

Módulo: BASES DE DATOS

Unidad 7: Uso de bases de datos no relacionales

Sesión 2: Instalación y explotación de SGBD no relacionales

Descripción:

La presente sesión vamos a ver cómo funciona un SGBD en concreto como en MongoDB.

Primero veremos sus características y cómo funciona. Para después probar las herramientas que nos proveen para trabajar con esta base de datos.

Para todo esto, instalaremos la base de datos MongoDB y el SGBD que nos da para poder administrarla. Haciendo algunas pruebas para ver como se puede insertar, modificar o consultar valores en esta BD.

Criterios de Evaluación:

- RA07_d: Se han identificado distintas formas de gestión de la información según el tipo de base de datos no relacionales.
- RA07_e: Se han utilizado las herramientas del sistema gestor para la gestión de la información almacenada.

Objetivos:

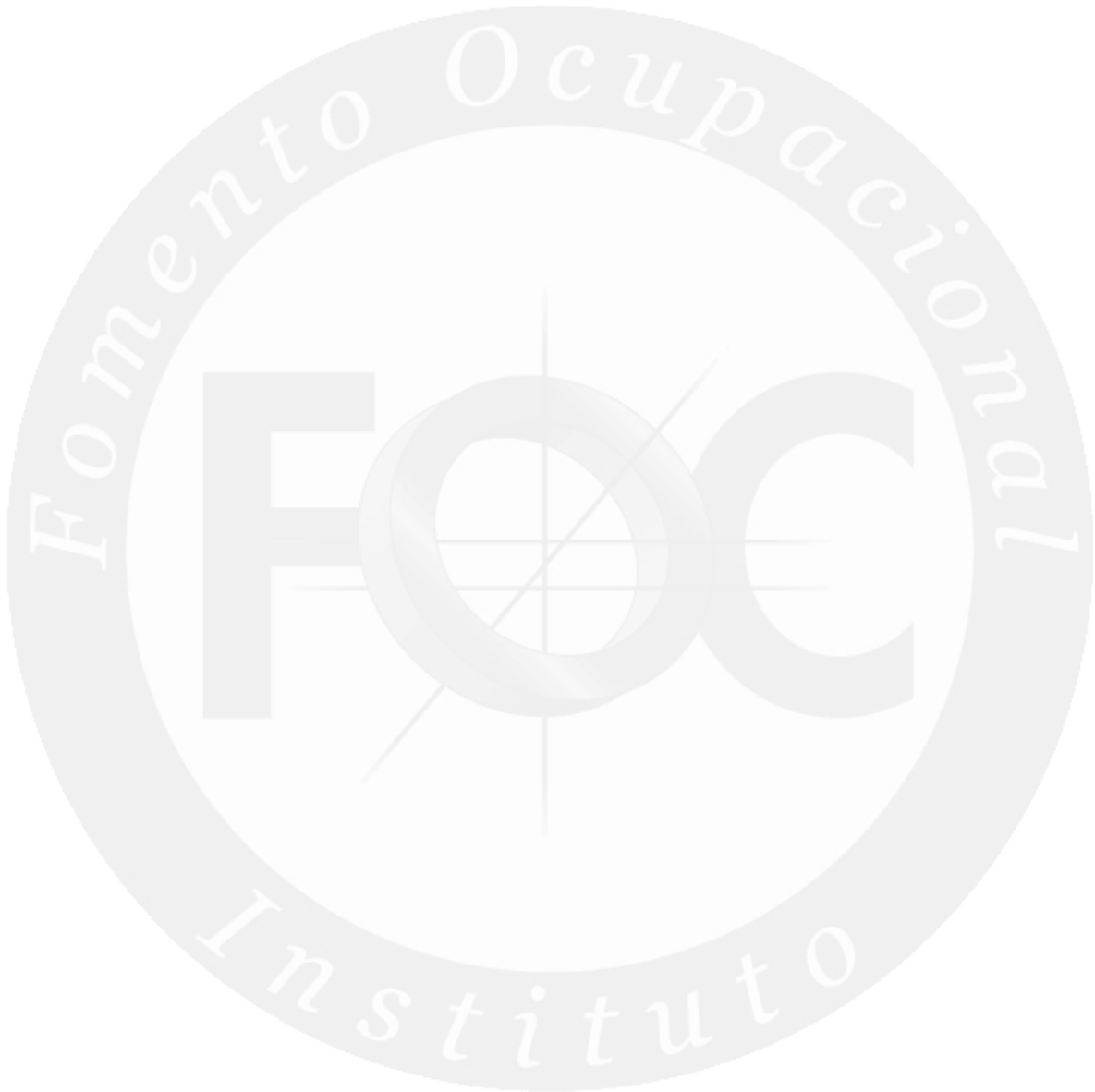
- Conocer distintas formas de gestión de la información según el tipo de base de datos no relacionales.
- Instalar herramientas para gestionar bases de datos no relacionales.
- Utilizar herramientas que nos permitan gestionar bases de datos no relacionales.

Recursos:

- Acceso a Internet.
- Software ofimático.
- MongoDB

Conceptos a revisar previamente:

- Herramientas de los sistemas gestores de bases de datos no relacionales para la gestión de la información almacenada.



Resolución de la práctica:

MongoDB

Es una base de datos NoSQL, orientada a documentos, escrita en C++ y de código abierto. Se basa en almacenar documentos de tipos BSON que es una representación binaria de JSON, lo cual es muy potente ya que permite que los datos dentro de los mismos sean consultables e indexables.



Imagen: Logo de MongoDB.

