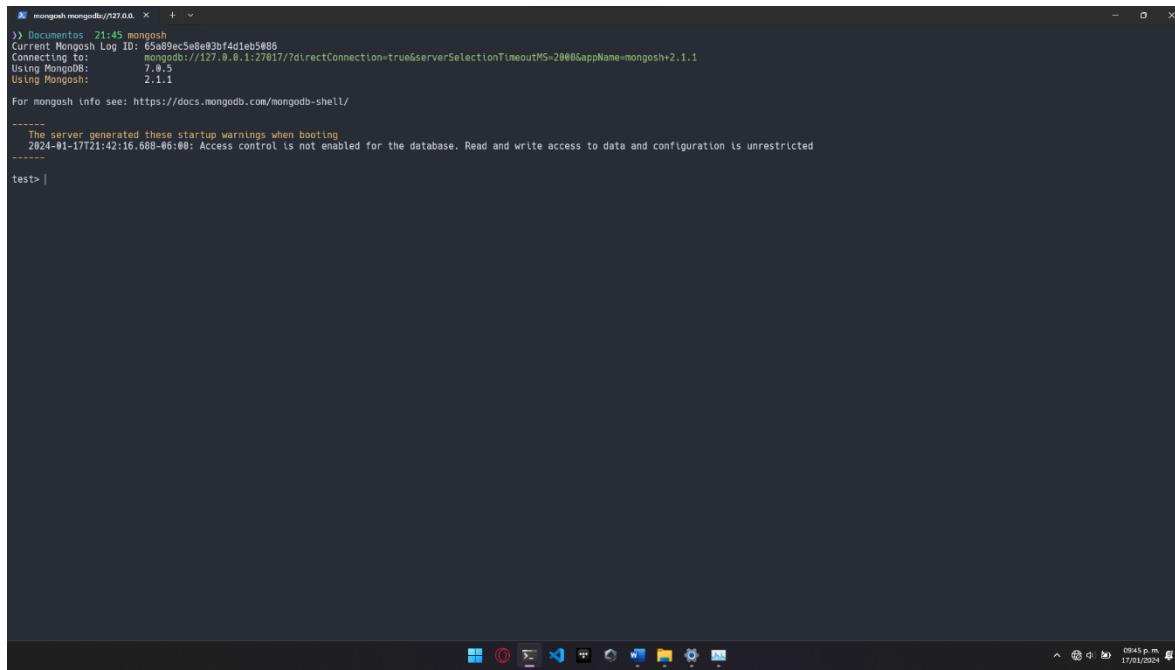


Imagen 33, comprobación del servicio MongoDB en TaskManager

Con esto claro, ahora solo hace falta abrir una terminal CMD o Powershell y ejecutar **mongosh**:



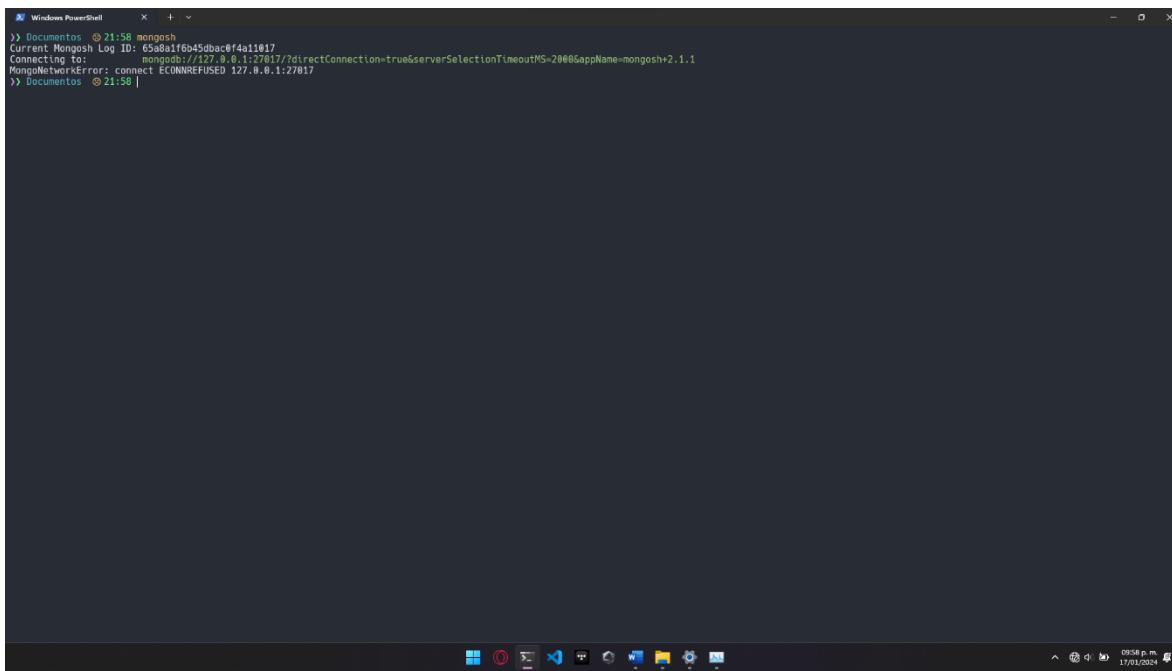
```
mongosh mongod:127.0.0.1:27017
> Current Mongosh Log ID: 65a89ec5e8e03bf4d1eb5#86
Connecting to:      mongod://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.1
Using MongoDB:     7.0.5
Using Mongosh:    2.1.1
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
-----
The server generated these startup warnings when booting
2024-01-17T21:42:16.688-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
-----
test> |
```

Imagen 34, conexión vía mongosh

Como podemos ver, automáticamente y sin necesidad de meter credenciales, se nos da acceso a *test* la base de datos por defecto seleccionada en MongoDB al iniciarla.

1.2.- Conexión a MongoDB como Aplicación

Esta forma de instalar MongoDB implica tener que levantar su proceso Daemon o Demonio (MongoD) manualmente cada que se quiera realizar conexión al mismo. En caso de probar a ejecutar mongosh sin levantar MongoD, pasa lo siguiente:



```
Windows PowerShell x + ~
>> Documentos 21:58 mongosh
Current Mongosh Log ID: 65a8a1f6b45dbac0f4a11017
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.1
MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017
>> Documentos 21:58 |
```

Imagen 35, conexión sin levantar mongod

Donde se nos devuelve el error de: *MongoNetworkError: connect ECONNREFUSED 127.0.0.1:27017*, ya que no existe proceso en segundo plano que este esperando conexiones y por ende nunca se conecta.