Características:

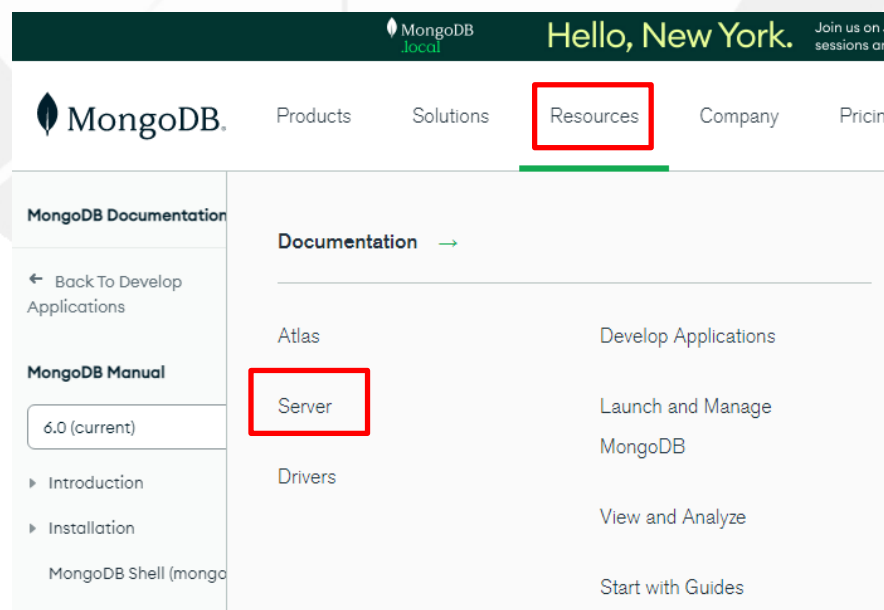
- **No es necesario seguir un esquema.** Los documentos de una misma colección, que sería el equivalente a una tabla, pueden utilizar esquemas diferentes.
- **Réplica primaria.** Hay un servidor principal, que interactúa con el cliente y realiza todas las operaciones. Además, hay réplicas secundarias mantienen una copia de los datos. Si hay un fallo en la primaria, se cambia a la secundaria automáticamente, para poder garantizar una alta disponibilidad.
- **Fragmentos.** Sharding es la partición y distribución de datos a través de múltiples nodos. Por lo que un fragmento es una colección de nodos MongoDB. El uso de fragmentos también implica escalabilidad horizontalmente en múltiples nodos. En el caso de que haya una aplicación que utilice un único servidor de base de datos, se puede convertir en clúster fragmentado.
- **Lenguaje de consulta.** Es una API RESTful, para hacer consultas tenemos que pasar objetos JSON como parámetro, por lo que crearemos un documento de consulta que contiene los campos que deben coincidir con los documentos deseados.
- **Indexación.** Existen índices similares los de las bases de datos relacionales, pero permitiendo que cualquier campo documentado puede ser indexado y añadir múltiples índices secundarios.

Instalar MongoDB

Para instalar MongoDB tenemos que ir a la web oficial. <https://www.mongodb.com/es>



Entrar en la zona de recursos (resources) y dentro de servidores (server), encontraremos las diferentes versiones que podemos descargar de MongoDB junto con documentación.



Debemos buscar la instalación de MongoDB Community Edition que es la que nos permite usar MongoDB de manera gratuita con fines educativos.

What is MongoDB?

MongoDB is a document database designed for ease of application development and scaling.

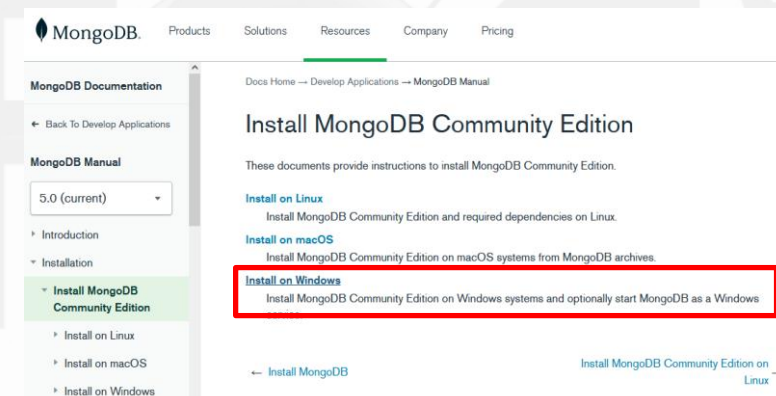
Run MongoDB with

- [MongoDB Atlas](#) fully managed in the cloud,
- the source available and free-to-use [MongoDB Community](#) or
- the [MongoDB Enterprise Advanced](#) subscription.

[Install MongoDB](#)

[Get started with MongoDB Atlas](#)

Seleccionamos la versión para la plataforma donde lo queramos instalar, nosotros lo haremos para Windows.



Veremos que nos indica que se puede usar desde la nube con MongoDB Atlas que es un servicio de base de datos totalmente gestionado de pago. Y tenemos un amplio tutorial donde nos explica como instalarlo paso a paso que si lo seguimos nos llevará a los archivos que tenemos que descargar.

Install MongoDB Community Edition on Windows

NOTE

MongoDB Atlas

MongoDB Atlas is a hosted MongoDB service option in the cloud which requires no installation overhead and offers a free tier to get started.

Overview

Use this tutorial to install MongoDB 6.0 Community Edition on Windows using the default installation wizard.

MongoDB Version

This tutorial installs MongoDB 6.0 Community Edition. To install a different version of MongoDB Community, use the version drop-down menu in the upper-left corner of this page to select the documentation for that version.

MONGODB COMMUNITY SERVER

MongoDB Community Server Download

The Community version of our distributed database offers a flexible document data model along with support for ad-hoc queries, secondary indexing, and real-time aggregations to provide powerful ways to access and analyze your data.

The database is also offered as a fully-managed service with [MongoDB Atlas](#). Get access to advanced functionality such as auto-scaling, serverless instances, full-text search, and data distribution across regions and clouds. Deploy in minutes on AWS, Google Cloud, and/or Azure, with no downloads necessary.

[Give it a try with a free, highly-available 512 MB cluster.](#) or get started from your terminal with the following two commands:

```
$ brew install mongodb-atlas  
$ atlas setup
```

Version
6.0.6 (current)

Platform
Windows

Package
msi

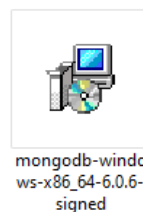
Download

Copy link

More Options

Imagen: MongoDB Descarga

Una vez descarguemos el archivo de instalación, lo ejecutaremos y nos saldrá un asistente.



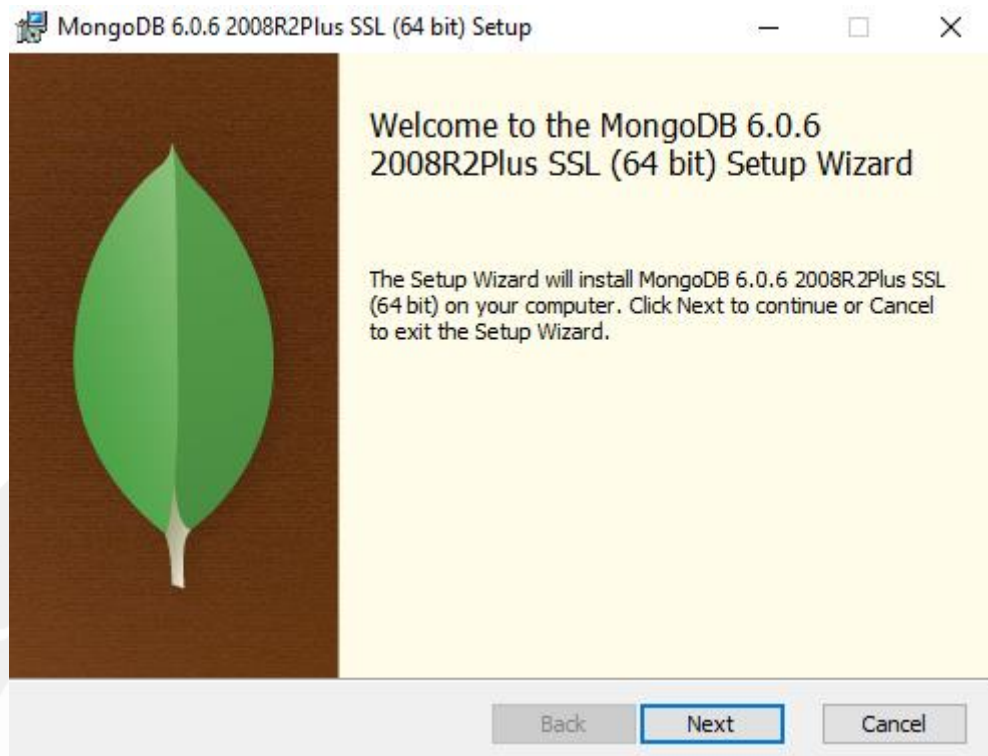


Imagen: MongoDB Asistente de Instalación (I)

Aceptamos la licencia y nos pedirá el tipo de instalación, con la instalación completa será suficiente.

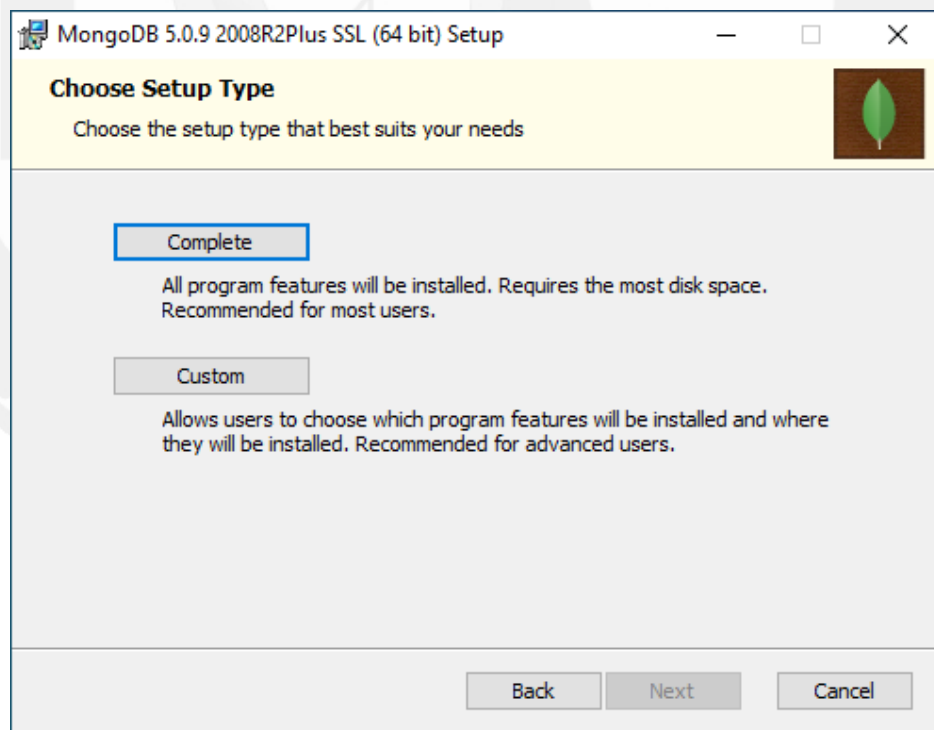


Imagen: MongoDB Asistente de Instalación (II)

Elegiremos instalarlo como un servicio web.