1.2.1.- Crear carpeta DATA

Para que mongo almacene la información y bases de datos, es necesario crearle manualmente las carpetas de data y db. Estás se colocan a nivel raíz del disco principal de instalación (normalmente C:):

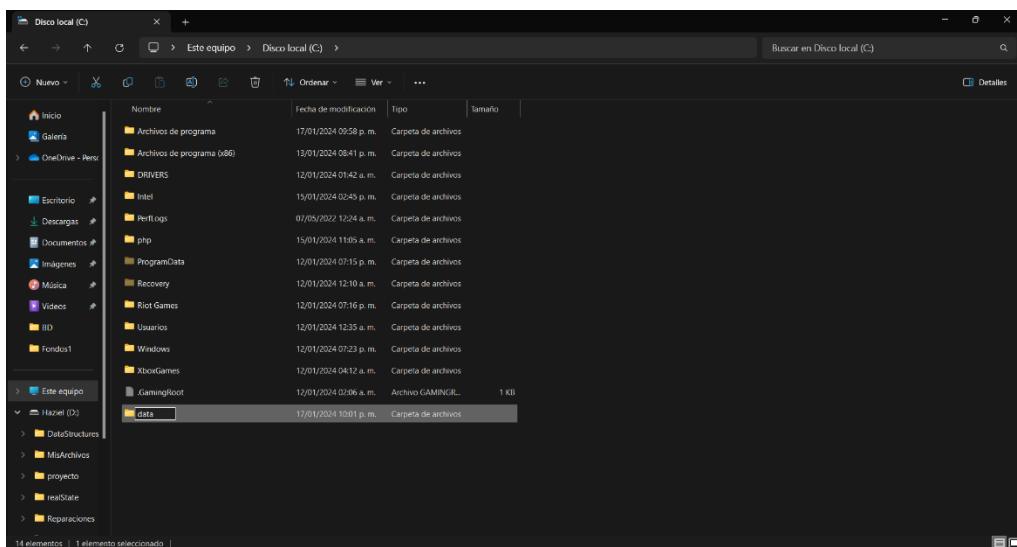


Imagen 36, creación de la carpeta data

Luego dentro de data, debemos crear db:

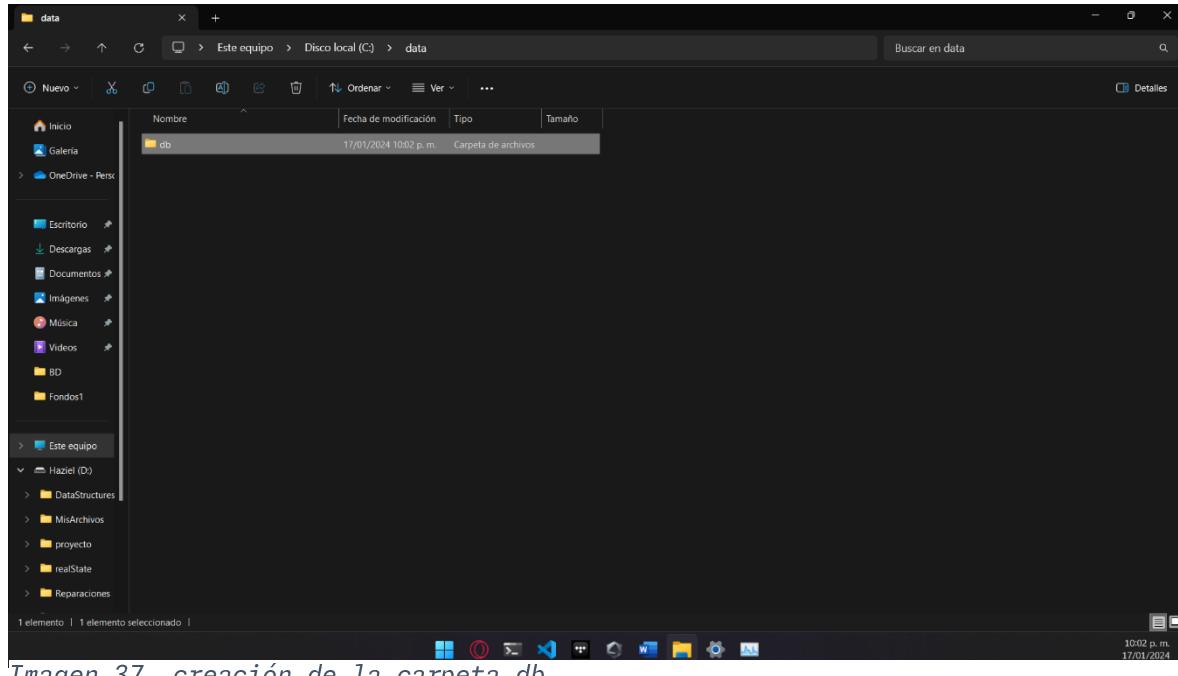


Imagen 37, creación de la carpeta db

1.2.2.- Ejecutar MongoDB

Para ejecutar el proceso demonio, debemos movernos a la carpeta donde se encuentra ese ejecutable, generalmente es en “*C:\Program Files\MongoDB\Server\7.0\bin*” una vez dentro, abrimos una terminal ahí y ejecutamos **mongod**:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

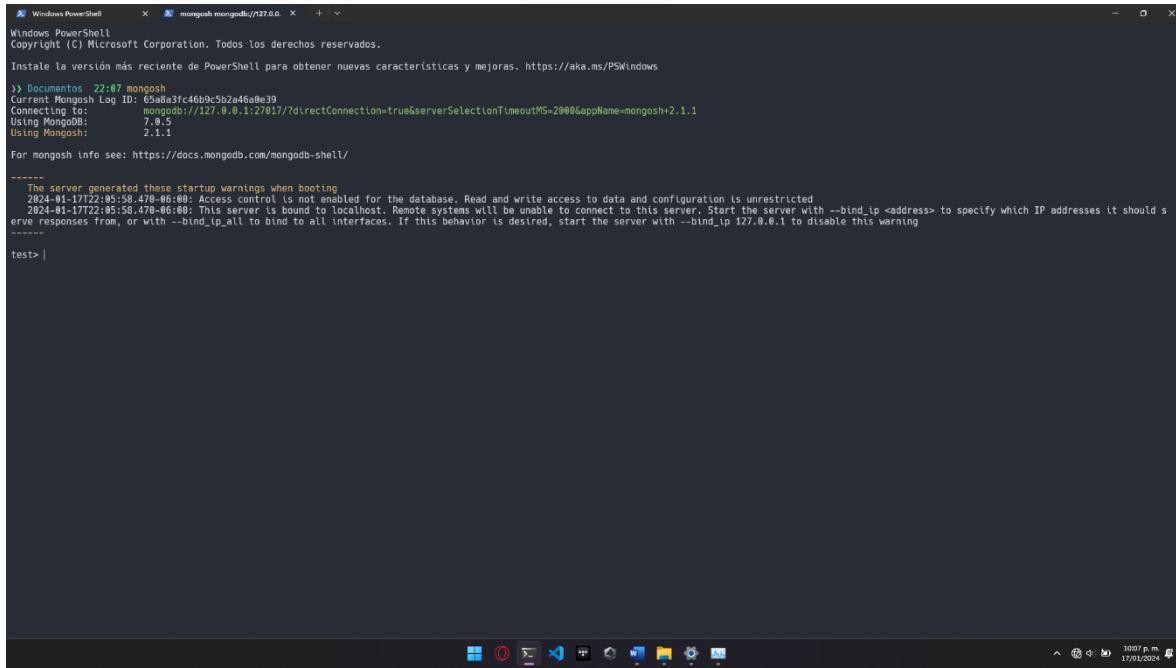
Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