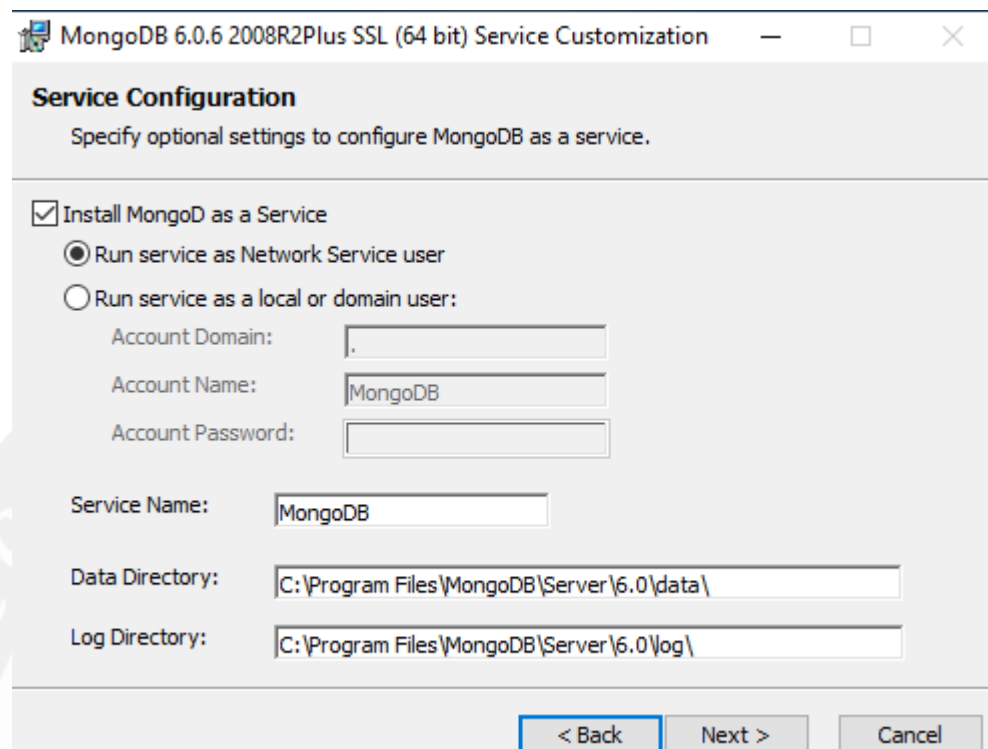
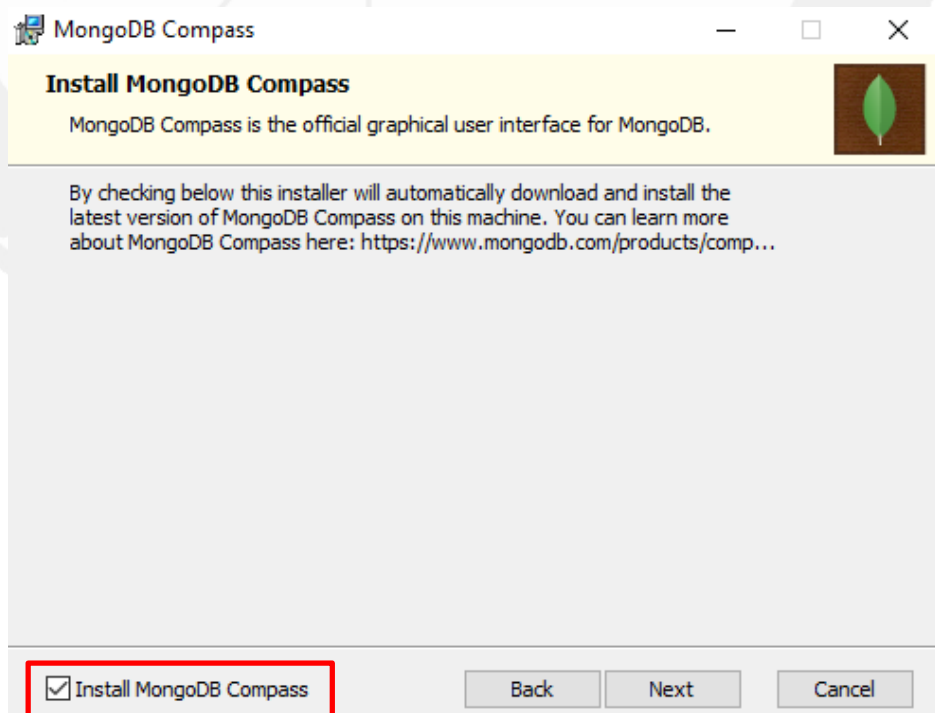


Imagen: MongoDB Asistente de Instalación (III)

Una vez configurada la instalación podemos comenzar con ella. Recodemos Instala MongoDB Compass que es el interfaz de usuario que nos va a permitir gestionar las BDs que creemos.



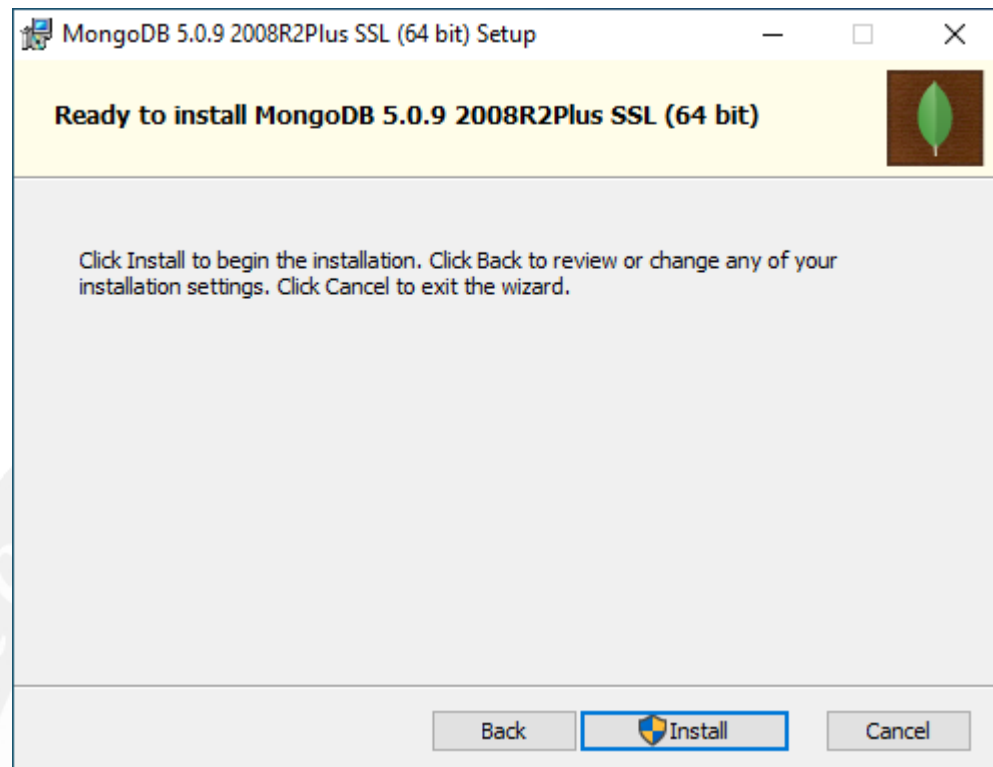


Imagen: MongoDB Asistente de Instalación (IV)

Tomará su tiempo en instalar, configurar y crear los servicios.

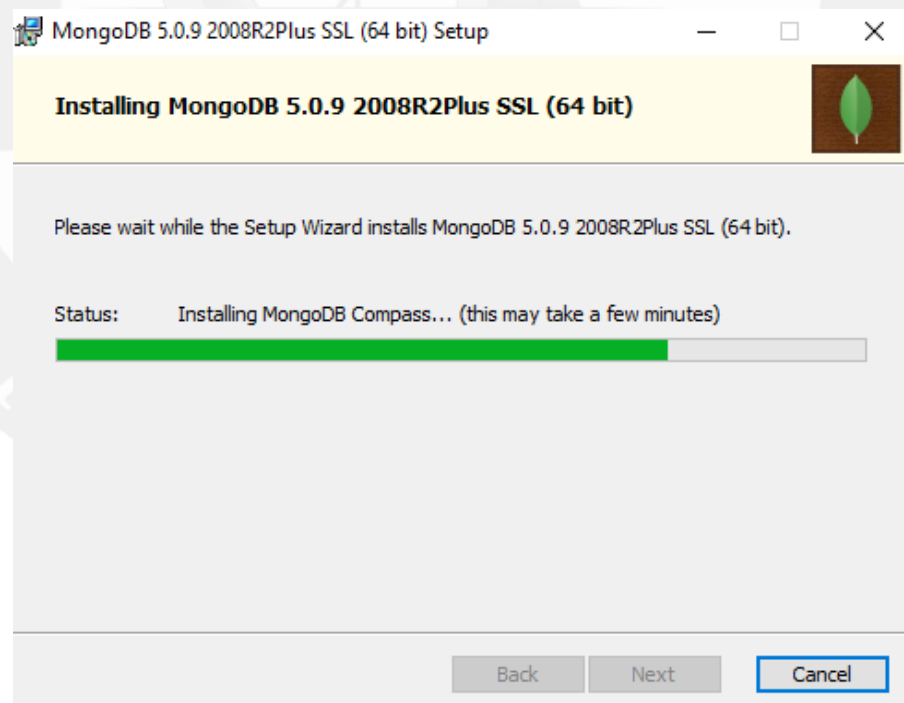


Imagen: MongoDB Asistente de Instalación (V)

Aparecerá una pantalla de carga de MongoDB Compass que es una herramienta interactiva para consultar, optimizar y analizar sus datos en MongoDB.

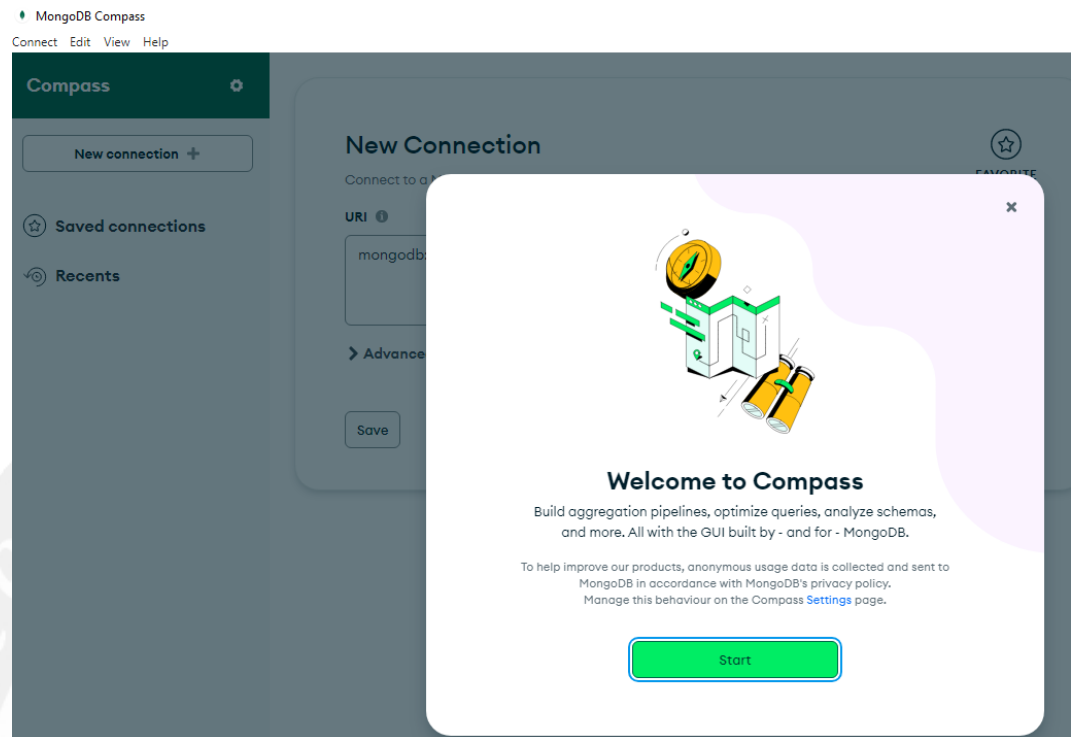
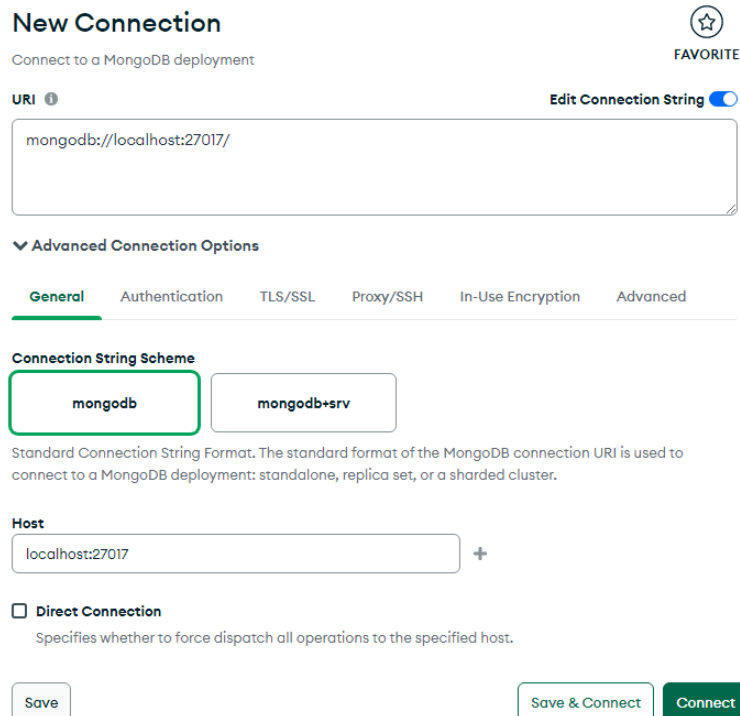


Imagen: MongoDB Pantalla de Carga

Explotar MongoDB

Una vez entremos accederemos al menú de gestión del SGBD. Nos conectaremos pulsando en el botón “connect”.