>>> Documento 22:05 cd "C:\Program Files\MongoDB\Server\7.0\bin"
>>> bin> mongod --version
{
  "$date": "2024-01-17T22:05:56.746-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 22385,
  "ctx": "thread1",
  "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols \"none\""
}
{
  "$date": "2024-01-17T22:05:58.133-06:00",
  "s": "I",
  "c": "NETWORK",
  "id": 4915701,
  "ctx": "thread1",
  "msg": "Initialized wire specification",
  "attr": {
    "spec": {
      "incomingExternalClient": {
        "minWireVersion": 8,
        "maxWireVersion": 21
      },
      "incomingInternalClient": {
        "minWireVersion": 8,
        "maxWireVersion": 21
      }
    }
  }
}
{
  "$date": "2024-01-17T22:05:58.134-06:00",
  "s": "I",
  "c": "NETWORK",
  "id": 4648692,
  "ctx": "thread1",
  "msg": "Implicit TCP FastOpen in use."
}
{
  "$date": "2024-01-17T22:05:58.135-06:00",
  "s": "I",
  "c": "REPL",
  "id": 5123680,
  "ctx": "thread1",
  "msg": "Successfully registered PrimaryOnlyService",
  "attr": {
    "service": "TenantMigrationDonorService",
    "namespace": "config.tenantMigrationOnDonor"
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "REPL",
  "id": 5123680,
  "ctx": "thread1",
  "msg": "Successfully registered TenantMigrationRecipientService",
  "attr": {
    "service": "TenantMigrationRecipientService",
    "namespace": "config.tenantMigrationRecipients"
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 5845082,
  "ctx": "thread1",
  "msg": "Initial threading initialized"
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "RENTANT_M",
  "id": 7691690,
  "ctx": "thread1",
  "msg": "Starting TenantManagementAccessBlockerRegistry"
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 4615611,
  "ctx": "initandlisten",
  "msg": "MongoDB starting",
  "attr": {
    "pid": 21464,
    "port": 27017,
    "dbPath": "C:/data/db/",
    "architecture": "64-bit",
    "host": "ASUS-TUF"
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 23398,
  "ctx": "initandlisten",
  "msg": "Target operating system minimum version",
  "attr": {
    "targetMinOS": "Windows 7/Windows Server 2008"
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 23483,
  "ctx": "initandlisten",
  "msg": "BuildInfo",
  "attr": {
    "buildInfo": {
      "version": "7.0.5",
      "gitVersion": "7080d71e84e314b497f282ea8aa06d7de2b295",
      "modules": [],
      "allocator": "tcmalloc",
      "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
      }
    }
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 51765,
  "ctx": "initandlisten",
  "msg": "Operating System",
  "attr": {
    "os": {
      "name": "Microsoft Windows 10",
      "version": "10.0 (build 22631)"
    }
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 21951,
  "ctx": "initandlisten",
  "msg": "Options set by command line",
  "attr": {
    "options": {}
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "STORAGE",
  "id": 22276,
  "ctx": "initandlisten",
  "msg": "Storage engine to use detected by file system",
  "attr": {
    "dbpath": "C:/data/db/",
    "storageEngine": "wiredTiger"
  }
}
{
  "$date": "2024-01-17T22:05:58.136-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 22315,
  "ctx": "initandlisten",
  "msg": "Options",
  "attr": {
    "create_cache_size": 7598M,
    "session_max": 33800,
    "eviction_min": 4,
    "threads_max": 4,
    "conflict_base_file": false,
    "stats_file": false,
    "log_rewind": true,
    "remove": true,
    "path": "journal",
    "compressor": "snappy",
    "builtin_extension_config": {
      "zfizile": 1,
      "compaction_progress": 1,
      "compaction_progress": 1,
      "backup": 8,
      "checkpoint": 8,
      "compact": 8,
      "evict": 8
    },
    "history_store": "store",
    "recovery": 8,
    "salvage": 8,
    "tiered": 8,
    "timestamp": 8,
    "transaction": 8,
    "log_rewind": true,
    "remove": true,
    "path": "journal",
    "compressor": "snappy",
    "builtin_extension_config": {
      "zfizile": 1,
      "compaction_progress": 1,
      "compaction_progress": 1,
      "backup": 8,
      "checkpoint": 8,
      "compact": 8,
      "evict": 8
    },
    "history_store": "store",
    "recovery": 8,
    "salvage": 8,
    "tiered": 8,
    "timestamp": 8,
    "transaction": 8
  }
}
{
  "$date": "2024-01-17T22:05:58.464-06:00",
  "s": "I",
  "c": "STORAGE",
  "id": 4795986,
  "ctx": "initandlisten",
  "msg": "WiredTiger opened",
  "attr": {
    "duration": 324
  }
}
{
  "$date": "2024-01-17T22:05:58.464-06:00",
  "s": "I",
  "c": "RECOVERY",
  "id": 23087,
  "ctx": "initandlisten",
  "msg": "WiredTiger recoveryTimestamp",
  "attr": {
    "recoverTimestamp": "$timestamp"
  }
}
{
  "$date": "2024-01-17T22:05:58.464-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 22212,
  "ctx": "initandlisten",
  "msg": "Access control is not enabled for the database. Read and write access to data and configuration is unrestricted."
}
{
  "$date": "2024-01-17T22:05:58.470-06:00",
  "s": "W",
  "c": "CONTROL",
  "id": 22249,
  "ctx": "initandlisten",
  "msg": "This server is bound to localhost. If remote systems will be unable to connect to this server. Start the server with --bind_ip address to specify which IP addresses it should receive responses from. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning."
}
{
  "$date": "2024-01-17T22:05:58.473-06:00",
  "s": "I",
  "c": "STORAGE",
  "id": 5071100,
  "ctx": "initandlisten",
  "msg": "Clearing temp directory"
}
{
  "$date": "2024-01-17T22:05:58.474-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 6668180,
  "ctx": "initandlisten",
  "msg": "Initializing cluster server parameters from disk"
}
{
  "$date": "2024-01-17T22:05:58.474-06:00",
  "s": "I",
  "c": "CONTROL",
  "id": 20536,
  "ctx": "initandlisten",
  "msg": "Flame Control is enabled on this deployment"
}
{
  "$date": "2024-01-17T22:05:58.836-06:00",
  "s": "W",
  "c": "FDC",
  "id": 2318,
  "ctx": "initandlisten",
  "msg": "Failed to initialize Performance Counters for FDC"
}
{
  "$date": "2024-01-17T22:05:58.837-06:00",
  "s": "I",
  "c": "FDC",
  "id": 20625,
  "ctx": "initandlisten",
  "msg": "Initializing full-time diagnostic data capture"
}
{
  "$date": "2024-01-17T22:05:58.839-06:00",
  "s": "I",
  "c": "REPL",
  "id": 60915317,
  "ctx": "initandlisten",
  "msg": "Setting new configuration state"
}
{
  "$date": "2024-01-17T22:05:58.840-06:00",
  "s": "I",
  "c": "STORAGE",
  "id": 22262,
  "ctx": "initandlisten",
  "msg": "Timestamp monitor starting"
}
```

Imagen 38, levantamiento de mongod

1.2.3.- Conectarse con Mongosh

Una vez levantado MongoD, ya podemos hacer una conexión mediante MongoSH:



A screenshot of a Windows PowerShell window titled "Windows PowerShell" and "mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.1". The window displays the following text:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

>> Documentos 22:07 mongosh
Current Mongosh Log ID: 65aa8a3fc46b9c5b2a46a8e39
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.1
Using MongodB: 7.0.5
Using Mongosh: 2.1.1

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

The server generated these startup warnings when booting
2024-01-17T22:05:58.470-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-01-17T22:05:58.470-06:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----
test> |
```

Imagen 39, conexión a mongod

2.- Configurar un puerto personalizado

Sí queremos cambiar el puerto por defecto **27017** de mongoDB, esto podría ser una buena práctica y hasta podría mejorar un poco la seguridad al ocultar algo tan importante. Tan solo basta con usar el parámetro **--port** al levantar el MongoD y especificar un puerto:

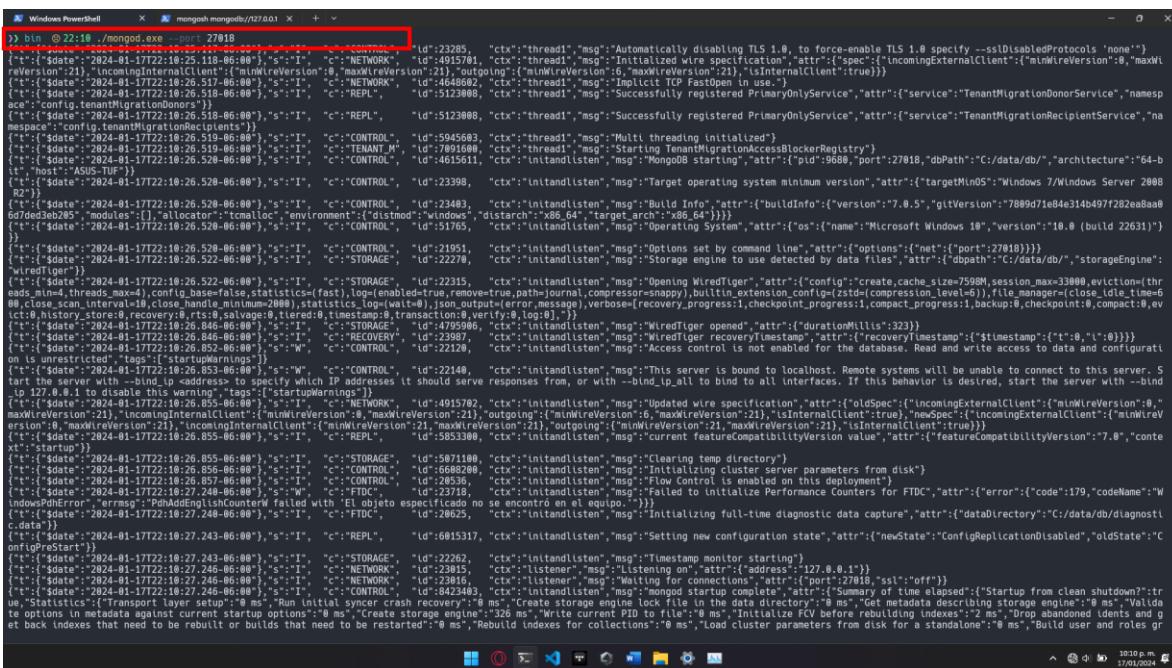


Imagen 40, levantando mongod con otro puerto

2.1.- Comprobar de nuevo la conexión

Con este cambio de puerto, no debería permitirse la conexión por defecto de Mongosh la cual usa 27017 como puerto:

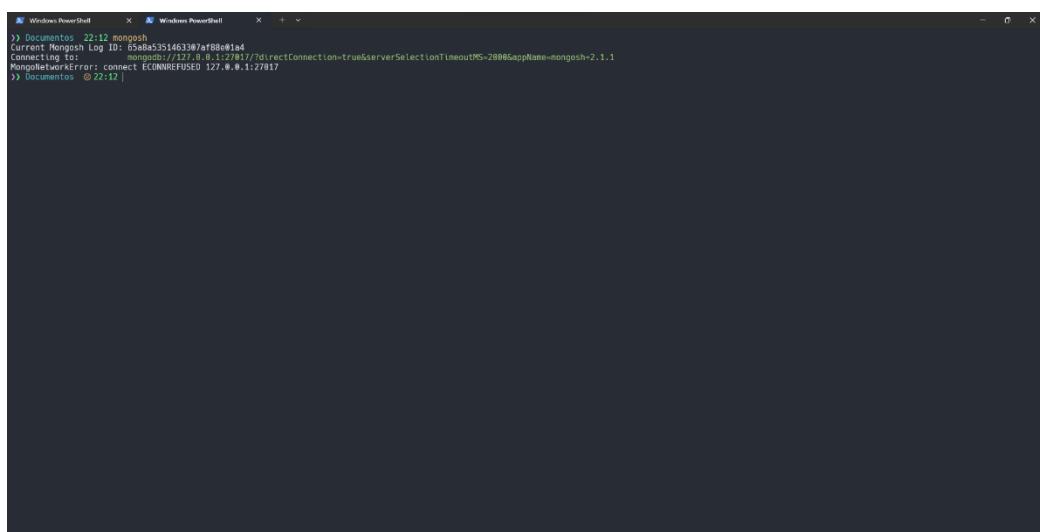
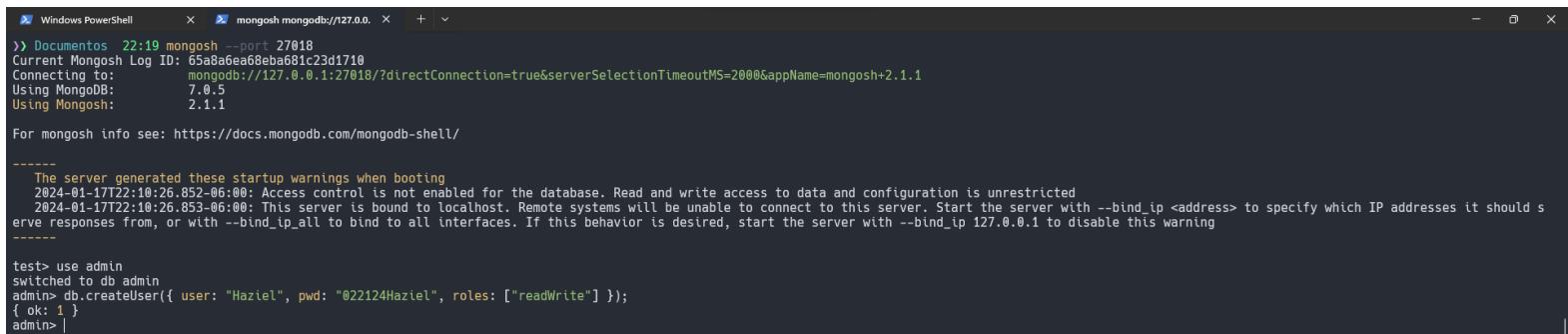


Imagen 41, conexión fallida por defecto de mongosh

3.- Crear un usuario

Los usuarios en MongoDB, son algo que por defecto trae configurado con un usuario “root” llamado aquí como **Admin**, para crear un usuario, primero es necesario indicar la base de datos, en caso de que se requiera crear con permisos de Admin, se debe crear dentro de la base de datos **admin**:



```
Windows PowerShell x mongosh mongodb://127.0.0.1:27018
Current Mongosh Log ID: 65a8a6ea68ba681c23d1710
Connecting to: mongodb://127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.1.1
Using MongoDB: 7.0.5
Using Mongosh: 2.1.1

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

-----
The server generated these startup warnings when booting
2024-01-17T22:10:26.852-06:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-01-17T22:10:26.853-06:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
-----

test> use admin
switched to db admin
admin> db.createUser({ user: "Haziel", pwd: "022124Haziel", roles: [ "readWrite" ] });
{ ok: 1 }
admin> |
```

Imagen 42, creación de un usuario

Para crear al usuario se usa **db.createUser()** el cual recibe un objeto con la configuración del usuario:

- **user**: Para especificar el nombre del usuario
- **pwd**: Para colocarle una contraseña
- **roles**: Es necesario para indicar que permisos se le quieren otorgar al usuario como **readWrite** para lectura y escritura.

Sí ejecutamos un **db.getUsers()**, debe aparecer nuestro nuevo usuario:

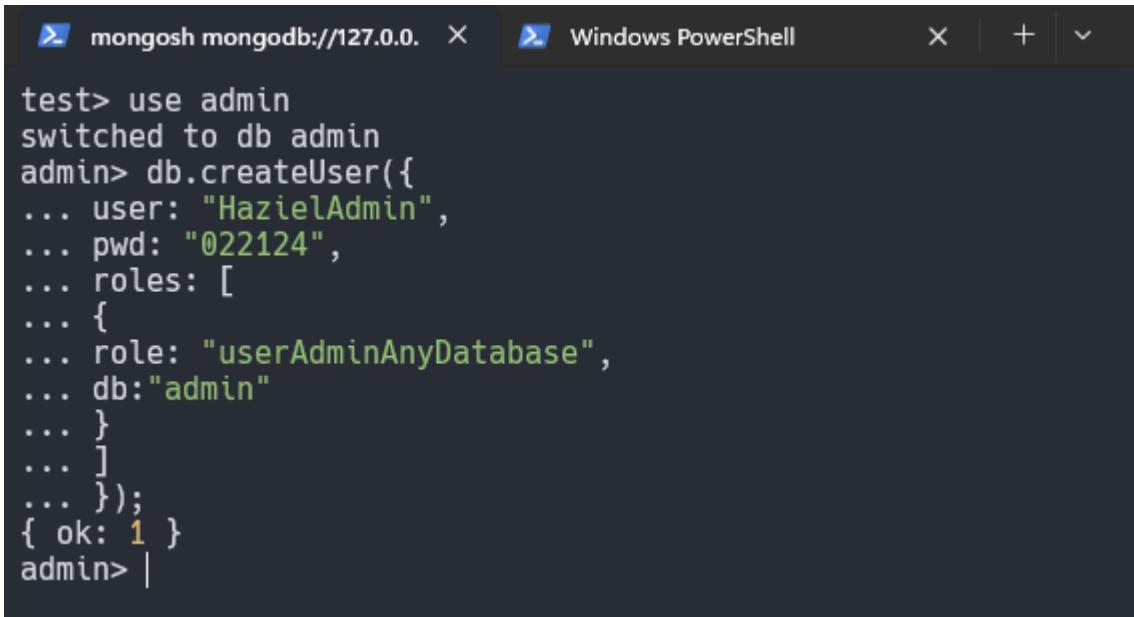
```
admin> db.getUsers()
{
  users: [
    {
      _id: 'admin.Haziel',
      userId: UUID('1a56dfbf-27ce-400a-8ac7-ddb9e7152374'),
      user: 'Haziel',
      db: 'admin',
      roles: [ { role: 'readWrite', db: 'admin' } ],
      mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
    }
  ],
  ok: 1
}
admin> |
```

Imagen 43, comprobación de la creación del usuario

3.1.- Crear Superusuario

Para activar el sistema de autenticación, es necesario crear un usuario que reemplace a admin, pues sin esto no podríamos logearnos.

Lo primero será ejecutar el paso anterior pero para crear un usuario con privilegios de admin se requiere la siguiente instrucción:

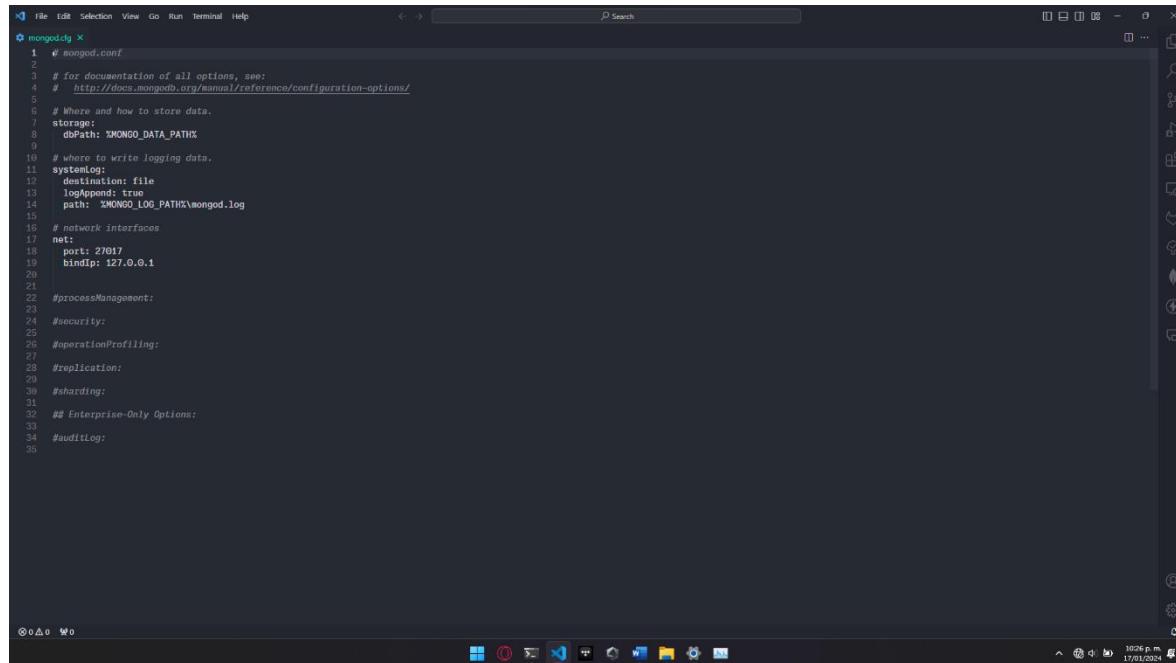


```
mongosh mongodb://127.0.0.1:27017
Windows PowerShell
test> use admin
switched to db admin
admin> db.createUser({
...   user: "HzielAdmin",
...   pwd: "022124",
...   roles: [
...     {
...       role: "userAdminAnyDatabase",
...       db: "admin"
...     }
...   ]
... });
{ ok: 1 }
admin> |
```

Imagen 44, creación de usuario admin

4.- Configurar un puerto por defecto

MongoDB tiene un archivo de configuración llamado **mongod.cfg** el cual contiene las configuraciones para el Daemon de Mongo, si lo abrimos con un editor de texto como **Visual Studio Code**, podemos observar su contenido:



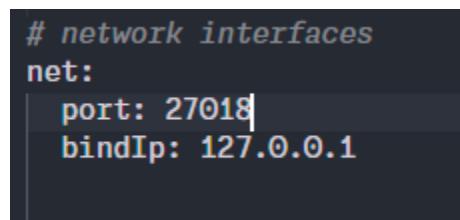
```

1 mongod.cfg
2
3 # for documentation of all options, see:
4 #   http://docs.mongodb.org/manual/reference/configuration-options/
5
6 # Where and how to store data.
7 storage:
8   dbPath: %MONGO_DATA_PATH%
9
10 # where to write logging data.
11 systemLog:
12   destination: file
13   logAppend: true
14   path: %MONGO_LOG_PATH%\mongod.log
15
16 # network interfaces
17 net:
18   port: 27017
19   bindIp: 127.0.0.1
20
21
22 #processManagement:
23
24 #security:
25
26 #operationProfiling:
27
28 #replication:
29
30 #sharding:
31
32 ## Enterprise-Only Options:
33
34 #auditLog:
35

```

Imagen 45, archivo de configuración de mongod

Donde podemos ver la opción de **port**: podemos modificar este valor para que por defecto cambie el puerto y no especificarlo cada que se inicia manualmente MongoD:



```

# network interfaces
net:
  port: 27018
  bindIp: 127.0.0.1

```

Imagen 46, cambio del puerto por defecto

5.- Activar autenticación por defecto

MongoD por defecto se salta la autenticación, por ello es que puede activarse usando el parámetro `--auth` al levantar el daemon o colocar la línea `auth=true (depreciado) authorization: enabled` dentro del `mongod.cfg`:

```
24 security:  
25   authorization: enabled  
26
```