New Connection FAVORITE

Connect to a MongoDB deployment

URI ⓘ Edit Connection String

mongodb://localhost:27017/

▼ Advanced Connection Options

General Authentication TLS/SSL Proxy/SSH In-Use Encryption Advanced

Connection String Scheme

mongodb mongodb+srv

Standard Connection String Format. The standard format of the MongoDB connection URI is used to connect to a MongoDB deployment: standalone, replica set, or a sharded cluster.

Host

localhost:27017 +

☐ Direct Connection

Specifies whether to force dispatch all operations to the specified host.

Save Save & Connect Connect

Imagen: MongoDB Conectarse

Una vez conectados nos aparecerá en el menú de gestión el acceso a las bases de datos que tenemos y también a otras opciones como pueden ser las consultas y el rendimiento.

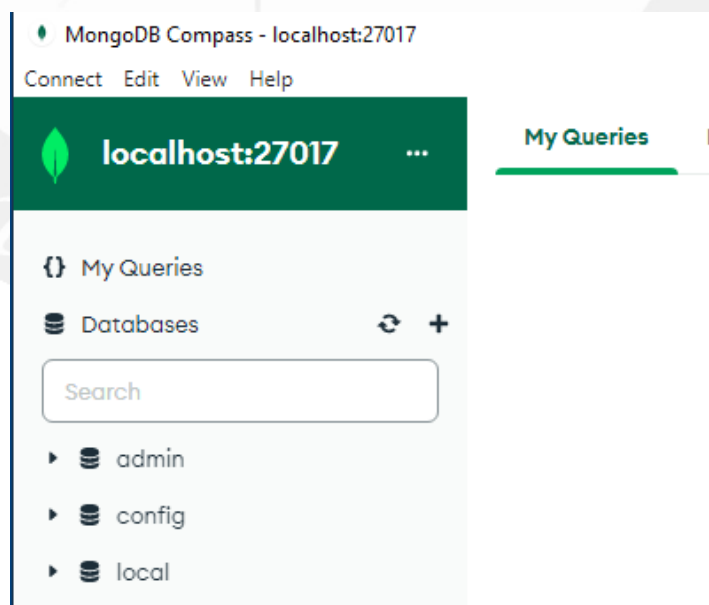


Imagen: MongoDB Menú de Gestión

Vamos al menú de bases de datos y encontraremos las predefinidas por el sistema de admin, config y local. Podremos crear nuevas bases de datos en el botón “Create Database” o en el símbolo de + donde están las Databases.

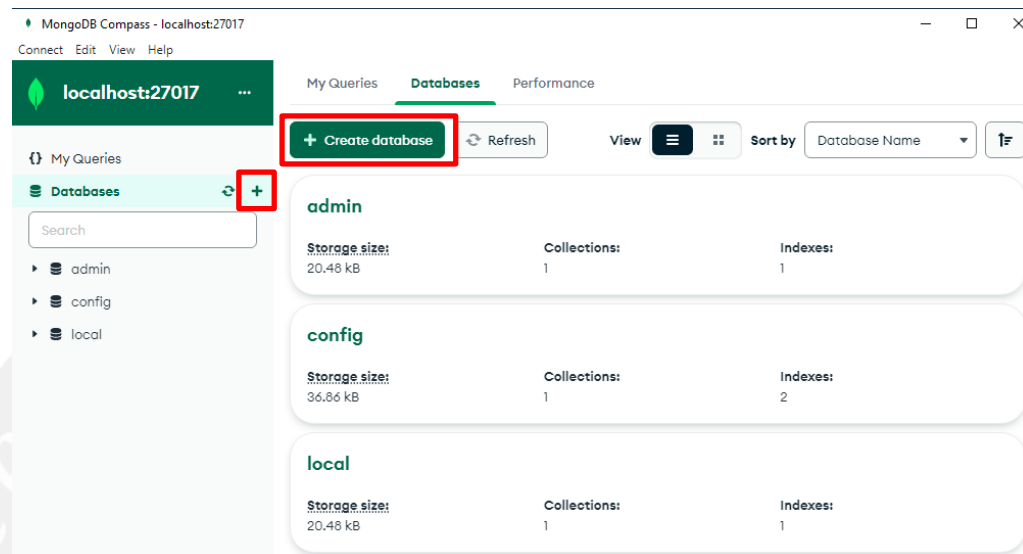


Imagen: MongoDB Menú de Bases de Datos

Nos aparecerá un formulario para rellenar el nombre de la base de datos y el nombre de la colección, que sería el equivalente a tabla o conjunto de tablas en un modelo relacional.