Imagen 47, añadiendo la autenticación

6.- Activar conexión remota

Por defecto MongoDB no admite conexiones fuera del propio localhost (127.0.0.1), para permitir que otra IP pueda conectarse es necesario modificar la siguiente linea dentro de `mongod.cfg`:

```
# network interfaces  
net:  
  port: 27018  
  bindIp: 127.0.0.1
```

Imagen 49, bindIp localhost

```
# network interfaces  
net:  
  port: 27018  
  bindIp: 0.0.0.0
```

Imagen 48, bindIp abierta

Verificación del Servicio

Para verificar que el servicio ha sido configurado correctamente, podemos probar a conectarnos con Mongosh haciendo uso de nuestro usuario en admin y probar el nuevo puerto por defecto:

Cargar Archivo de Configuración

MongoD usa por defecto sus parámetros de configuración, sin embargo en la misma ubicación del ejecutable de MongoD, se encuentra mongod.cfg, el cual puede ayudarnos a especificar las configuraciones de MongoD, sin embargo al ejecutar este, hay que indicar que use este archivo:

```
>> bin  @@ 22:38 ./mongod.exe --config .\mongod.cfg
```

Imagen 50, ejecutando mongod con el archivo .cfg

Esto cargará mongod con la configuración indicada, esto es útil si queremos tener diferentes configuraciones para diferentes situaciones.

Conectarse con mongosh

Para realizar la nueva conexión, se usan nuevos parámetros para poder configurar mongosh:

```
>> Documentos 23:12 mongosh --port 27018 -u HazielAdmin -p --authenticationDatabase admin  
Enter password: *****  
Current Mongosh Log ID: 65a8b3b5f0e92adaf6e3d646  
Connecting to: mongodb://<credentials>@127.0.0.1:27018/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=admin&appName=mongosh+2.1.1  
Using MongoDB: 7.0.5  
Using Mongosh: 2.1.1  
  
For mongosh info see: https://docs.mongodb.com/mongodb-shell/  
test> |
```

Imagen 51, conexión personalizada desde mongosh

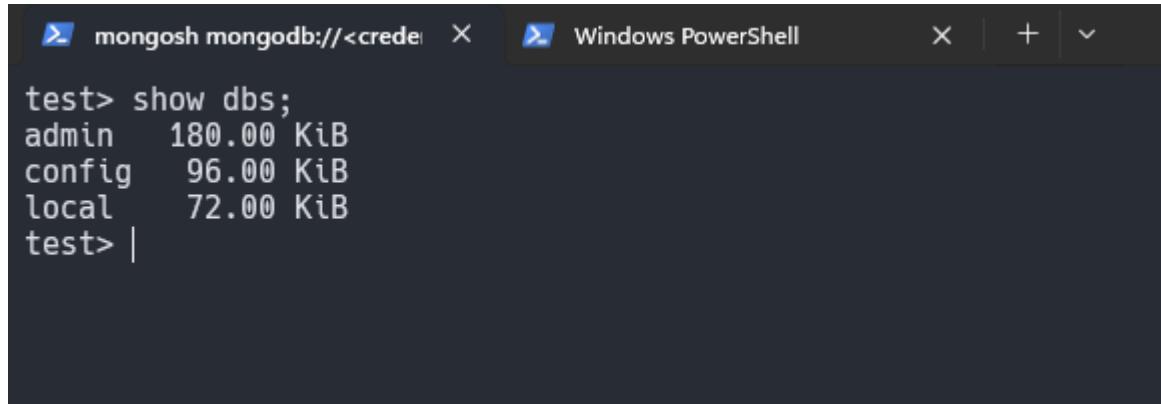
Entre los que destacan:

- **--port:** para especificar el puerto a utilizar
- **-u:** para indicar el nombre de usuario
- **-p:** el cual recibe la contraseña o la solicita al intentar acceder

Creación de una Base de Datos

Crear bases de datos en MongoDB es muy sencillo, para ello se usa el comando **use**, el cual permite cambiar entre bases de datos aunque estás no existan, creandolas en su defecto aunque no por completo.

Primero al estar dentro de mongo ingresaremos el comando **show dbs;** para mostrar las bases de datos igual que en SQL:

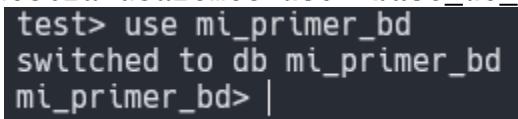


```
mongosh mongodb://<crede...  X  Windows PowerShell  X  +  v
test> show dbs;
admin    180.00 KiB
config   96.00 KiB
local    72.00 KiB
test> |
```

Imagen 52, demostración de las bds disponibles

Por defecto existen admin, config y local, bases de datos propias del sistema y que no deben ser modificadas a la ligera.

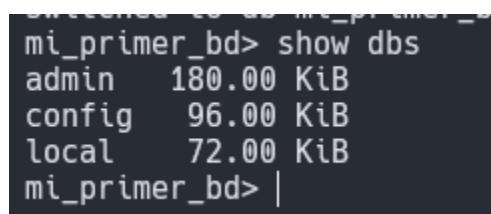
Para crear la nuestra usaremos **use <base_de_datos>**:



```
test> use mi_primer_bd
switched to db mi_primer_bd
mi_primer_bd> |
```

Imagen 53, selección/creación de una bd

Sin embargo, al mostrar las bases de datos, notaremos que no existe **mi_primer_bd**:



```
switched to db mi_primer_bd
mi_primer_bd> show dbs
admin    180.00 KiB
config   96.00 KiB
local    72.00 KiB
mi_primer_bd> |
```

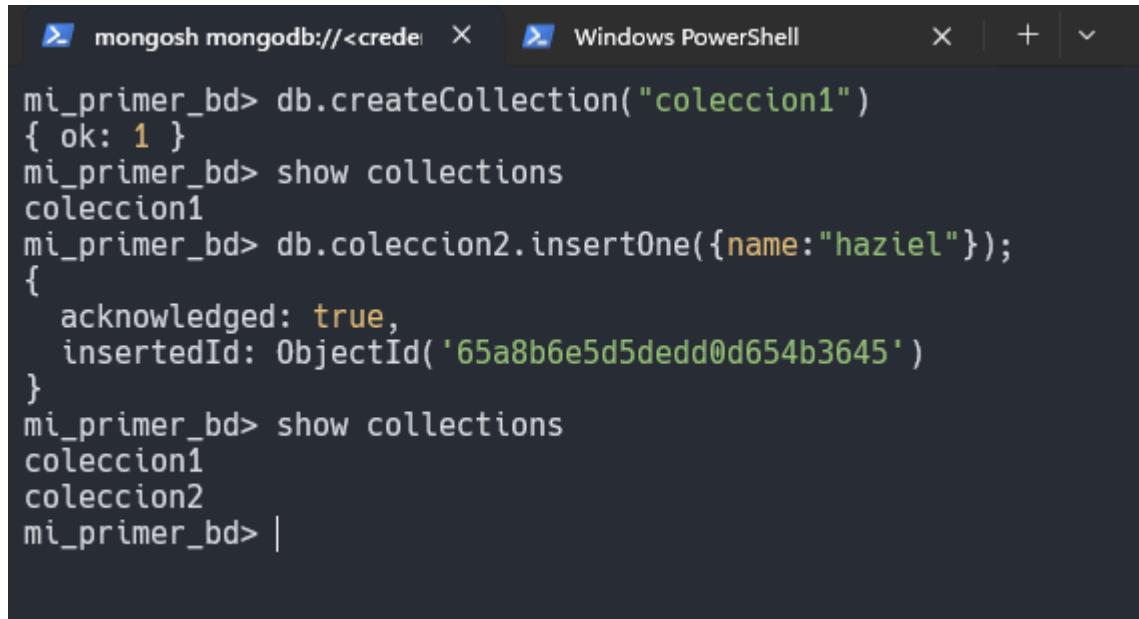
Imagen 54, mostrar bds

Esto es porque Mongo no crea la base de datos hasta que contenga alguna colección al menos y la mantiene en RAM.

Crear una colección

Las colecciones son como las **tablas** de SQL, estás almacenan documentos que en su defecto serían los registros de SQL. Para crear una colección existen dos formas; la manual o la indirecta.

Mongo proporciona el comando **createCollection** para crear manualmente la colección, sin embargo al insertar un documento en una colección, Mongo la crea en caso de que no exista:

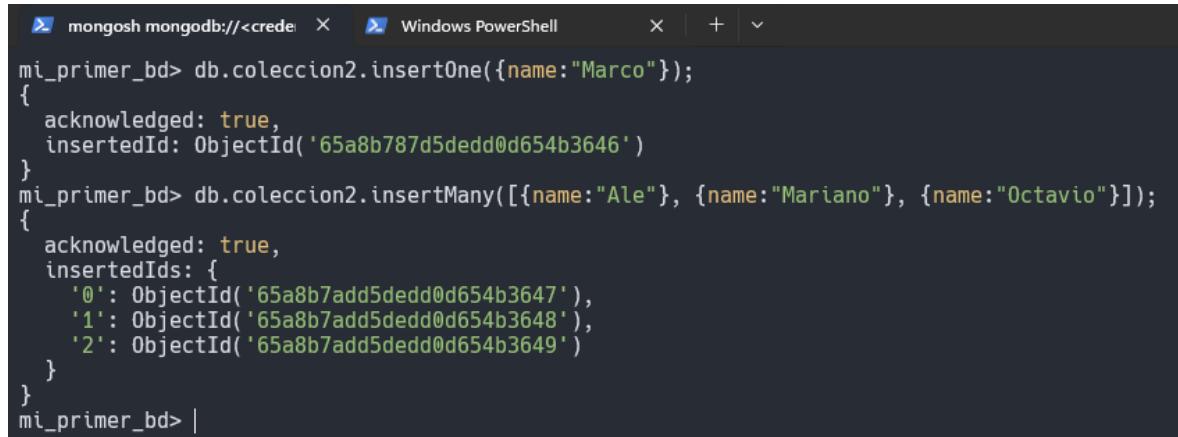


```
mongosh mongodb://<crede... > Windows PowerShell
mi_primer_bd> db.createCollection("colección1")
{ ok: 1 }
mi_primer_bd> show collections
colección1
mi_primer_bd> db.colección2.insertOne({name:"haziel"});
{
  acknowledged: true,
  insertedId: ObjectId('65a8b6e5d5dedd0d654b3645')
}
mi_primer_bd> show collections
colección1
colección2
mi_primer_bd> |
```

Imagen 55, creación de una colección

Insertar Datos

Para crear documentos o los llamados **registros** de SQL, es necesario usar los comandos **insertOne** o **insertMany** los cuales reciben un objeto o un arreglo de objetos respectivamente para insertar estos datos dentro de una colección:



```

mongosh mongodb://<crede> × Windows PowerShell × + ▾
mi_primer_bd> db.coleccion2.insertOne({name:"Marco"});
{
  acknowledged: true,
  insertedId: ObjectId('65a8b787d5dedd0d654b3646')
}
mi_primer_bd> db.coleccion2.insertMany([{name:"Ale"}, {name:"Mariano"}, {name:"Octavio"}]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65a8b7add5dedd0d654b3647'),
    '1': ObjectId('65a8b7add5dedd0d654b3648'),
    '2': ObjectId('65a8b7add5dedd0d654b3649')
  }
}
mi_primer_bd>

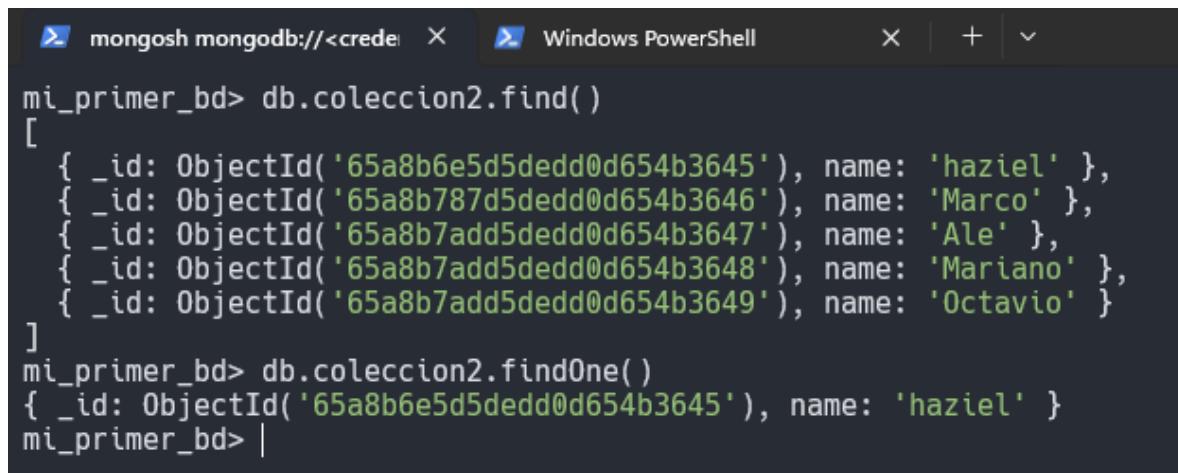
```

Imagen 56, inserción de datos

Consultar Datos

Para mostrar estos datos o documentos almacenados en las colecciones, es necesario usar **find** o **findOne**, los cuales devuelven un arreglo de objetos o un objeto respectivamente.

Para ambos casos es necesario pasarles como parámetro un **filter**, para el caso de un **findOne** que encuentre varios documentos que cumplan con **filter**, devuelve el primero que encontró:



```

mongosh mongodb://<crede> × Windows PowerShell × + ▾
mi_primer_bd> db.coleccion2.find()
[
  { _id: ObjectId('65a8b6e5d5dedd0d654b3645'), name: 'haziel' },
  { _id: ObjectId('65a8b787d5dedd0d654b3646'), name: 'Marco' },
  { _id: ObjectId('65a8b7add5dedd0d654b3647'), name: 'Ale' },
  { _id: ObjectId('65a8b7add5dedd0d654b3648'), name: 'Mariano' },
  { _id: ObjectId('65a8b7add5dedd0d654b3649'), name: 'Octavio' }
]
mi_primer_bd> db.coleccion2.findOne()
{ _id: ObjectId('65a8b6e5d5dedd0d654b3645'), name: 'haziel' }
mi_primer_bd>