Create Database

Database Name:

Collection Name:

☐ Time-Series
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

☒ Additional preferences (e.g. Custom collation, Capped, Clustered collections)

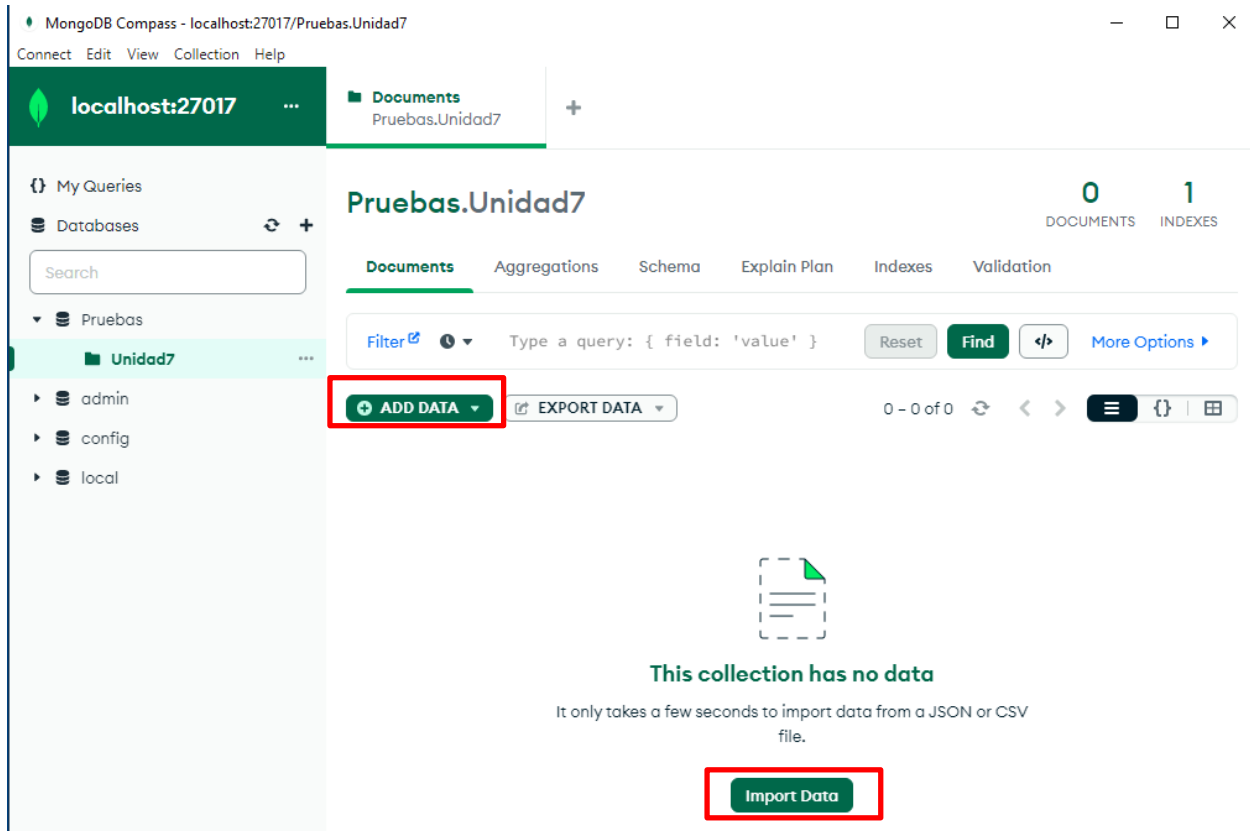
☐ Capped Collection
Fixed-size collections that support high-throughput operations that insert and retrieve documents based on insertion order. [Learn More](#)

☐ Use Custom Collation
Collation allows users to specify language-specific rules for string comparison, such as rules for lettercase and accent marks. [Learn More](#)

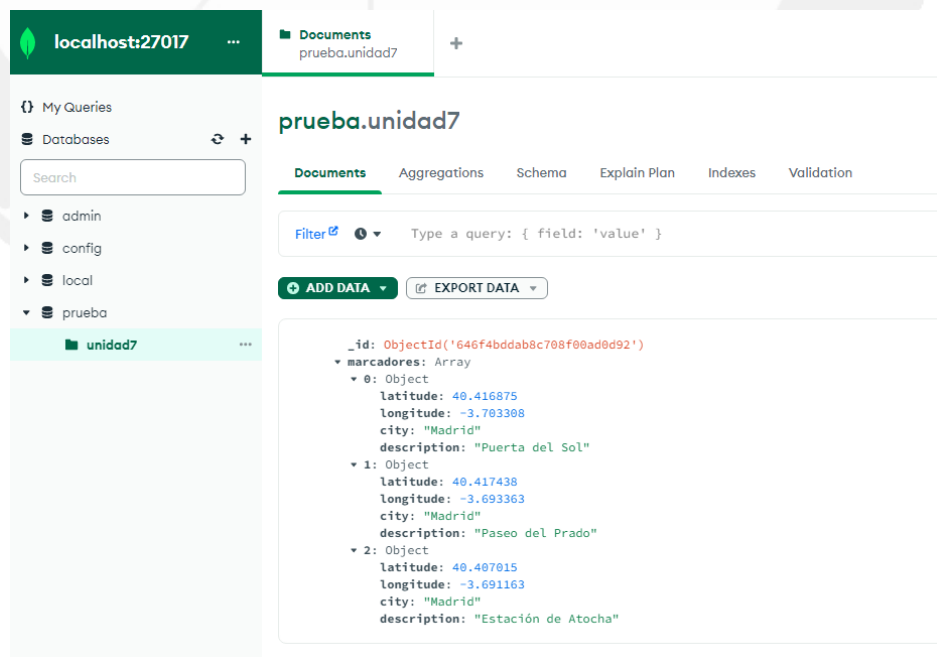
Imagen: MongoDB Crear nueva base de datos

Hay opciones para crear BD para series temporales, para poner un máximo de tamaño, reglas propias de comparación de strings o indexadas en un cluster.

Si vamos a la nueva base de datos que hemos creado, nos aparecerán las opciones que podemos hacer sobre ella.



Ahora podemos importar un archivo JSON a la base de datos, para ello vamos al botón Add data o a Import Data y nos aparecerá un menú seleccionamos un fichero JSON o CSV y le damos al botón Import. Podemos pedirle que se pare si hay algún error.



Vamos a añadir diferentes archivos JSON para ver como funcionan estos archivos y ver

Se trata de 3 archivos que contienen aproximadamente la misma información. ¿Cuál es la diferencia entre ellos?

En el archivo colores1.json existe un único objeto de datos donde el nombre es arrayColores y el valor un array de objetos JSON. Cada objeto del array está formado por los pares nombreColor y su valor, y valorHexadec y su valor. En este ejemplo en concreto el array consta de 7 elementos con información correspondiente a 7 colores.

En el archivo colores2.json existe un único objeto de datos donde el nombre es arrayColores y el valor un array que contiene un único objeto JSON formado por siete pares nombre – valor que representa información sobre siete colores.