```

Imagen 57, consultar datos de una colección

Conexión Remota

Para conectarse remotamente a un servidor externo, es necesario tener instalado en ambas máquinas mongosh y tomar en cuenta las IP de ambas, estas pueden conocerse usando el comando de terminal **ipconfig /all**:

```
Adaptador de LAN inalámbrica Wi-Fi:
Sufijo DNS específico para la conexión. . . : ti.utxicotepec.edu.mx
Descripción . . . . . : MediaTek Wi-Fi 6 MT7921 Wireless LAN Card
Dirección física. . . . . : 90-E8-68-D4-A6-27
DHCP habilitado . . . . . : sí
Configuración automática habilitada . . . . . : sí
Vínculo: dirección IPv6 local. . . . . : fe80::5ff9:be1:848c:cc21%15(Preferido)
Dirección IPv4. . . . . : 172.16.0.69(Preferido) 172.16.0.69(Preferido)
Máscara de subred . . . . . : 255.255.252.0
Concesión obtenida. . . . . : jueves, 18 de enero de 2024 04:31:45 p. m.
La concesión expira . . . . . : jueves, 18 de enero de 2024 05:31:45 p. m.
Puerta de enlace predeterminada . . . . . : 172.16.0.1
Servidor DHCP . . . . . : 172.16.0.1
IAID DHCPv6 . . . . . : 261154920
DUID de cliente DHCPv6. . . . . : 00-01-00-01-2D-32-90-0E-04-42-1A-D6-0D-F6
Servidores DNS. . . . . : 172.16.0.1
8.8.8.8
8.8.4.4
NetBIOS sobre TCP/IP. . . . . : habilitado

Adaptador de Ethernet Conexión de red Bluetooth:
```

Imagen 58, encontrar IP de la maquina servidor

Lo siguiente será desactivar el firewall en ambas máquinas:

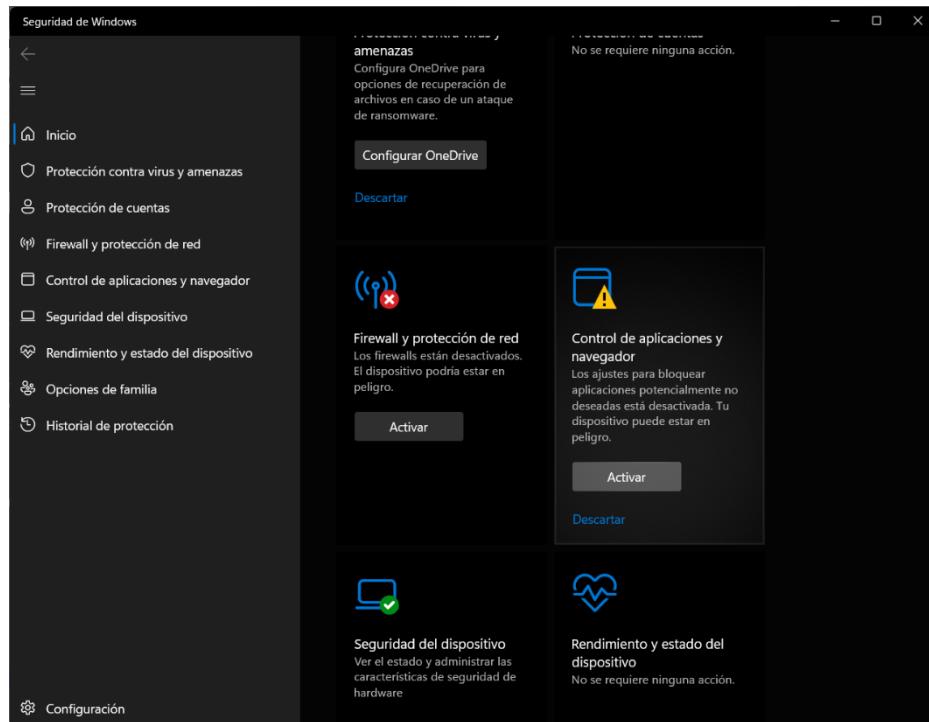


Imagen 59, firewall de Windows desactivado

Por último, desde la máquina cliente, se debe ejecutar **mongosh** usando los siguientes parámetros:

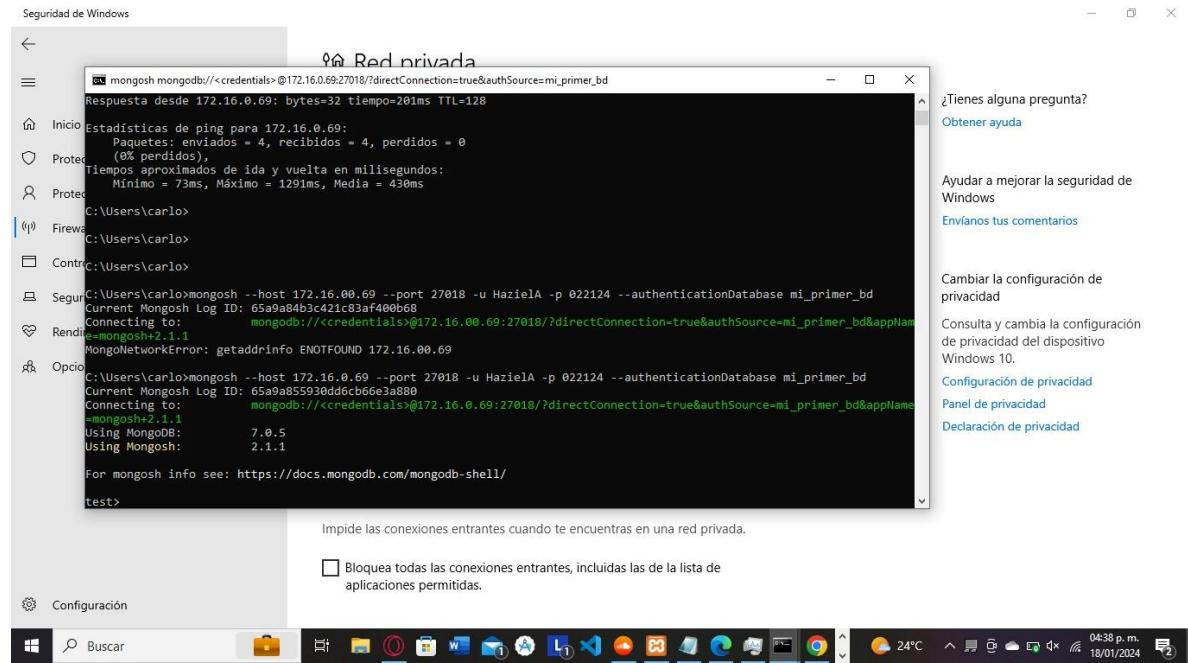


Imagen 60, conexión desde la máquina cliente

Los parámetros importantes son:

- **--host:** La ip del servidor
- **--port:** El puerto del servidor
- **-u:** Nombre del usuario
- **-p:** Contraseña del usuario
- **--authenticationDatabase:** Base de datos a conectarse

Conclusiones

Las bases de datos NoSQL han tomado mucha fuerza en los últimos años y han crecido bastante en popularidad y mejorado bastante, siendo actualmente una gran competencia para SQL, sin embargo es cierto que no planean competir, solo ser una alternativa con sus propios fines como lo es el **rendimiento** a coste de integridad.

Uno de los puntos que las han favorecido es su facilidad de uso y comprensión frente a las algo tediosas SQL y sus esquemas rigurosos, sintaxis más avanzada que lleva más tiempo acostumbrarse.

MongoDB es un potente gestor de base de datos basada en documentos, la cual es una de las tecnologías más solicitadas para los nuevos desarrolladores y que es realmente beneficioso aprender para hacer más robusto nuestro perfil profesional.

A pesar de que Mongo por defecto no viene activado con ninguna medida de seguridad, es necesario profundizar un poco en su documentación y leer para poder hacer buenas prácticas de configuración del mismo que se suelen omitir a la hora de aprender, pero que a profesionalmente son temas que son bastante importantes y que es seguro deben aplicarse en escenarios reales para salvaguardar la integridad de la empresa y sus productos.