En el archivo colores3.json existe un único objeto de datos que está formado por siete pares nombre – valor que representa información sobre siete colores.

¿Cuál de los 3 archivos nos conviene usar? ¿Cuál es el mejor?

Estas preguntas no tienen una respuesta directa. La respuesta es: depende. Un archivo de datos se puede organizar de múltiples maneras. Cómo hacerlo entra dentro del campo de lo que se denomina “diseño de datos”. En cada aplicación, programa, desarrollo web, etc. debe tomarse en consideración cuál es el diseño de datos más adecuado para alcanzar los objetivos propuestos.

Consultas.

Ahora podemos hacer consultas sobre los datos aplicando 3 opciones principalmente, entre otras, para obtener la información:

- Un **filtro (filter)**, lo que sería poner una condición a uno de los campos, equivalente a hacer un WHERE en SQL.

Si por ejemplo ponemos, como se ve en la imagen “*MongoDB Consultas con filtros*”, { gender: “Male” } nos devolverá los datos que tengan el campo gender y su valor coincida exactamente con Male.

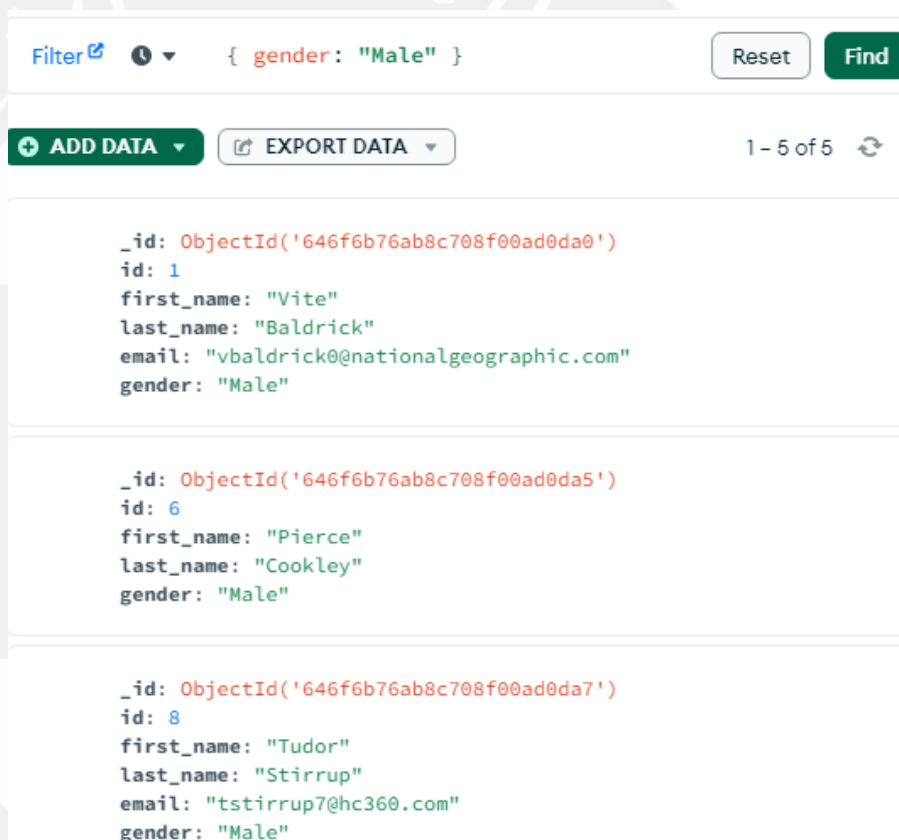


Imagen: MongoDB Consultas con filtros

- Una **proyección (project)**, lo que sería decir qué campos quieres que sean devueltos, equivalente a hacer un SELECT en SQL. Si al campo que proyectamos le damos el valor 1 nos devolverá ese campo si le ponemos el valor 0 nos devolverá los campos que no sean ese.

Si por ejemplo ponemos, como se ve en la imagen “*MongoDB consultas usando proyección*”, {first_name : 1} nos devolverá los campos first_name que encuentre en los datos.

Documents Aggregations Schema Explain Plan Indexes Validc

Filter Type a query: { field: 'value' }

Project {first_name: 1}

Sort { field: -1 } or [['field', -1]] MaxTimeMS

Collation { locale: 'simple' Skip 0 Limit

EXPORT DATA 1 - 10 of 10

```

_id: ObjectId('646f6b76ab8c708f00ad0da0')
first_name: "Vite"

_id: ObjectId('646f6b76ab8c708f00ad0da1')
first_name: "Iseabal"

_id: ObjectId('646f6b76ab8c708f00ad0da2')
first_name: "Liz"


_id: ObjectId('646f6b76ab8c708f00ad0da3')
first_name: "Nickie"

```

Imagen: MongoDB consultas usando proyección

- Una **ordenación (sort)**, lo que sería especificar porqué campos quieres se ordene los resultados que nos devuelva, equivalente a hacer un ORDER BY en SQL. Si al campo que ordenamos le damos el valor 1 nos devolverá los resultados ordenados ascendentemente y si le ponemos el valor -1 nos devolverá ordenado descendentemente.


Si por ejemplo ponemos, como se ve en la imagen “MongoDB consultas con ordenación”, {first_name : -1} nos devolverá los campos ordenados descendentemente por el valor del campo first_name.

Filter  ⌵ Type a query: { field: 'value' } Reset Find

Project { field: 0 }

Sort {first_name: 1} **MaxTimeMS**

Collation { locale: 'simple' **Skip** 0 **Limit**

+ ADD DATA EXPORT DATA 1 - 10 of 10 

```
_id: ObjectId('646f6b76ab8c708f00ad0da4')
id: 5
first_name: "Bettine"
last_name: "McQuirk"
gender: "Female"
```

```
_id: ObjectId('646f6b76ab8c708f00ad0da8')
id: 9
first_name: "Corbin"
last_name: "Perring"
gender: "Male"
```

```
_id: ObjectId('646f6b76ab8c708f00ad0da6')
id: 7
first_name: "Donaugh"
last_name: "Norssister"
```

Imagen: MongoDB consultas con ordenación